



RADIANCEKIT

Manual de Usuario

Reconstrucción 3D fotorrealista
mediante Gaussian Splatting

Versión 1.5.0 · macOS 26.0+ · Mayo 2026

BJOERN KINDLER · KINDLER-DEV.DE

Resumen

Introducción — Lo que debes saber	3
¿Qué es RadianceKit?	3
¿Qué es Gaussian Splatting?	3
Capítulo 1 — Barra de menús	5
Menú File	6
Menú Mode	9
Menú Training	11
Menú Viewport	14
Menú Export	20
Menú Help	25
Aviso: Cmd-Z en el menú Edit	29
Resumen de atajos de teclado	30
Capítulo 2 — Inspector (Vista Experto)	31
Sección Look (L1–L5)	34
Sección Presets (I1–I11)	37
Sección Training Configuration (I12–I22)	43
Sección Enhancements (I26–I29, I42–I44)	50
Sección Metrics (I30–I38)	57
Sección Loss Chart (I39–I41)	63
¿Cuándo recurrir al Inspector?	66
Capítulo 3 — Ajustes	68
Pestaña General	69
Pestaña AI Helpers	75
Ajustes espejo del Inspector	78
¿Cuándo qué?	79
Capítulo 4 — Ventanas auxiliares	80
User Guide (W1–W4)	81
Keyboard Shortcuts (W5–W6)	84
Manage Storage (W7–W12)	86
Pareto Dashboard (W13–W22)	90
Holdout Analysis (W23–W29)	97
BayesOpt Console (W30–W39)	102
Ventana principal: curva de loss y número de Gaussianos (I39–I41, referencia cruzada)	110
Caja de reglas prácticas	111
Capítulo 6 — Configuración del entrenamiento	112
Iteración (T1–T2)	114

Learning Rates (T3–T10)	116
Densificación — Classic (T11–T16)	124
Loss (T17–T20)	129
Progresión de grado SH (T21)	132
Rendimiento (T22–T25)	133
Diagnóstico y preparación de la nube de puntos (T26–T30)	136
Regularización (T31–T37)	140
Refinamiento (T38–T44)	144
Sky Dome (T45–T48)	149
Adam + schedule de LR (T49–T55)	151
Post-procesado + Apple AI (T56–T60)	155
Densificación MCMC (T61–T73)	157
Mip-Splatting (Q1.5) (T74–T76)	164
Densificación adaptativa (Q5) (T77–T79)	167
Currículum (Q6) (T80–T81)	168
Ajustes preestablecidos estáticos (TP1–TP9)	169
Método:	172
¿Qué campo para qué? (Cheat Sheet)	173
Campos peligrosos	174
Capítulo 7 — Ajustes preestablecidos de calidad integrados	175
¿Qué ajuste preestablecido en cada caso?	185
Comparativa rápida	186
Ajustes preestablecidos personalizados	188
Capítulo 8 — Formatos de exportación	189
¿Qué formato en cada caso?	203
Comparativa rápida	204
Capítulo 9 — Backends de SfM	205
¿Qué backend en cada caso?	212
Comparativa rápida	213
Capítulo 10 — Modo principiante	214
Z1 — Importar (elegir imágenes y ajuste preestablecido)	215
Z2 — Procesado (SfM + Entrenamiento)	224
Z3 — Previsualización (rotar modelo 3D)	232
Z4 — Exportar (elegir formato y guardar)	235
Cambiar a Modo Experto	240
Preguntas frecuentes	241

Cómo leer este manual

Cada entrada de este manual sigue el mismo esquema. En el lado izquierdo están las rutas de operación y los detalles técnicos; a la derecha, en una barra lateral cálida, encuentras siempre la explicación en lenguaje claro. Pequeños iconos al inicio de cada línea te indican de un vistazo qué tipo de información viene a continuación.

LOS CUATRO ICONOS



Dónde lo encuentro. La ruta concreta de clics por la app — barra de menús, sección del Inspector o paso del modo principiante. Aquí también aparecen los atajos de teclado asociados. El icono es un pin de mapa y muestra: aquí se sitúa la función en la interfaz de usuario.



Detalles. Valores predeterminados, rangos de valores y rutas de código. Te lo encuentras sobre todo en los ajustes de entrenamiento, que no son una entrada de menú sino parámetros numéricos. El icono muestra una pequeña tarjeta de especificación.



Técnico. Lo que la función hace internamente, qué parámetros surten efecto, a qué reacciona y qué efectos secundarios tiene. Para lectores que quieren entender qué está pasando detrás de las bambalinas. El icono es un bloque de deslizadores y representa simbólicamente las palancas bajo el capó.



En pocas palabras. El mensaje principal con palabras claras — sin jerga, sin código. Lee esta sección primero si solo quieres saber rápidamente para qué sirve una función y cuándo la necesitas. El icono es un bocadillo y representa «en resumen». Esta columna siempre está sobre un tono arena cálido para que el ojo la encuentre de inmediato.

COLORES DE CAPÍTULO

Cada capítulo tiene su propio color de acento, que reconocerás por la insignia de ID (por ejemplo **M1**) a la izquierda de cada título de entrada y por los pequeños iconos previos. Al hojear, ves de un vistazo en qué capítulo te encuentras.

- | | | | | |
|----------------------------------|------------------------|------------------|------------------------------|------------------------|
| 1 Menús | 2 Inspector | 3 Ajustes | 4 Ventanas auxiliares | 6 Entrenamiento |
| 7 Ajustes preestablecidos | 8 Exportaciones | 9 SfM | 10 Modo principiante | |

CONSEJOS DE NAVEGACIÓN

Inicio rápido. Si solo te interesa la operación de la app, salta directamente al **Capítulo 10 — Modo principiante**. Es la variante guiada con cuatro pasos y no requiere conocimientos previos.

Profundizando. **Capítulo 2 — Inspector** y **Capítulo 7 — Ajustes preestablecidos** explican los controles y los perfiles de calidad preconfigurados disponibles en el Modo Experto.

Consultar. El índice y la búsqueda de texto completo del PDF te ayudan a encontrar una función concreta. No tienes que leer el manual de principio a fin.

Introducción — Lo que debes saber

¿Qué es RadianceKit?

RadianceKit es una app nativa de macOS que convierte una serie de fotos normales o un vídeo en una reconstrucción 3D recorrible. La entrada es, por ejemplo, de 50 a 500 tomas que hayas hecho alrededor de un objeto, por una habitación o sobre un paisaje. La salida es lo que se conoce como una escena de Gaussian-Splatting — un modelo 3D que puedes ver en el Mac en tiempo real desde cualquier ángulo, que se puede exportar e incrustar en sitios web y que, en sus aspectos principales, parece fotorrealista.

La app funciona por completo de forma local en tu Mac — no se suben imágenes a la nube, no se requiere login y no hay suscripción. Hace uso intensivo de la GPU de tu Mac con Apple Silicon (serie M): un entrenamiento completo puede durar entre dos minutos y varias horas según la escena y el ajuste preestablecido. Mientras la computación está en curso, puedes seguir trabajando con total normalidad en el Mac; RadianceKit continúa en segundo plano y te avisa cuando el resultado está listo.

Hay dos modos de operación: el *Modo principiante* (Simple Mode) te guía en cuatro pasos por el flujo Importar → Elegir ajuste preestablecido → Entrenar → Exportar. El *Modo Experto* (Expert Mode) abre un gran Inspector con todos los controles, una ventana de previsualización en vivo y gráficos de diagnóstico. Puedes cambiar entre modos en cualquier momento; los datos de la escena se conservan.

¿Qué es Gaussian Splatting?

Gaussian Splatting (a menudo abreviado como 3DGS o simplemente *Splatting*) es un método relativamente nuevo para representación 3D fotorrealista, presentado en 2023 en un paper de Graz e INRIA. La idea: en vez de modelar una escena como una malla clásica de polígonos (triángulos) o como una rejilla de vóxeles, se compone a partir de millones de pequeñas y suaves «nubecitas» 3D — cada nubecita individual es una distribución Gaussiana 3D (de ahí el nombre) con su propia posición, tamaño, forma, color y transparencia. Estas nubecitas se entrenan para que, en conjunto desde todos los ángulos de visión de tus fotos de entrada, den la imagen correcta.

En la práctica, esto significa que Gaussian Splatting puede representar reflejos, brillos, follaje suave, pelo o cortinas como no puede el modelado 3D clásico — o solo a un coste enorme. A cambio, el resultado no es un modelo 3D editable en sentido clásico — no puedes simplemente mover una pared o reubicar un jarrón. Es más bien una *captura congelada* del espacio por la que puedes moverte con libertad. Para muchas aplicaciones — visualización arquitectónica, presentación de producto, tours virtuales, ciencias forenses, patrimonio cultural — esa es exactamente la fortaleza adecuada.

Para convertir las imágenes de entrada en una escena 3D se necesitan dos pasos. Primero, la app calcula mediante un proceso llamado *Structure-from-Motion (SfM)* dónde estaba la cámara en cada foto. Como subproducto, se genera una nube de puntos gruesa de la escena. Después comienza el entrenamiento Gaussian-Splatting propiamente dicho: partiendo de esa nube gruesa, los millones de «nubecitas» 3D se distribuyen, hacen crecer, refinan y reajustan en posición y color de forma gradual hasta que, desde todos los ángulos de visión de entrada, dan la imagen correspondiente.

No tienes por qué saber nada de esto para usar RadianceKit. El modo principiante oculta por completo estos pasos. Pero si quieres entender qué significan los números de diagnóstico del Modo Experto (iteración, loss, Gaussianos, SSIM ...) o por qué algunas escenas quedan más bonitas que otras, los capítulos posteriores del manual te darán las respuestas.

CAPÍTULO

Capítulo 1 — Barra de menús

La barra de menús de RadianceKit organiza todas las funciones que no se encuentran directamente en la ventana principal o el Inspector. Son sobre todo acciones que afectan a toda la escena (Open, Save, New Project), controlan el entrenamiento (Start, Pause, Continue), manejan la vista 3D (auto-rotación, captura de pantalla, color de fondo) y disparan exportaciones a varios formatos 3D y multimedia. Además, hay accesos a todas las ventanas auxiliares (User Guide, Pareto Dashboard, Holdout Analysis, BayesOpt Console).

Los atajos de teclado se enumeran a la derecha de cada entrada de menú. Convenciones: `⌘` es la tecla Command (tecla de Apple), `⇧` es Shift, `⌥` es Option (Alt) y `⌞` es Control. Ejemplo: `⇧⌘T` significa Shift+Command+T. Todos los atajos documentados aquí también se enumeran en una ventana general dedicada en `Help → Keyboard Shortcuts` (`⌘/`).

Las siguientes 42 entradas están documentadas en orden de inventario (M1–M42), agrupadas por el menú de nivel superior al que pertenecen. Todas las entradas se han verificado contra el estado actual del código (líneas 175–477). No se han eliminado ni reemplazado entradas respecto al inventario; el sistema `NSUndoManager` proporciona una nueva entrada del menú Edit (`Cmd-Z` para «Remove Image») y por eso no aparece en el código de `RadianceKitApp` (véase la nota al final del capítulo).

Menú File

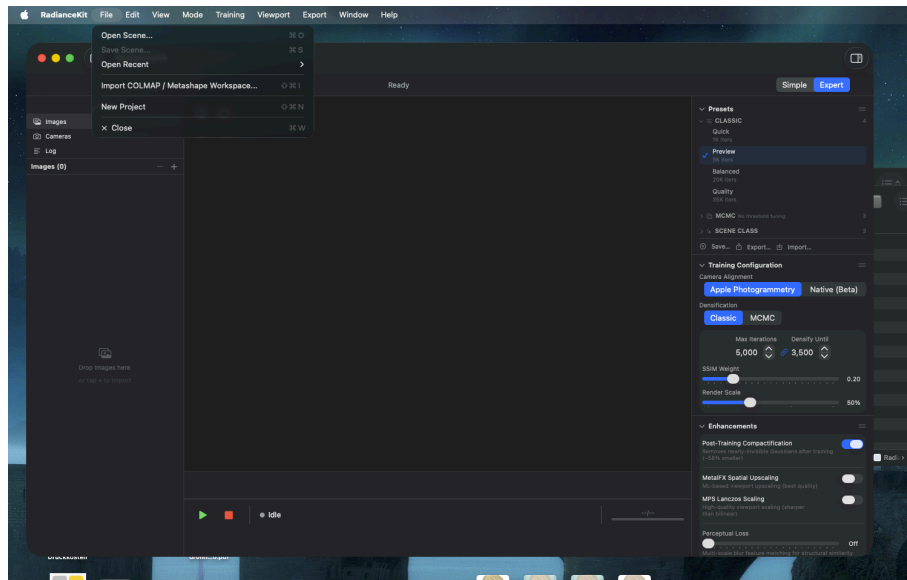


Figura 1: Menú File expandido — entradas M1 a M6

El menú File reemplaza la entrada estándar «New Window» de Apple por acciones específicas del proyecto. Cubre la carga/guardado de escenas, una lista dinámica de Recientes, la importación de workspace y un reinicio total a un estado vacío.

M1 File > Open Scene...

📍 DÓNDE

Barra de menús → File → Open Scene... (⌘O).

🔧 TÉCNICO

Abre un diálogo de archivo para los formatos paquete RadianceScene, .ply, .splat y .spz. Selección única, puede mostrar tanto archivos como directorios (para el formato de paquete). Tras una selección correcta, la ruta se añade a la lista de Recientes y la escena se carga de forma asíncrona — la anterior se reemplaza y la pipeline de entrenamiento se inicializa con el estado cargado. Los archivos PLY/SPZ/Splat se leen mediante sus respectivos cargadores de formato; el paquete .radiancecene es un directorio con manifest, snapshot de la nube y resultado de SfM.

🗨️ EN POCAS PALABRAS

Así recargas en la app una escena ya entrenada. Funciona con el formato propio de RadianceKit y con los formatos estándar PLY, SPLAT y SPZ que producen otros programas de splatting. Úsalo cuando, p. ej., hayas entrenado una escena durante la noche y quieras seguir o exportar al día siguiente. Al abrir, el estado actual de la ventana principal se reemplaza — así que guarda antes si la escena actual aún te importa. La ruta aterriza automáticamente en «Open Recent» (M3) para que vuelvas más rápido la próxima vez.

M2 File > Save Scene... **DÓNDE**

Barra de menús → File → Save Scene... (#S).

 **TÉCNICO**

Abre un diálogo de guardar archivo con el tipo de contenido paquete `RadianceScene` y un nombre de archivo precargado `scene.radiancecene`. Escribe un paquete de directorio con `manifest.json`, la nube Gaussiana serializada (snapshot PLY) y un volcado del resultado de SfM, de modo que el entrenamiento continuado también funcione tras reabrir. La entrada está deshabilitada mientras no existan aún Gaussianos. No guarda en la ruta de logs de entrenamiento, sino donde apunte el diálogo de guardar — típicamente en `~/Documents/`.

 **EN POCAS PALABRAS**

Guarda tu escena actual como archivo (más exactamente: como una carpeta paquete que parece un archivo). Solo después podrás reabrirla más tarde con «Open Scene...» (M1). El paquete contiene tanto la nube Gaussiana como el resultado de SfM, así que también puedes añadir entrenamiento continuado (M12–M14) después. Mientras no hayas terminado un entrenamiento, la entrada aparece atenuada. El nombre predeterminado es `scene.radiancecene` — pero en el diálogo de guardar puedes ponerle tu propio nombre.

M3 File > Open Recent > [nombres de escena] **DÓNDE**

Barra de menús → File → Open Recent → (lista).

 **TÉCNICO**

Submenú dinámico generado a partir de una lista de rutas abiertas recientemente (guardadas en los ajustes). Cada entrada de la lista está etiquetada con el nombre de archivo y se carga al hacer clic. Si la lista está vacía, aparece en su lugar la etiqueta deshabilitada «No Recent Scenes». Al estilo Apple, la lista contiene las N escenas abiertas más recientemente — el límite se aplica al escribir en los ajustes, no en el constructor del menú.

 **EN POCAS PALABRAS**

Aquí ves las escenas abiertas más recientemente y puedes saltar a una con un clic, sin pasar por el diálogo de archivo. Si acabas de empezar, la lista está vacía y atenuada en el menú. Cada escena que abres mediante «Open Scene...» (M1) aterriza automáticamente en esta lista. Si la lista llega a estar demasiado llena o quieres limpiarla por razones de privacidad, usa «Clear Recent» (M4).

M4 File > Open Recent > Clear Recent **DÓNDE**

Barra de menús → File → Open Recent → Clear Recent.

 **TÉCNICO**

Borra la lista de Recientes en los ajustes. Surte efecto de inmediato, sin diálogo de confirmación. La entrada solo aparece en el submenú cuando hay efectivamente entradas en la lista de Recientes (queda bajo un separador después de las rutas).

 **EN POCAS PALABRAS**

Borra la lista de escenas abiertas recientemente. Útil cuando has estado jugueteando con un dataset de prueba y ya no quieres ver las rutas. Los archivos de escena en sí no se eliminan — solo el enlace en el menú. La acción surte efecto de inmediato, sin confirmación; después, el submenú muestra «No Recent Scenes». La entrada solo se ve si hay efectivamente escenas en la lista — cuando está vacía, es invisible.

M5 File > Import COLMAP / Metashape Workspace... **DÓNDE**

Barra de menús → File → Import COLMAP / Metashape Workspace... (⇧⌘I).

 **TÉCNICO**

Abre un selector de carpeta. Espera una carpeta con el layout de workspace COLMAP (p. ej., `sparse/0/cameras.{bin,txt}` más `images/`). Tras la selección, se ejecuta una comprobación previa del workspace — detecta los tres layouts (`sparse/0/`, `sparse/`, raíz) y si la reconstrucción es binaria (`cameras.bin`) o en formato de texto ETH3D (`cameras.txt`). Si todo está bien, se importa el workspace; en caso contrario, solo aparece un aviso en el registro de la app. Véase también el Capítulo 9 «Backends de SfM», Q6 para la lógica completa de la pipeline.

 **EN POCAS PALABRAS**

Si usas Metashape, COLMAP, RealityCapture o un software similar para la reconstrucción de cámara y tienes una exportación, carga la carpeta aquí. RadianceKit se salta entonces la etapa de SfM y empieza a entrenar directamente — lo que ahorra horas en escenas grandes. Arrastrar y soltar sobre la ventana principal funciona igual. Se espera una carpeta con el layout COLMAP (es decir, `sparse/0/` con `cameras.*` más una carpeta `images/`). Más información sobre los layouts y flujos soportados en el Capítulo 9 «Backends de SfM».

M6 File > New Project

DÓNDE

Barra de menús → File → New Project (⇧⌘N).

TÉCNICO

Comprueba si hay trabajo sin guardar. En ese caso, aparece un diálogo de confirmación antes de perder nada. Si no hay nada que guardar, el reinicio se ejecuta de inmediato — limpia las imágenes importadas, el resultado de SfM, la nube Gaussiana, el estado del entrenamiento y todos los indicadores de UI dependientes. Nota: la biblioteca de ajustes preestablecidos creada por el usuario se conserva porque vive en los ajustes de la app y no en el estado del proyecto.

EN POCAS PALABRAS

Restablece todo a un inicio vacío — como si acabaras de abrir la app recién. Si aún tienes trabajo sin guardar, la app pregunta primero. Úsalo cuando quieras empezar de cero con una escena completamente distinta. Las imágenes importadas, el resultado de SfM, la nube Gaussiana y el estado de entrenamiento se limpian por completo. Tus propios ajustes preestablecidos se conservan, en cambio, porque viven en los ajustes de la app y no pertenecen a la escena.

Menú Mode

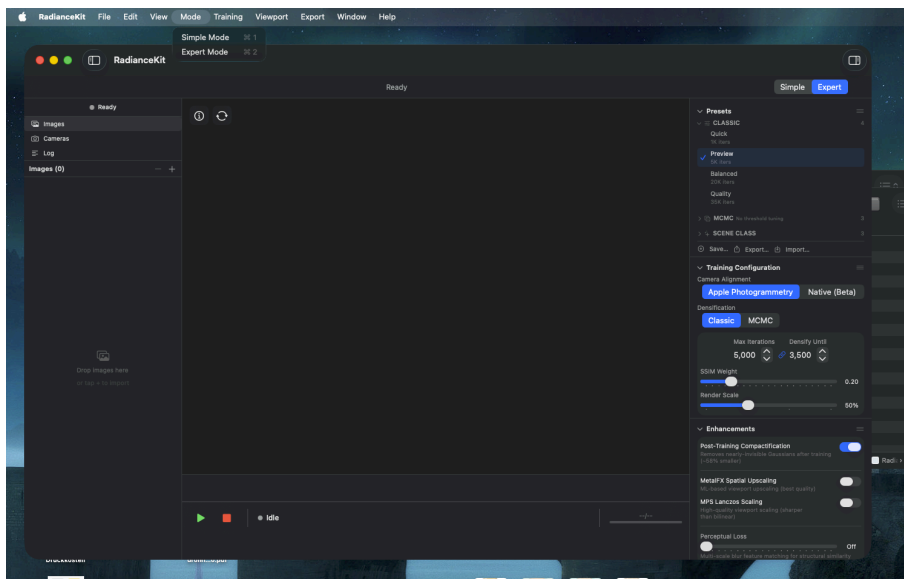


Figura 2: Menú Mode con conmutadores Simple y Expert Mode

Dos conmutadores simples entre el Simple Mode guiado (tipo asistente, 4 pasos) y el Expert Mode completo (layout clásico del Inspector con todos los controles).

M7 Mode > Simple Mode **DÓNDE**

Barra de menús → Mode → Simple Mode (⌘1).

 **TÉCNICO**

Cambia el estado de la app a Simple Mode. El área principal de la app muestra entonces el flujo guiado en lugar del layout experto. El estado de modo se guarda en los ajustes (véase S1 «Default Mode» en el Capítulo 3 Settings).

 **EN POCAS PALABRAS**

Cambia a la variante paso a paso en la que la app te guía por importar, procesar, previsualizar y exportar. Recomendado si estás empezando o necesitas un resultado rápido. La mayoría de los controles de detalle están ocultos — trabajas con predeterminados sensatos. Si más tarde quieres profundizar, cambia simplemente a Expert Mode (M8). Cuál es el modo activo al iniciar la app se puede configurar en los ajustes (Capítulo 3, S1).

M8 Mode > Expert Mode **DÓNDE**

Barra de menús → Mode → Expert Mode (⌘2).

 **TÉCNICO**

Cambia el estado de la app a Expert Mode. Esto despliega el layout completo del Inspector con todas las secciones (Presets, TrainingConfig, Enhancements, Metrics, LossChart, ProjectNavigator). En Expert Mode son accesibles todos los parámetros de entrenamiento, el selector COLMAP, los conmutadores mid-compact y los diagnósticos. La previsualización en vivo también funciona solo en este modo.

 **EN POCAS PALABRAS**

Cambia a la vista completa con todos los controles. Aquí ves gráficas de loss en tiempo real, puedes afinar todos los parámetros y gestionar en paralelo varias configuraciones de comparación mediante ajustes preestablecidos. Recomendado si quieres entender qué está haciendo el entrenamiento internamente o si quieres experimentar de manera dirigida. La previsualización en vivo, el selector COLMAP y los diagnósticos también son accesibles solo aquí. Si te sientes abrumado, vuelve a Simple Mode mediante M7 — tu escena se conserva en el proceso.

Menú Training

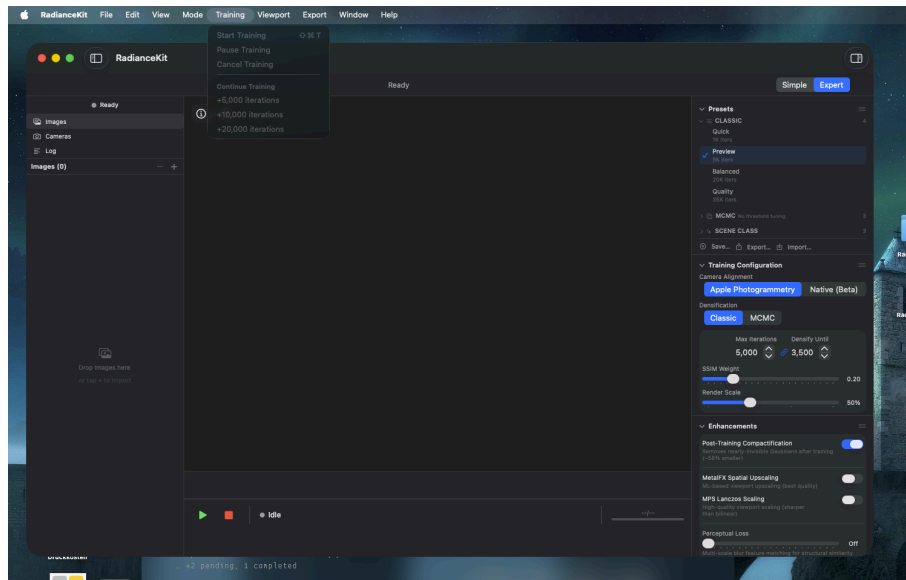


Figura 3: Menú Training con el submenú Continue — entradas M9 a M14

Cuatro acciones en torno al run de entrenamiento: iniciar, pausar, cancelar y extender por un número fijo de iteraciones. Las tres entradas Continue tienen bloqueo IAP (no son clicables en la versión de prueba gratuita).

M9 Training > Start Training

📍 DÓNDE

Barra de menús → Training → Start Training (⇧⌘T).

🔧 TÉCNICO

Inicia la pipeline de entrenamiento de forma asíncrona. Requisito: existe un resultado de SfM y no hay otra pipeline en curso. Ambas condiciones bloquean la entrada si no se cumplen. Al iniciar, se leen los valores de configuración actuales, se crea un nuevo log JSONL en `~/Documents/RadianceKit/Logs/training_YYYY-MM-DD_HHmms.jsonl` y, según la elección de estrategia, se toma la vía clásica o la MCMC. El estado de entrenamiento pasa de «idle» a «training».

🗨️ EN POCAS PALABRAS

Pulsa el gran botón verde — una vez hayas importado las fotos y hecha la reconstrucción de cámara, arranca el entrenamiento Gaussian-Splatting propiamente dicho. Deja correr la app; según el ajuste preestablecido, entre 1 minuto (Quick) y varias horas (MCMC Quality). La entrada se mantiene atenuada mientras no haya aún un resultado de SfM o haya otra pipeline en curso. Cada run escribe además un log en `~/Documents/RadianceKit/Logs/`, que luego puedes analizar mediante el Pareto Dashboard (M40).

M10 Training > Pause Training **DÓNDE**

Barra de menús → Training → Pause Training.

 **TÉCNICO**

Pausa el entrenamiento en curso. Solo está habilitado cuando el estado de entrenamiento es «training». Pausar detiene el bucle de iteración en el siguiente punto seguro de sincronización, mantiene todo el estado GPU (buffers de Gaussianos, momentos del optimizador, posición del scheduler) y cambia a «paused». Reanudar ocurre pulsando de nuevo (el título de la entrada es estático — pero la app alterna entre Pause/Resume en la lógica real). Los entrenamientos pausados no sobreviven al cierre de la app; en ese caso, guarda la escena y extiéndela más tarde mediante una entrada Continue Training (M12–M14).

 **EN POCAS PALABRAS**

Detiene brevemente el entrenamiento sin perder progreso. Útil cuando necesitas el ordenador para algo más importante por un momento. Haz clic de nuevo para reanudar. No funciona entre reinicios de la app — si quieres continuar realmente más tarde, termina el entrenamiento con Cancel (M11), guarda la escena con Save Scene (M2) y luego usa Continue Training (M12–M14). Durante la pausa la GPU está totalmente inactiva; la memoria, sin embargo, permanece reservada.

M11 Training > Cancel Training **DÓNDE**

Barra de menús → Training → Cancel Training.

 **TÉCNICO**

Cancela el entrenamiento en curso. Activo cuando el estado de entrenamiento no es «idle». Establece la bandera de cancelación en el motor de entrenamiento, que termina limpiamente el bucle de iteración en el siguiente punto de sincronización, escribe la entrada de resumen final en el log JSONL y restablece el estado a «idle». La nube entrenada hasta ese momento se conserva (se puede guardar o exportar) pero queda marcada como «cancelled».

 **EN POCAS PALABRAS**

Cancela definitivamente el entrenamiento en curso. El estado actual se conserva — así que si tras unas miles de iteraciones ya tienes un resultado presentable, aún puedes exportarlo. Si solo quieres interrumpir brevemente, usa Pause (M10) en su lugar. En el log de entrenamiento, el run se marca como «cancelled» y el valor final de loss aún se escribe. Una escena cancelada también se puede continuar mediante Continue Training (M12–M14) más tarde, siempre que entretanto no se haya cerrado la app.

M12 Training > Continue Training > +5.000 iteraciones **DÓNDE**

Barra de menús → Training → Continue Training → +5,000 iterations.

 **TÉCNICO**

Continúa el entrenamiento durante 5.000 iteraciones más. Activo cuando un entrenamiento completado es reanudable y la versión completa está desbloqueada. La reanudabilidad se da cuando existe un entrenamiento completado y todo el estado del optimizador sigue en memoria. Al continuar, los momentos de Adam y el scheduler de LR se conservan, así que la continuación se comporta como un run continuo de 25K/45K/60K en lugar de un reinicio. El log JSONL recibe una nueva entrada de config con el setup incremental. Solo disponible en la versión completa.

 **EN POCAS PALABRAS**

Añade otros 5.000 pasos de entrenamiento. Úsalo cuando el resultado tras el primer run esté cerca, pero aún no del todo nítido. Solo funciona en la versión completa de pago. A diferencia de un run totalmente nuevo, el estado del optimizador se conserva, así que la continuación se siente como un run continuo. Si necesitas más de 5.000 pasos, ve directo a M13 (+10.000) o M14 (+20.000).

M13 Training > Continue Training > +10.000 iteraciones **DÓNDE**

Barra de menús → Training → Continue Training → +10,000 iterations.

 **TÉCNICO**

Idéntico a M12, pero con 10.000 iteraciones adicionales. Mismos requisitos, misma vía del scheduler de LR. Recomendado cuando el entrenamiento inicial se ejecutó con un ajuste preestablecido intermedio y quieres ver una mejora significativa sin reiniciar el run desde cero.

 **EN POCAS PALABRAS**

Extiende el entrenamiento en 10.000 pasos — el del medio de los tres valores Continue disponibles. Buena elección cuando el primer run estuvo bien pero aún quieres mejorar claramente. Como en M12 y M14, el schedule de la tasa de aprendizaje continúa de forma fluida en lugar de reiniciarse. Solo disponible en la versión completa.

M14 Training > Continue Training > +20.000 iteraciones**DÓNDE**

Barra de menús → Training → Continue Training → +20,000 iterations.

TÉCNICO

Idéntico a M12 / M13, pero con 20.000 iteraciones adicionales. El mayor salto Continue preestablecido. En entrenamientos MCMC suele ser lo que marca la diferencia entre «vale» y «calidad de benchmark»; en Classic, más allá de 35–40K la experiencia muestra que se añade poco.

EN POCAS PALABRAS

Añade 20.000 pasos de entrenamiento adicionales, el valor máximo de Continue. Úsalo cuando de verdad quieras expresar el último ápice de calidad. En entrenamientos clásicos más allá de 40.000 pasos esto a menudo no aporta mucho más — mientras que en MCMC suele valer la pena, porque la convergencia se establece más despacio allí. Espera un tiempo de ejecución extra significativo según la escena. Como M12 y M13, esta entrada también solo está disponible en la versión completa.

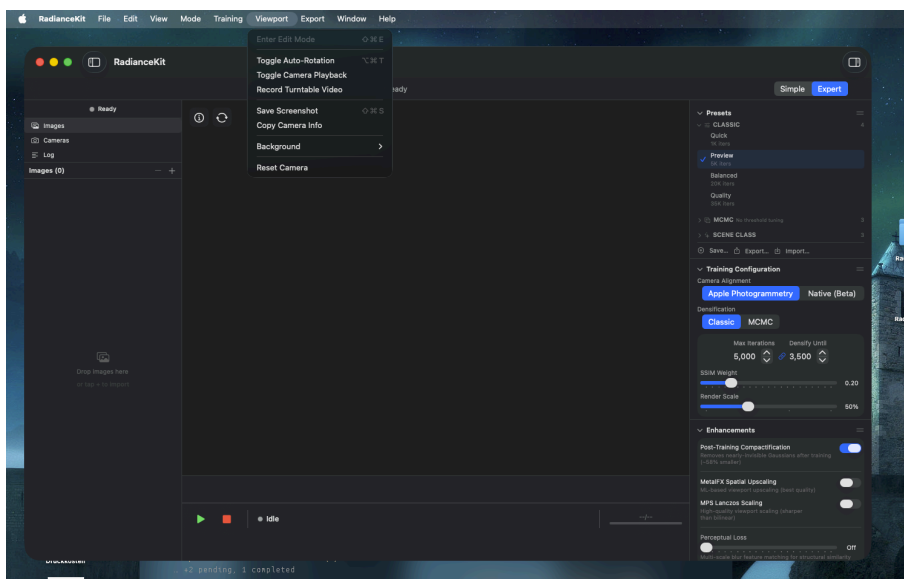
Menú Viewport

Figura 4: Menú Viewport con Edit Mode, controles de cámara y submenú Background

Controla la vista 3D: Edit Mode para selección y limpieza de Gaussianos, controles de cámara (auto-rotación, reproducción, grabación), captura de pantalla, color de fondo y reset.

M15 Viewport > Enter/Exit Edit Mode **DÓNDE**

Barra de menús → Viewport → Enter Edit Mode (o «Exit Edit Mode», según el estado). ⌘E.

 **TÉCNICO**

El título de la entrada es dinámico y muestra «Exit Edit Mode» o «Enter Edit Mode» según el estado. Al pulsar, se conmuta el Edit Mode en el renderer de la vista. Al salir del Edit Mode, la selección actual también se restablece. El Edit Mode activa la selección por clic sobre Gaussianos, la selección por caja y el borrado de Gaussianos marcados (véase el área de editor de la UI). Deshabilitado mientras no haya un renderer de vista adjunto.

 **EN POCAS PALABRAS**

Alterna entre la vista 3D normal y un modo de edición en el que puedes marcar y eliminar Gaussianos individuales (p. ej., floaters o outliers en el fondo). Al salir, la selección se restablece automáticamente. La entrada se mantiene atenuada mientras no haya aún una escena visible en la vista. La etiqueta cambia entre «Enter Edit Mode» y «Exit Edit Mode» según el estado — así siempre sabes en qué modo estás.

M16 Viewport > Toggle Auto-Rotation **DÓNDE**

Barra de menús → Viewport → Toggle Auto-Rotation (⌘T).

 **TÉCNICO**

Activa o desactiva la rotación continua de la cámara de la vista alrededor de un eje vertical a través del centro de la escena. El eje y la velocidad provienen de la configuración del control de cámara. La auto-rotación es un efecto puramente de vista y no influye ni en el entrenamiento ni en la grabación — si usas en paralelo la grabadora de vídeo de plato giratorio (M18), la auto-rotación entrega exactamente la trayectoria que la grabadora captura.

 **EN POCAS PALABRAS**

Rota continuamente la cámara despacio alrededor de tu escena para verla desde todos los lados sin arrastrar con el ratón. Haz clic de nuevo para detener la rotación. Útil al revisar escenas ya entrenadas o como animación de fondo para una demostración en vivo. Si grabas un vídeo en paralelo (M18), la auto-rotación entrega exactamente el movimiento que la grabadora captura.

M17 Viewport > Toggle Camera Playback **DÓNDE**

Barra de menús → Viewport → Toggle Camera Playback.

 **TÉCNICO**

Alterna la reproducción de la trayectoria de cámara. Si existe una trayectoria de cámara grabada (p. ej., de una grabación previa o porque se ha cargado un `transforms.json`), la trayectoria se reproduce — la cámara de la vista ya no se mueve en respuesta al ratón/trackpad, sino que reproduce la trayectoria fotograma a fotograma. Pulsar de nuevo pausa la reproducción.

 **EN POCAS PALABRAS**

Reproduce una trayectoria de cámara grabada o importada previamente. Así puedes retrazar la trayectoria original con la que se capturó la escena, o comprobar un movimiento orbital planificado antes de la exportación de vídeo. Mientras la reproducción está activa, las entradas de ratón y trackpad están deshabilitadas — la cámara sigue estrictamente la trayectoria. Hacer clic de nuevo pausa la reproducción. Si no has cargado ni grabado una trayectoria, no pasa nada.

M18 Viewport > Record Turntable Video **DÓNDE**

Barra de menús → Viewport → Record Turntable Video.

 **TÉCNICO**

Alterna la grabación de la vista. En la primera pulsación arranca una grabación de fotogramas a una ruta temporal; en la segunda, la grabación se termina, se codifica y se escribe en una ruta MP4 (la ruta se pide mediante un diálogo de guardar). A diferencia de Export → Media → Orbit Video (M31), que produce una trayectoria 360° fija con una duración configurable, la grabadora de plato giratorio captura *en vivo* lo que ves en la vista — así también puedes grabar un movimiento de cámara manual.

 **EN POCAS PALABRAS**

Graba un vídeo directamente en la vista. Da igual si la cámara rota automáticamente o la mueves tú con el ratón — todo lo que ves se guarda en un archivo MP4. A diferencia de la exportación «Orbit Video» (M31), aquí controlas tú el movimiento de cámara. El primer clic arranca la grabación, el segundo la termina y pregunta dónde guardar. Útil cuando, p. ej., quieres mostrar un travelling concreto de detalle que no sería posible con el movimiento orbital rígido.

M19 Viewport > Save Screenshot **DÓNDE**

Barra de menús → Viewport → Save Screenshot (⇧⌘S).

 **TÉCNICO**

Captura un único fotograma de la vista a resolución de renderizado completa (es decir, no el layout de píxeles de la ventana, sino el contenido completo del render target) como archivo PNG. La ruta se pide mediante un diálogo de guardar. El color de fondo (M21–M23) queda fijado. Los ajustes de upscaling MetalFX/MPS de Enhancements (véase I27/I28) surten efecto si están activos — la captura muestra entonces la salida upscaled.

 **EN POCAS PALABRAS**

Guarda una instantánea de tu vista 3D actual como imagen PNG. Útil para material de marketing o una comparación rápida. Nota: el fondo es parte de la imagen — si necesitas transparencia, exporta en su lugar un archivo de escena. La resolución corresponde al render target interno, no al tamaño de ventana — por eso la imagen suele ser más nítida de lo que se ve en la ventana. También entran los ajustes de upscaling cualesquiera (Inspector → Enhancements).

M20 Viewport > Copy Camera Info **DÓNDE**

Barra de menús → Viewport → Copy Camera Info.

 **TÉCNICO**

Lee la pose actual de la cámara de la vista (posición, punto look-at, vector up) y los valores de FOV del control de cámara y los escribe como texto multilínea al portapapeles. El formato es legible para humanos (etiqueta = valor por línea), no JSON. Útil para reproducir una vista concreta con fines de depuración o para compartir con soporte.

 **EN POCAS PALABRAS**

Copia la posición y dirección de visión actuales de la cámara como texto al portapapeles. Si quieres, p. ej., mostrar a un compañero de desarrollo dónde un punto de la escena se ve raro, solo tienes que pegar el texto en un correo o ventana de chat. El formato es legible para humanos (una línea por valor), no JSON. Pensado sobre todo para reportes de bugs o solicitudes de soporte.

M21 Viewport > Background > Dark Gray **DÓNDE**

Barra de menús → Viewport → Background → Dark Gray.

 **TÉCNICO**

Establece el color de fondo de la vista a un gris oscuro (RGB 0,1/0,1/0,1). El renderer usa este color como fondo contra el que se componen los Gaussianos. El color predeterminado al iniciar la app se controla mediante la opción de ajustes S3 «Default Viewport Background».

 **EN POCAS PALABRAS**

Tiñe el fondo de la vista 3D de gris oscuro. Elección estándar para la mayoría de las escenas — ofrece buen contraste tanto para Gaussianos claros como oscuros sin que el ojo se ancle en un área puramente negra o blanca. El color también queda fijado en capturas (M19) y vídeos orbitales (M31). Si Dark Gray te resulta poco espectacular, prueba también Black (M22) o White (M23) para comparar. Cuál es el color activo al iniciar la app se puede configurar en los ajustes (S3).

M22 Viewport > Background > Black **DÓNDE**

Barra de menús → Viewport → Background → Black.

 **TÉCNICO**

Establece el color de fondo de la vista a negro puro (RGB 0/0/0). Ayuda cuando la escena tiene muchos floaters brillantes y quieres identificarlos, o para material de marketing con un look-and-feel oscuro.

 **EN POCAS PALABRAS**

Fondo negro. Bueno para escenas muy luminosas o cuando quieres mirar en Edit Mode y buscar pequeños Gaussianos brillantes (floaters) que se pierden en el gris. También ideal para material de marketing con un look oscuro y dramático. El color queda fijado en capturas y vídeos orbitales — si necesitas transparencia para un composite posterior, el negro es la peor elección. Para floaters oscuros, cambia al revés a White (M23).

M23 Viewport > Background > White **DÓNDE**

Barra de menús → Viewport → Background → White.

 **TÉCNICO**

Establece el color de fondo de la vista a blanco puro (RGB 1/1/1). Útil cuando la escena tiene contenido predominantemente oscuro y quieres ver floaters oscuros (ruido de fondo típico de exteriores).

 **EN POCAS PALABRAS**

Fondo blanco. Útil cuando el motivo se ve mejor claro sobre oscuro, o para encontrar outliers oscuros que luego puedes eliminar en Edit Mode (M15). En escenas exteriores, el blanco es a menudo más útil que el negro, porque los floaters típicos de exteriores tienden a ser oscuros. Como con las otras opciones de fondo, el color queda fijado en capturas y vídeos.

M24 Viewport > Reset Camera **DÓNDE**

Barra de menús → Viewport → Reset Camera.

 **TÉCNICO**

Restablece la cámara de la vista, abandona la vista de la cámara de entrenamiento y detiene la auto-rotación. Eso devuelve la cámara a la posición inicial (típicamente: delante de la escena, mirando ligeramente desde arriba), la auto-rotación queda apagada, y si el renderer estaba mostrando la cámara de entrenamiento (una de las poses de SfM), vuelve a la cámara libre.

 **EN POCAS PALABRAS**

Trae la cámara de la vista de vuelta a la posición inicial. Si te has perdido moviéndote o has empujado la escena fuera de cuadro — haz clic aquí una vez y vuelves a ver lo que deberías estar viendo. A la vez, apaga la auto-rotación si está activa y vuelve, desde una cámara de entrenamiento congelada, a la vista libre. Así obtienes un reinicio limpio de la vista en cualquier caso.

Menú Export

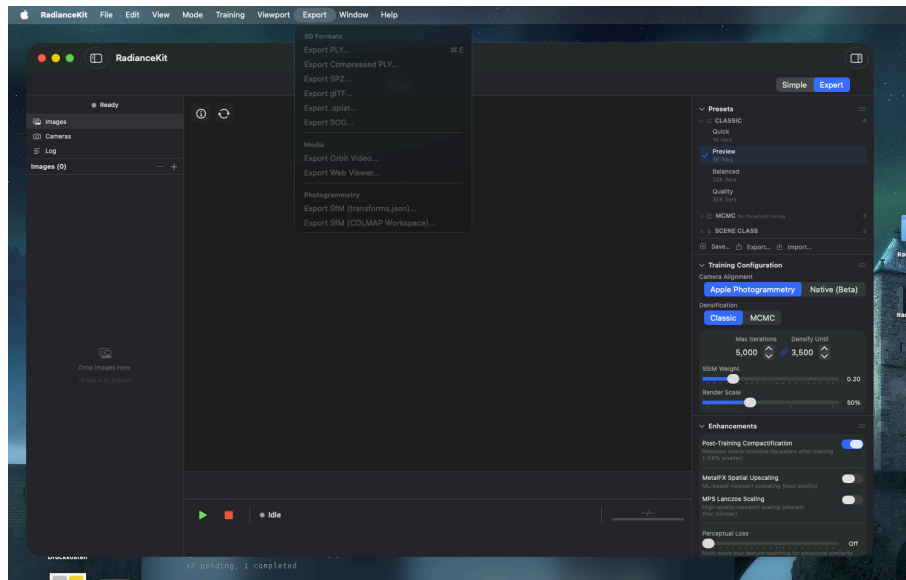


Figura 5: Menú Export con tres grupos de submenú — 3D Formats, Media y Photogrammetry

Ocho destinos de exportación más dos exportaciones de fotogrametría, agrupados en tres secciones (3D Formats, Media, Photogrammetry). Los seis primeros se construyen mediante una rutina helper común que abre un diálogo de guardar y registra la exportación en el catálogo de formato. Las entradas de fotogrametría tienen lógica individual. Todas las exportaciones de fotogrametría y algunas exportaciones 3D solo están disponibles en la versión completa.

M25 Export > 3D Formats > Export PLY...

DÓNDE

Barra de menús → Export → 3D Formats → PLY (⌘E).

TÉCNICO

Abre un diálogo de guardar con nombre de archivo predeterminado `gaussians.ply`. Al aceptar, la nube Gaussiana actual se escribe en el formato PLY ASCII/binario estandarizado — compatible con SuperSplat, PolyCam, PlayCanvas y todos los visores 3DGS habituales. Coeficientes SH completos, precisión completa (Float32 por campo). El tamaño de archivo a menudo es de varios cientos de MB con $\geq 500K$ Gaussianos.

EN POCAS PALABRAS

Guarda tu escena 3D como archivo PLY estándar. Este es el formato más universal — casi cualquier software puede cargarlo, desde SuperSplat a PolyCam o PlayCanvas. Sin embargo, los archivos se vuelven grandes, a menudo varios cientos de megabytes. Usa PLY cuando quieras seguir trabajando con calidad completa o archivar. Si quieres compartir la escena por la web, mira SPZ (M27) o Compressed PLY (M26) en su lugar — son mucho más pequeños.

M26 Export > 3D Formats > Export Compressed PLY... **DÓNDE**

Barra de menús → Export → 3D Formats → Compressed PLY.

 **TÉCNICO**

Escribe la nube Gaussiana en el formato PLY comprimido con cuantización propia de los campos de posición, escala, rotación y SH. Archivos 5–10x más pequeños que el PLY sin comprimir (M25) con pérdida visual mínima. Compatible con SuperSplat (que lee el estándar PLY comprimido) y PlayCanvas. Nombre predeterminado `gaussians_compressed.ply`.

 **EN POCAS PALABRAS**

Como PLY normal, pero 5–10 veces más pequeño. La calidad se mantiene casi igual. Úsalo cuando quieras compartir el archivo online o enviarlo por correo. Funciona directamente con SuperSplat y PlayCanvas. Si tu sistema de destino necesita archivos aún más pequeños (móvil, demos en navegador), usa SPZ (M27) en su lugar — está comprimido aún más agresivamente. Para calidad de edición plena, usa el PLY sin comprimir (M25).

M27 Export > 3D Formats > Export SPZ... **DÓNDE**

Barra de menús → Export → 3D Formats → SPZ.

 **TÉCNICO**

Escribe la nube Gaussiana en el formato SPZ — el formato de splat comprimido publicado por Niantic con cuantización agresiva (~90 % más pequeño que PLY sin comprimir). Optimizado sobre todo para visores web y apps móviles. Compatible con Niantic Splatt3R, `gsplat.js` y el visor de navegador de Niantic.

 **EN POCAS PALABRAS**

Uno de los formatos más pequeños. Cerca de 10x más pequeño que un PLY normal. Úsalo principalmente cuando quieras mostrar la escena en un navegador o verla en una app de móvil. Para máxima calidad, PLY es la mejor elección. SPZ lo desarrolló Niantic y funciona directamente con `gsplat.js`, Splatt3R y el visor web de Niantic. Por la fuerte compresión, ya no puedes seguir entrenando con archivos SPZ fácilmente — usa PLY para edición.

M28 Export > 3D Formats > Export glTF... DÓNDE

Barra de menús → Export → 3D Formats → glTF.

 TÉCNICO

Escribe un archivo `.gltf` (glTF binario) con la extensión `KHR_gaussian_splatting`. Conforme al estándar, apto para pipelines que usan engines glTF como Babylon.js o Three.js e implementan la extensión `KHR_gaussian_splatting`.

 EN POCAS PALABRAS

Guarda la escena en el formato glTF que muchos programas 3D y engines web entienden — siempre que soporten la extensión Gaussian-Splatting. Si tienes una pipeline 3D específica (p. ej., Three.js o Babylon.js) que la entiende, este es tu formato. El archivo sale como `.gltf` binario — un único paquete que contiene todo. Para flujos de splatting clásicos, PLY o SPZ suelen ser la mejor elección, ya que más herramientas los entienden directamente.

M29 Export > 3D Formats > Export .splat... DÓNDE

Barra de menús → Export → 3D Formats → .splat.

 TÉCNICO

Escribe el formato `.splat` de Antimatter15 — tamaño fijo de 32 bytes por Gaussiano (posición como $3 \times \text{Float32}$, escala como $3 \times \text{Float32}$, rotación como cuaternión normalizado $4 \times \text{Uint8}$, RGB+opacidad como $4 \times \text{Uint8}$). Sin coeficientes SH superiores al DC. El archivo más pequeño con compatibilidad directa de navegador. Para `gsplat.js` y el visor de demo online de `antimatter15`.

 EN POCAS PALABRAS

El formato más sencillo para visores web. Pequeño y mostrable inmediatamente en cualquier navegador. Pero pierde la iluminación de detalle (los coeficientes SH superiores se pierden — el splat se ve igual desde todos los ángulos en vez de reaccionar a la luz). Bueno para máximo rendimiento web, pero para fotorrealismo, SPZ o PLY es mejor. Funciona con el visor online de `antimatter15` y `gsplat.js`. Cada Gaussiano ocupa 32 bytes fijos, lo que hace al formato simple y compatible — pero a costa de la profundidad de detalle.

M30 Export > 3D Formats > Export SOG... DÓNDE

Barra de menús → Export → 3D Formats → SOG.

 TÉCNICO

Escribe la nube Gaussiana en el formato SOG. SOG («Self-Organizing Gaussian») es el formato PlayCanvas con layout de atlas de texturas y compresión WebP de los datos cuantizados. Escala con un ratio de tamaño 15–20× mejor que PLY. La exportación llama internamente a `cwebp` como herramienta externa — por lo que en la variante sandbox (App Store) puede estar restringido.

 EN POCAS PALABRAS

Formato muy pequeño para flujos PlayCanvas. Cerca de 15–20 veces más pequeño que PLY porque los datos se empaquetan en un layout de atlas de texturas y se comprimen en WebP. Si no tienes un flujo PlayCanvas, SPZ o Compressed PLY suelen ser la mejor elección. La exportación llama internamente a `cwebp` como herramienta externa — en la versión de App Store (sandbox), este paso puede estar restringido.

M31 Export > Media > Export Orbit Video... DÓNDE

Barra de menús → Export → Media → Orbit Video.

 TÉCNICO

Renderiza una órbita 360° alrededor del centro de la escena y la codifica como MP4 (H.264) o MOV (HEVC, según el predeterminado del sistema). A diferencia de M18 (grabación en vivo), la trayectoria aquí es fija — la duración se elige en los ajustes o en el paso de exportación de Simple Mode.

 EN POCAS PALABRAS

Produce automáticamente un vídeo de giro alrededor de tu escena. No hace falta mover nada a mano. Bueno para redes sociales o una demo rápida. Si quieres controlar tú la cámara, usa Record Turntable Video (M18) en su lugar. La trayectoria es fija: una órbita completa de 360° alrededor del centro de la escena; eliges la duración en los ajustes o en el paso de exportación de Simple Mode. El vídeo sale como MP4 H.264 o MOV HEVC según el sistema.

M32 Export > Media > Export Web Viewer... DÓNDE

Barra de menús → Export → Media → Web Viewer.

 TÉCNICO

Empaqueta un visor HTML autónomo (basado en gsplat.js) más los datos Gaussianos codificados en Base64 en un único archivo `.html`. Este archivo funciona offline en cualquier navegador moderno — sin dependencias de servidor, sin URLs externas. El tamaño de archivo es un factor de aprox. 1,3 mayor que la variante SPZ (por el overhead de Base64).

 EN POCAS PALABRAS

Guarda tu escena como una página web autoarrancable. Doble clic en el archivo HTML → el navegador se abre → escena 3D interactiva lista. Funciona sin internet, se puede enviar por correo, es la forma más sencilla de compartir el resultado con amigos o clientes. El archivo contiene el visor gsplat.js completo y los datos Gaussianos en un único documento — nada se carga desde la web. El tamaño de archivo es alrededor de un tercio mayor que una exportación SPZ, pero a cambio el destinatario no necesita software adicional.

M33 Export > Photogrammetry > Export SfM (transforms.json)... DÓNDE

Barra de menús → Export → Photogrammetry → Export SfM (transforms.json).

 TÉCNICO

Vía de exportación dedicada (no a través de la rutina helper común), porque no se exporta una nube Gaussiana sino el resultado de SfM. Abre un diálogo de guardar con `transforms.json` como predeterminado y tipo de contenido `json`. Al aceptar, se escribe un `transforms.json` compatible con nerfstudio con intrínsecos de cámara, poses (como matriz 4×4 en convención NeRF) y rutas de frame. El texto de ayuda de la UI indica que las imágenes de entrenamiento deben copiarse en paralelo como carpeta hermana `images/`. Activo solo si existe un resultado de SfM y la versión completa está desbloqueada.

 EN POCAS PALABRAS

Si quieres usar el resultado de SfM en otro software como nerfstudio, Brush, gsplat u OpenSplat, aquí exportas las posiciones de cámara. Pon tus imágenes de entrenamiento además en una carpeta `images/` junto al archivo `transforms.json` — si no, el programa de destino no puede emparejar las imágenes. La entrada aparece atenuada mientras no exista resultado de SfM, y bloqueada en la versión de prueba gratuita. Para el flujo COLMAP-workspace, usa M34 en su lugar.

M34 Export > Photogrammetry > Export SfM (COLMAP Workspace)...

📍 DÓNDE

Barra de menús → Export → Photogrammetry → Export SfM (COLMAP Workspace).

🔧 TÉCNICO

Abre un diálogo de guardar con nombre predeterminado `colmap-workspace` (sin extensión, porque es una carpeta). Escribe un workspace COLMAP estándar con `sparse/0/cameras.bin`, `images.bin`, `points3D.bin`. Permite abrir en otras herramientas como Postshot, Nerfstudio o Meshroom una reconstrucción SfM calculada o importada en RadianceKit, o, para un re-run A/B, volverla a cargar como entrada ya calculada dentro del propio RadianceKit (mediante M5) — ahorra tiempo de cálculo. Activo solo si existe un resultado de SfM y la versión completa está desbloqueada.

🗨️ EN POCAS PALABRAS

Como M33 pero en formato COLMAP en lugar de nerfstudio. Si usas Postshot, Meshroom, Nerfstudio u otra herramienta con un flujo COLMAP, esta es tu exportación. Un efecto colateral útil: puedes recargar esta carpeta más tarde mediante M5 en RadianceKit y ahorrar el tiempo de cálculo de SfM en el siguiente run — especialmente en escenas grandes, un ahorro de horas. Como M33, solo disponible cuando existe un resultado de SfM, y bloqueada en la versión de prueba gratuita.

Menú Help

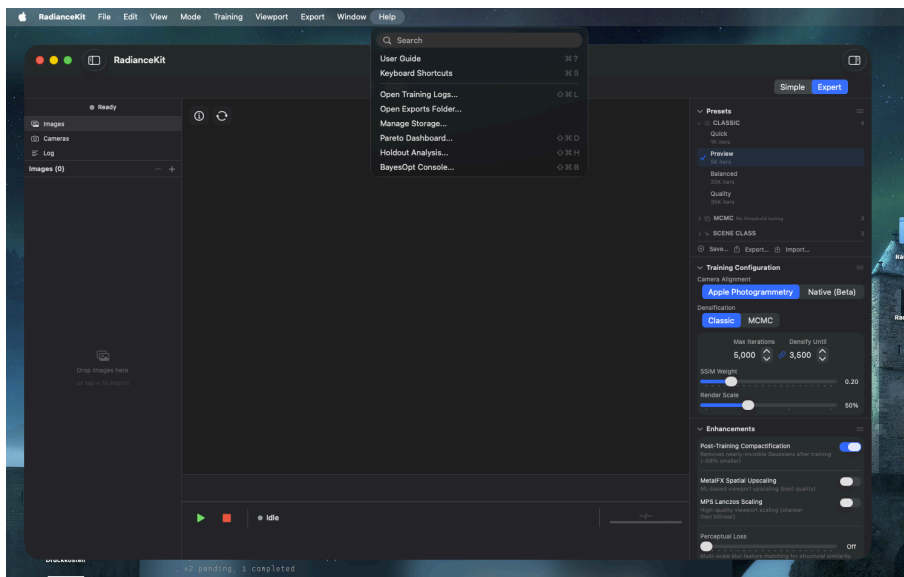


Figura 6: Menú Help con documentación, carpeta y entradas de análisis

Siete entradas: dos ventanas de documentación (User Guide, Keyboard Shortcuts), tres atajos a carpeta (Training Logs, Exports, Storage) y tres ventanas de análisis (Pareto Dashboard, Holdout Analysis, BayesOpt Console). Al estilo Apple, el menú Help aparece a la derecha del todo. El menú Help estándar se reemplaza por completo por la variante propia de RadianceKit.

M35 Help > User Guide **DÓNDE**

Barra de menús → Help → User Guide (⌘?).

 **TÉCNICO**

Abre la ventana User Guide. Muestra navegación con una barra lateral de temas y un área de detalle desplazable con tamaño predeterminado 860×640. Los contenidos están almacenados de forma estática (no parseados desde Markdown).

 **EN POCAS PALABRAS**

Abre la guía dentro de la app. Si no quieres leer todo este manual, encuentras los pasos clave directamente en el programa. La guía está construida como su propia ventana con una barra lateral de temas — así puedes saltar directamente a temas individuales. Los contenidos son más cortos que este manual y se centran en los flujos más habituales.

M36 Help > Keyboard Shortcuts **DÓNDE**

Barra de menús → Help → Keyboard Shortcuts (⌘/).

 **TÉCNICO**

Abre la ventana Keyboard Shortcuts — un layout simple de scroll con todos los atajos de la app, agrupados por menú de nivel superior. Tamaño predeterminado 440×560. Los contenidos también están almacenados de forma estática.

 **EN POCAS PALABRAS**

Abre una ventana con la lista completa de todos los atajos de teclado. Si no te acuerdas de, p. ej., qué tecla arranca el entrenamiento, mira ahí. Un resumen también está al final de este capítulo. La lista se agrupa por menú de nivel superior, así saltas rápido al área correcta. Útil cuando estás cambiando de un estilo de ratón a un estilo de teclado.

M37 Help > Open Training Logs... **DÓNDE**

Barra de menús → Help → Open Training Logs...
(⇧⌘L).

 **TÉCNICO**

Calcula la carpeta de logs como `~/Documents/RadianceKit/Logs`, la crea si es necesario y la abre en Finder. Cada run de entrenamiento escribe allí su propio archivo JSONL `training_YYYY-MM-DD_HHmss.jsonl`.

 **EN POCAS PALABRAS**

Abre en Finder la carpeta con todos los protocolos de entrenamiento previos. Si algo ha salido mal o quieres consultar exactamente cuándo convergió el entrenamiento a qué valor, lo encuentras aquí en archivos JSONL. Por run de entrenamiento se crea exactamente un archivo con marca de tiempo — también puedes alimentarlo a otras herramientas o enviarlo por correo a soporte. Si quieres un análisis gráfico, el Pareto Dashboard (M40) es el mejor punto de partida.

M38 Help > Open Exports Folder... **DÓNDE**

Barra de menús → Help → Open Exports Folder...

 **TÉCNICO**

Análogo a M37 pero con `~/Documents/RadianceKit/Exports`. Creada en el primer run de auto-test o en el primer clic; después aterrizan ahí las rutas predefinidas de todas las exportaciones de auto-test (p. ej., `autotest_<timestamp>.ply`). Las exportaciones seleccionadas manualmente mediante el diálogo de guardar NO necesariamente van aquí, sino donde el usuario las guarde — así que esta carpeta interesa sobre todo para auto-tests.

 **EN POCAS PALABRAS**

Abre la carpeta donde la app coloca sus propias exportaciones (sobre todo runs de auto-test). Si pusiste manualmente una exportación en otro sitio con el diálogo de guardar, está allí y no en esta carpeta. Útil para limpiar o comprobar cuánto espacio ocupan las exportaciones de prueba previas. Si quieres una visión general completa con logs y paquetes de escena, usa Manage Storage (M39) en su lugar.

M39 Help > Manage Storage... **DÓNDE**

Barra de menús → Help → Manage Storage...

 **TÉCNICO**

Abre el navegador de almacenamiento (véase Capítulo 4 Ventanas auxiliares, IDs W7–W12). Lista todas las escenas persistidas, logs de entrenamiento, exportaciones y cachés en la carpeta `~/Documents/RadianceKit/` con tamaño, permite reveal en Finder y mover a la papelera por entrada.

 **EN POCAS PALABRAS**

Abre un navegador de ventana que te muestra cuánto espacio en disco ocupa RadianceKit — por escena, log y exportación. Puedes eliminar elementos individuales directamente, sin tener que ir a Finder. Útil tras un uso prolongado, cuando el disco se llena — los logs y exportaciones de auto-test anteriores pueden sumar varios gigabytes. Mediante reveal en Finder siempre puedes llegar también a la vista clásica.

M40 Help > Pareto Dashboard... **DÓNDE**

Barra de menús → Help → Pareto Dashboard...
(⇧⌘D).

 **TÉCNICO**

Abre el Pareto Dashboard (véase Capítulo 4, IDs W13–W22). El dashboard carga todos los logs JSONL de entrenamiento desde `~/Documents/RadianceKit/Logs/`, los organiza por escena y ajuste preestablecido y dibuja un scatter plot de Pareto (predeterminado: loss vs Gaussianos, opcionalmente loss vs wallclock o PSNR vs iteraciones).

 **EN POCAS PALABRAS**

Abre una visión general de todos los entrenamientos previos como gráfico. Ves de un vistazo qué run entregó el mejor equilibrio de calidad y tamaño. Útil cuando quieres comparar distintos ajustes preestablecidos. Por defecto, el gráfico muestra loss frente a número de Gaussianos — pero también puedes cambiar a tiempo de wallclock o PSNR. Los datos proceden de los logs JSONL de entrenamiento (M37); cuantos más runs tengas, más significativo es el análisis.

M41 Help > Holdout Analysis... **DÓNDE**

Barra de menús → Help → Holdout Analysis...
(⇧⌘H).

 **TÉCNICO**

Abre la ventana Holdout Analysis (véase Capítulo 4, IDs W23–W29). Carga un `transforms.json`, dibuja las cámaras como un globo 3D y permite divisiones train/test fold (angulares o lineales, 2–8 folds). La salida es un `fold-assignment.json` que el entrenamiento puede usar como conjunto de test en las respectivas configuraciones de entrenamiento.

 **EN POCAS PALABRAS**

Te ayuda a dividir tus capturas de cámara en conjuntos de entrenamiento y test — así puedes medir objetivamente lo buena que es tu escena (sobre imágenes que el entrenamiento nunca vio). Más bien una herramienta de investigación y benchmark. Las cámaras se muestran como un globo 3D; puedes elegir entre 2 y 8 folds, ya sea uniformemente en ángulo o linealmente a lo largo del orden. El resultado es un pequeño archivo JSON que el entrenamiento usa después como conjunto de test.

M42 Help > BayesOpt Console... **DÓNDE**

Barra de menús → Help → BayesOpt Console...
(⇧⌘B).

 **TÉCNICO**

Abre la BayesOpt Console (véase Capítulo 4, IDs W30–W39). Carga espacios de búsqueda predefinidos (p. ej., «MCMC scale-reg + opacity-reg + ssim»), ejecuta trials de optimización bayesiana de forma asíncrona y muestra la curva de convergencia y el log de trials en vivo.

 **EN POCAS PALABRAS**

Una consola de auto-tuning integrada. En vez de probar manualmente distintos parámetros, la app puede correr sola durante la noche y al final sugerir los mejores valores para tu escena. Herramienta muy avanzada — para la mayoría de los flujos basta con un buen ajuste preestablecido (véase Capítulo 7). Eliges un espacio de búsqueda predefinido (p. ej., «MCMC scale-reg + opacity-reg + ssim») y ves la curva de convergencia en vivo, así como el log de trials. Planifica de varias horas a días según el setup.

Aviso: Cmd-Z en el menú Edit

Desde mayo de 2026, el Project Navigator en Expert Mode soporta eliminar imágenes importadas mediante el botón menos o la tecla Backspace y deshacer mediante `Cmd-Z`. Esta acción `Cmd-Z` aparece en el menú Edit de macOS (proporcionado por SwiftUI) como «Undo Remove Image» mientras una imagen eliminada sea aún recupere-

rable. Está registrada mediante el sistema conforme al estándar, no en ; por eso no hay una entrada M-ID dedicada en el inventario.

Resumen de atajos de teclado

Entrada de menú	Atajo
File > Open Scene...	⌘O
File > Save Scene...	⌘S
File > Import COLMAP / Metashape Workspace...	⇧⌘I
File > New Project	⇧⌘N
Mode > Simple Mode	⌘1
Mode > Expert Mode	⌘2
Training > Start Training	⇧⌘T
Viewport > Enter/Exit Edit Mode	⇧⌘E
Viewport > Toggle Auto-Rotation	⌘\T
Viewport > Save Screenshot	⇧⌘S
Export > 3D Formats > PLY	⌘E
Help > User Guide	⌘?
Help > Keyboard Shortcuts	⌘/
Help > Open Training Logs...	⇧⌘L
Help > Pareto Dashboard...	⇧⌘D
Help > Holdout Analysis...	⇧⌘H
Help > BayesOpt Console...	⇧⌘B

Menú Edit (provisto por el sistema, en Expert Mode con selección activa del Project Navigator):

Acción	Atajo
Undo Remove Image	⌘Z
Remove Selected Image	Backspace / Delete

CAPÍTULO

Capítulo 2 — Inspector (Vista Experto)

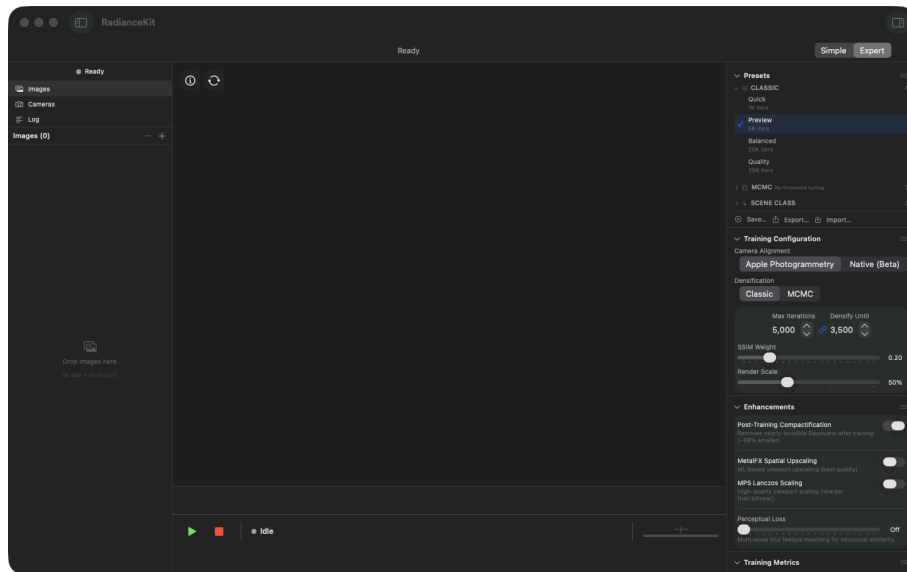


Figura 7: Modo Experto vacío — Project Navigator a la izquierda (Images 0, Cameras, Log), vista vacía en el medio, Inspector a la derecha con las secciones Presets/Training Configuration/Enhancements/ Training Metrics

Inspector vacío antes de la importación: La barra lateral izquierda muestra el contador de imágenes 0 y la pista de soltar «Drop images here / or tap + to import». El Inspector de la derecha está totalmente funcional, pero los ajustes preestablecidos son solo informativos (sin entrenamiento activo). El ajuste preestablecido predeterminado «Preview» (5K iter) está marcado. Camera Alignment está en Apple Photogrammetry, Densification en Classic, SSIM Weight en 0,20, Render Scale al 50 %. En Training Metrics se muestran estados vacíos («Start training to see live metrics») y en Loss History («Loss curve will appear during training»).

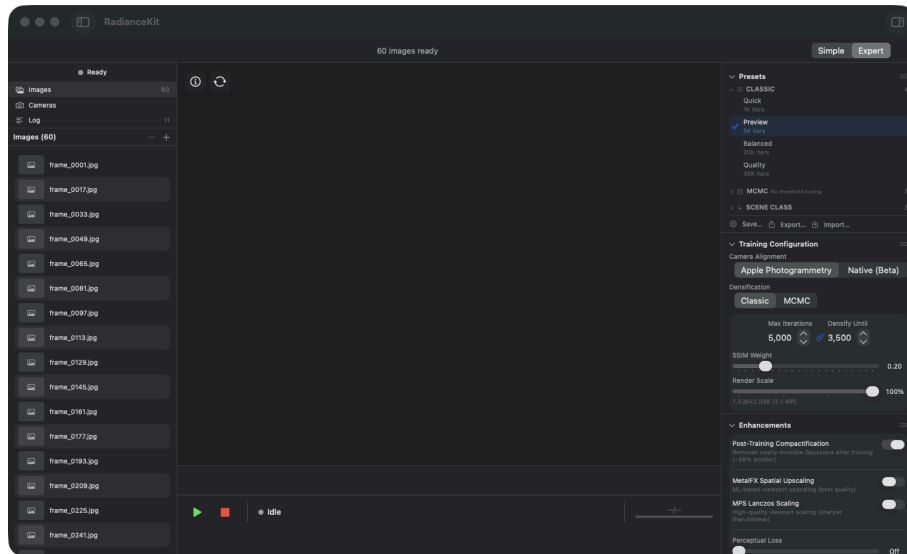


Figura 8: Inspector con 60 imágenes de un ramo cargadas — la barra lateral de imágenes muestra los primeros nombres de archivo frame_0001.jpg ff, la cabecera dice «60 images ready»

Inspector tras la importación: El estado de la cabecera dice «60 images ready». La barra lateral de imágenes lista los 60 fotogramas importados (frame_0001.jpg a frame_0945.jpg , cada 16º fotograma del dataset de 960 cámaras como subconjunto para iteraciones rápidas). La lógica auto de la escala de renderizado comprueba la resolución de imagen ($1536 \times 2048 = 3,1$ MP) y ajusta Render Scale en consecuencia. El botón Play (verde, abajo a la izquierda) está ahora activo e inicia el entrenamiento con el ajuste preestablecido activo.

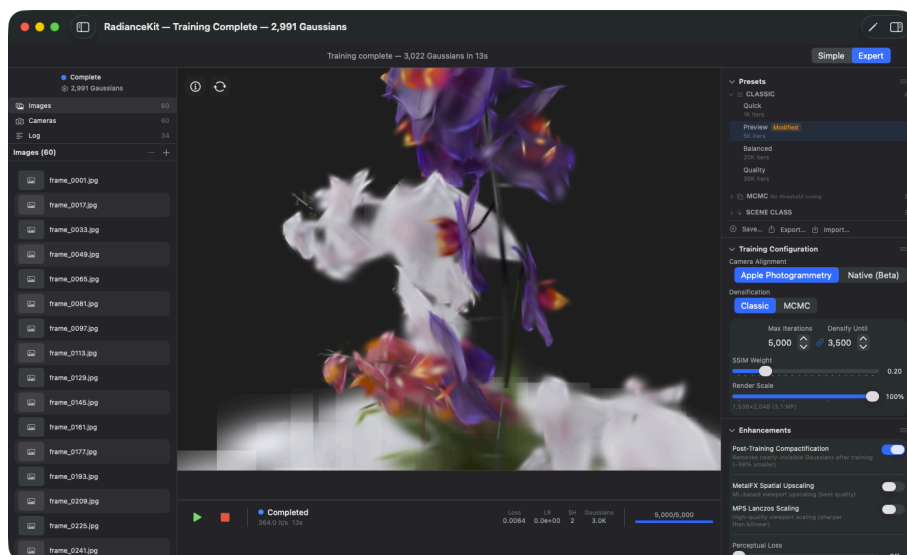


Figura 9: Inspector a mitad de entrenamiento — la vista en vivo muestra la reconstrucción del ramo, barra de métricas debajo (Loss / LR / n° de Gaussianos / iteraciones), tarjeta de ajuste preestablecido «Preview» con insignia «Modified» si se ajustaron parámetros

Inspector durante el entrenamiento: La barra de título muestra el progreso global «RadianceKit — Training NN %». La vista renderiza la reconstrucción Gaussiana en curso en tiempo real (actualizada cada 50 iteraciones — el intervalo de previsualización

en vivo se puede configurar en Settings → General → Training → Live Preview). Barra de métricas bajo la vista: Loss actual, Learning Rate, número de Gaussianos y contador de iteración (p. ej., 1.600/5.000 con el ajuste preestablecido Preview). La tarjeta de ajuste preestablecido «Preview» del Inspector recibe una insignia «Modified» en cuanto cualquier parámetro se desvía del predeterminado integrado. La barra lateral «Log» recopila los eventos de las etapas de SfM y entrenamiento.

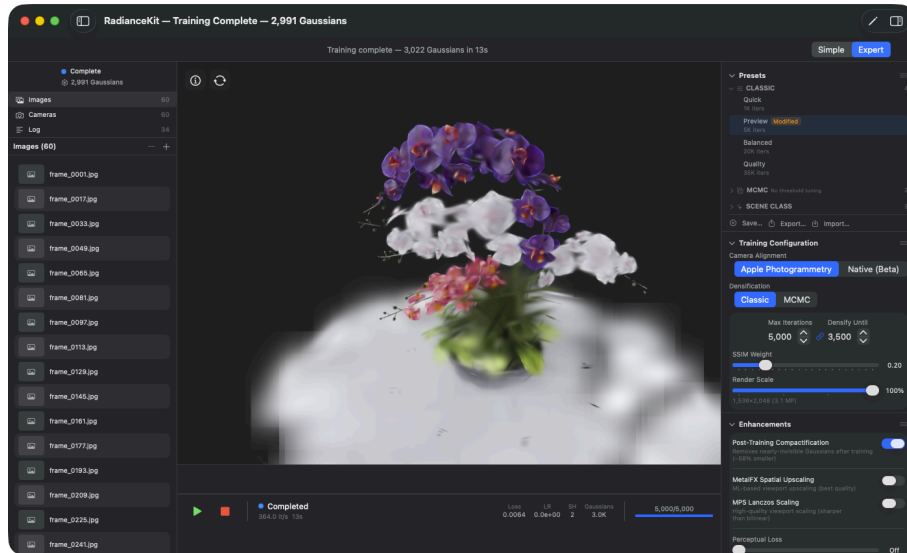


Figura 10: Inspector tras completarse el entrenamiento — la vista muestra la reconstrucción terminada del ramo (2.991 Gaussianos tras 5K iteraciones en 13s), barra de título «Training Complete — 2.991 Gaussians»

Inspector tras el entrenamiento: La barra de título muestra el número final de Gaussianos (aquí 2.991 — muy compacta, porque la escena sintética del ramo en Blender tiene geometría sencilla sobre fondo claro). La vista muestra la nube de puntos terminada — la navegación de arrastre orbital está activa (rota alrededor del centro de la escena). La sección Training Metrics está ahora rellena con valores finales, la gráfica de Loss History muestra el transcurso de las 5.000 iteraciones completas. La sección Export, abajo, está ahora activa (todos los botones de formato habilitados).

El Inspector es la barra lateral derecha en Modo Experto (⌘2). Agrupa todos los parámetros relevantes para el entrenamiento en siete secciones colapsables. El orden predeterminado de arriba abajo en el primer inicio es: Look, Presets, Training Configuration, Metrics, Loss Chart, Enhancements y Export. La sección «Look» (ajustes de imagen post-entrenamiento) es el renombrado real en la UI de la antigua sección «Finishing» — su `rawValue` interno de enum sigue siendo «Finishing» por razones de persistencia, pero la cabecera mostrada se llama «Look». Cada sección se puede colapsar haciendo clic en la cabecera, y el orden se puede reorganizar por arrastrar y soltar (InspectorView.swift:81–97). **En el primer inicio, las siete secciones arrancan colapsadas** (InspectorCollapsibleSections toma por defecto `Set(InspectorSection.allCases)`); después, el estado de la app persiste las preferencias de colapsado y orden entre inicios.

Varios controles del Inspector también aparecen, en forma casi idéntica, en los Ajustes (Capítulo 3) — típicamente backend de SfM, sky masking y defaults similares. La división es deliberada: Settings provee la plantilla global de la app para proyectos recién

creados, el Inspector sobrescribe esos valores para el proyecto abierto actualmente. Una vez conoces la lógica de operación de un lado, puedes usar el otro a ciegas.

La columna izquierda del Modo Experto — el Project Navigator — no forma parte del Inspector, pero es su vecino directo. Allí puedes seleccionar imágenes importadas haciendo clic, previsualizarlas con la barra espaciadora vía Quick Look y eliminarlas con el botón menos o la tecla Suprimir (con Cmd-Z para deshacer). El Inspector sigue la selección actual de la barra lateral con información de detalle según el contexto, pero las siete secciones principales se mantienen siempre disponibles.

Sección Look (L1–L5)

La sección Look (su `rawValue` interno sigue siendo «Finishing») es la sección superior del Inspector y reúne en un solo lugar los ajustes de imagen **post-entrenamiento**. Todos los reguladores trabajan de forma **no destructiva**: cada deslizador vuelve a aplicar el `FinishingPass` sobre una instantánea pristine sin modificar (color DC, opacidad y escalado originales) — el ajuste es por tanto **idempotente**, no acumulativo. El resultado aparece **en vivo en la vista** (WYSIWYG, exactamente como la exportación posterior) y se **hornea en cada exportación**. La sección solo está disponible **tras completarse un entrenamiento** (antes aparece «Available after a training run completes.»); sus valores se **restablecen con cada nuevo entrenamiento**. Mientras una exportación está en curso, todos los reguladores están **bloqueados** — aparece un aviso de bloqueo «Locked while exporting — the file uses the current settings.» y el GroupBox queda deshabilitado.

L1 Deslizador Saturation

DÓNDE

Inspector → Sección Look → GroupBox → Saturation.

TÉCNICO

Deslizador 0,5–1,2, visualización a dos decimales (p. ej., «1.00»). Escala el croma SH-DC de cada splat alrededor del valor de luminancia: 1,0 = sin cambios, < 1,0 = desaturado (color tirado hacia la escala de grises), > 1,0 = más intenso. Matemáticamente, el color DC se recalcula desde la instantánea pristine (`desaturateDC`), de modo que mover el deslizador repetidamente no se acumula. Se validó sobre material de dron DJI (viaducto de Pensford), que tiende a exagerar el color — el default de dron es 0,82. Solo actúa sobre la base de color (SH grado 0); los coeficientes SH superiores quedan intactos.

EN POCAS PALABRAS

Cómo de intensos son los colores del splat terminado. 1.00 deja todo como se entrenó, valores por debajo tiran el color hacia el gris — bien para material de dron o vídeo, que a menudo sale sobresaturado. Valores por encima de 1.0 lo hacen más intenso. Puedes mover el deslizador adelante y atrás cuanto quieras sin que nada se «amontone», porque la app siempre recalcula desde el estado original sin tocar. Visible en vivo en la vista y exactamente así en la exportación.

L2 Deslizador Splat length

DÓNDE

Inspector → Sección Look → GroupBox → Splat length.

TÉCNICO

Deslizador 0,3–1,0, visualización a dos decimales. Acerca los tres ejes de escalado de cada Gaussiano hacia su media en el espacio logarítmico (`shortenScale` , factor `alpha`): 1,0 = sin cambios, valores más pequeños hacen más redondos los splats alargados en «aguja», 0 serían esferas puras. Ataca los splats con forma de aguja, sobreestirados, sin cambiar el tamaño global, y reduce así los típicos artefactos de «confeti». Se aplica desde la instantánea pristina (escalado logarítmico original), por eso es idempotente. Conmuta con Splat size (L3), porque ambos trabajan en el espacio logarítmico.

EN POCAS PALABRAS

Hace más redondos los splats demasiado largos y astillados. 1.00 deja la forma como se entrenó, valores más bajos encogen las «agujas» alargadas hacia manchas más redondeadas — eso calma las reconstrucciones granuladas, plagadas de artefactos de confeti. El tamaño global se mantiene igual, solo se trata de la elongación. Se puede combinar sin problemas con Splat size (L3).

L3 Deslizador Splat size

DÓNDE

Inspector → Sección Look → GroupBox → Splat size.

TÉCNICO

Deslizador 0,5–2,0, visualización a dos decimales. Escala cada Gaussiano de forma uniforme en **los tres** ejes (`sizeScale`): 1,0 = sin cambios, < 1,0 = más pequeño/denso/nítido, > 1,0 = más grande/«esponjoso» (rellena los huecos entre los splats). Como los escalados están en el espacio logarítmico, la multiplicación se realiza como un offset aditivo `log(factor)` — eso conmuta con Splat length (L2), porque un offset constante deja intacta la desviación respecto a la media. Desde la instantánea pristina, por tanto idempotente. Nuevo en esta versión.

EN POCAS PALABRAS

Escala todos los splats de forma uniforme, más grandes o más pequeños. 1.00 es el estado entrenado, valores por debajo aprietan la nube de puntos y la hacen más nítida, valores por encima cubren los huecos entre los splats (queda más suave, «esponjoso»). Práctico para cerrar ópticamente una reconstrucción agujereada o, al revés, para revelar más detalle. Se lleva sin problemas con Splat length (L2) — ambos reguladores no se influyen mutuamente.

L4 Fade far region (con sub-deslizadores)

DÓNDE

Inspector → Sección Look → GroupBox → conmutador «Fade far region» más los sub-deslizadores «Fade start xradius» y «Fade floor».

TÉCNICO

Conmutador que activa una caída radial de opacidad con la distancia al baricentro de la cámara — el «confeti lejano», débilmente observado, del fondo se desvanece. **Solo para capturas orbitales:** el conmutador está deshabilitado cuando `finishingContext.fadeEligible` es `false` (vuelos lineales, demasiadas pocas cámaras o cámaras degeneradas); entonces, en lugar de los sub-deslizadores, aparece el aviso «Far-fade applies only to orbit captures (not this scene).» La aptitud se determina por la cobertura de azimut de las posiciones de cámara (una órbita rodea el baricentro y rellena muchos sectores de brújula, un vuelo lineal solo ~2). Dos sub-deslizadores controlan la geometría: **Fade start xradius** (1,0–3,0) fija el radio interior como múltiplo del radio orbital, dentro del cual rige la opacidad plena; **Fade floor** (0,0–1,0) es el factor de opacidad muy más allá del radio de fade. Importante: el fade **se salta la zona del sky-dome** (los Gaussianos congelados de los índices [0, frozen-Count]), para que la cúpula de fondo deliberada no se atenúe también.

EN POCAS PALABRAS

Desvanece los restos esponjosos del borde exterior de la escena — justo los grumos de «confeti lejano» que flotan muy atrás en las capturas envolventes. Funciona solo con capturas orbitales/envolventes reales; con vuelos de dron rectos o demasiadas pocas cámaras, el conmutador queda atenuado y un aviso explica por qué. Si está activo, se añaden dos reguladores finos: «Fade start xradius» fija a partir de qué distancia (como múltiplo del radio del círculo) empieza el desvanecimiento, «Fade floor» cuánto siguen viéndose al final los splats lejanos (0 = del todo fuera, 1 = sin cambios). Un sky-dome reconstruido a propósito (I44) nunca se toca — el cielo se conserva.

L5 Botón Reset finishing

DÓNDE

Inspector → Sección Look → GroupBox → «Reset finishing» (abajo, botón pequeño).

TÉCNICO

Restablece todos los ajustes de Look a los defaults (`FinishingPass.Settings() = Saturation 1,0, Fade desactivado, Splat length 1,0, Splat size 1,0`) y dispara de inmediato un nuevo finishing, de modo que la vista salta de vuelta al estado entrenado sin modificar `controlSize(.small)`. Como todo el stack de Look recalcula de forma idempotente desde la instantánea pristine, «volver al default» es exactamente la salida original del entrenamiento — sin pérdida de calidad por ir y venir varias veces. Como todos los reguladores de la sección, queda bloqueado durante una exportación en curso.

EN POCAS PALABRAS

Restablece de un clic todos los reguladores de Look a su estándar (Saturation 1.00, Fade desactivado, ambos deslizadores Splat a 1.00) — la vista muestra después de nuevo exactamente el resultado recién entrenado. Práctico cuando te has dejado llevar y quieres empezar limpio de cero. Como la app siempre recalcula desde el estado original, no hay pérdida de calidad. Mientras una exportación está en curso, el botón (como los deslizadores) está bloqueado.

Sección Presets (I1–I11)

La sección Presets es la forma más rápida de aplicar una configuración probada. Los ajustes preestablecidos integrados (Capture Class, Classic, MCMC, Hybrid) ofrecen puntos de partida reproducibles a partir de más de 560 experimentos documentados; los propios se pueden guardar, exportar, importar y compartir. La lista se agrupa por categorías (Capture Class, Classic, MCMC, Hybrid, Custom) y puede haber más de una categoría expandida al mismo tiempo. El menú contextual (clic derecho sobre una fila) hace accesibles Export, Duplicate y — para ajustes preestablecidos personalizados — Delete.

I1 Botón Save...

DÓNDE

Inspector → Sección Presets → botón Save... (barra de acciones de abajo).

TÉCNICO

Abre un popover con campo de texto y botones Save/Cancel. El estado actual de TrainingConfig se persiste como nuevo ajuste preestablecido personalizado (codificado en JSON, almacenado a nivel de app). El proceso de guardado copia los 81 parámetros de entrenamiento más la estrategia de densificación actual. El ajuste preestablecido aterriza automáticamente en la categoría Custom, con independencia de cuál fue el ajuste integrado del que derivó. Los nombres vacíos y las entradas con solo espacios en blanco se rechazan. Los nombres ya existentes no se rechazan — cada ajuste tiene su propio ID interno; los nombres duplicados son técnicamente válidos pero, en la práctica, confusos.

EN POCAS PALABRAS

Guarda tu configuración actual como ajuste preestablecido reutilizable. Haz clic en el botón, escribe un nombre en el popover y pulsa Save — los 81 parámetros, incluida la estrategia de densificación, aterrizan bajo el nombre elegido en la categoría Custom. Lo necesitas cuando has dedicado esfuerzo y no quieres empezar de cero en tu próximo proyecto. Especialmente útil para setups recurrentes como «Drone 4K» o «Indoor fast». Los nombres duplicados son técnicamente válidos pero, en la práctica, confusos — mejor escoge algo descriptivo.

I2 Campo de texto Preset Name

DÓNDE

Popover de Save → campo de texto «Preset Name».

TÉCNICO

Campo de texto simple con borde redondeado y forma ancha. El valor se toma como nombre del ajuste preestablecido al pulsar Save. Sin límite de longitud en la UI, pero el nombre almacenado debe ser codificable en JSON y mostrable en las listas de la UI — emojis y acentos funcionan. El contenido se restablece automáticamente a cadena vacía al abrir el popover. El botón Save se mantiene deshabilitado mientras el campo esté vacío tras un trim. No hay autosugerencia ni autocompletado con el nombre del ajuste preestablecido activo.

EN POCAS PALABRAS

Aquí escribes el nombre de tu ajuste preestablecido. Escoge algo descriptivo como «Drone 4K 30fps» o «Indoor fast» — así lo encontrarás más tarde en la categoría Custom. Se permiten emojis y acentos, no hay un límite duro de longitud. Mientras el campo esté vacío o solo contenga espacios, el botón Save se mantiene atenuado. Al reabrir el popover, el campo vuelve a estar vacío — no hay autocompletado con el nombre del ajuste activo.

I3 Botón Cancel (diálogo Save)

DÓNDE

Popover de Save → botón Cancel (izquierda).

TÉCNICO

Cierra el popover sin guardar. Descarta el contenido del campo de texto — en la próxima apertura, la lógica del botón Save... (I1) lo restablece a vacío. Estilo de botón estándar, sin diálogos de confirmación, sin hotkeys. La TrainingConfig actual no cambia, porque la vía de guardado ni siquiera se ejecutó.

EN POCAS PALABRAS

Cierra el popover Save sin guardar nada. Si has cambiado de idea, te has equivocado al teclear o has abierto el diálogo por accidente — simplemente pulsa Cancel. Tu configuración de entrenamiento actual no cambia, porque no se ha escrito nada. Al reabrir el popover, el campo de nombre vuelve a estar vacío. Sin aviso de seguridad, sin hotkey — clic y fuera.

I4 Botón Save (diálogo Save)

DÓNDE

Popover de Save → botón Save (derecha, estilo prominente).

TÉCNICO

Dispara la persistencia real. Valida de nuevo que el nombre no esté vacío (chequeo defensivo) y entonces escribe la TrainingConfig actual como JSON al almacenamiento de la app. Luego cierra el popover. Resaltado en azul, atenuado mientras el campo de texto esté vacío. Si guardar falla (p. ej., porque el almacenamiento de la app está lleno — muy improbable), por ahora no hay un diálogo de error visible; el ajuste preestablecido simplemente no aparecería en el próximo arranque de la app.

EN POCAS PALABRAS

Con un clic en Save tomas el nombre y escribes tu setup actual como nuevo ajuste preestablecido. El popover se cierra, el ajuste aparece de inmediato en la categoría Custom de la lista y desde entonces puede activarse con un clic. El botón está resaltado en azul (`borderedProminent`) y se mantiene atenuado mientras el campo de nombre esté vacío. Si guardar falla (p. ej., UserDefaults lleno), no hay diálogo de error visible — el ajuste simplemente faltaría en el próximo arranque de la app.

I5 Botón Export... **DÓNDE**

Inspector → Sección Presets → barra de acciones → botón Export...

 **TÉCNICO**

Exporta el ajuste preestablecido actualmente seleccionado como archivo `.radiancepreset` (internamente JSON). Deshabilitado si no hay ajuste seleccionado. Al hacer clic, la app abre un diálogo de guardar con un nombre de archivo de ajuste (nombre del ajuste + extensión `.radiancepreset`). El formato guardado contiene la TrainingConfig completa más metadatos (nombre, categoría, ID, flag de integrado). El doble clic en Finder abre la app — pero **no** automáticamente la importación; el usuario tiene que usar el botón Import (I6).

 **EN POCAS PALABRAS**

Selecciona un ajuste preestablecido en la lista y pulsa Export — entonces puedes guardarlo como archivo `.radiancepreset` y, p. ej., enviárselo a un compañero o transferirlo a un segundo Mac. El receptor lo carga en su lado con el botón Import... (I6) de nuevo. Funciona igual con integrados y propios. El botón aparece atenuado mientras no se haga clic en nada de la lista. Truco: por el menú contextual (I8) es aún más rápido — ahí no tienes que seleccionar primero el ajuste.

I6 Botón Import... **DÓNDE**

Inspector → Sección Presets → barra de acciones → botón Import...

 **TÉCNICO**

Abre un diálogo de archivo que solo permite archivos `.radiancepreset` (multiselección deshabilitada). Al seleccionar, el archivo JSON se carga, valida e inserta en la categoría Custom — con un nuevo ID interno, para que no haya colisiones con los integrados. La importación fija automáticamente la categoría a Custom, incluso si el ajuste exportado era, p. ej., originalmente un integrado. Los archivos corruptos o incompatibles con una versión de schema más antigua se rechazan en silencio, sin diálogo de error (el log de consola sí da información).

 **EN POCAS PALABRAS**

Lee un archivo `.radiancepreset` del disco. Útil cuando alguien te envía un setup probado o tú mismo quieres mantener tus ajustes preestablecidos favoritos sincronizados entre varios Macs. Los ajustes importados siempre aterrizan en la categoría Custom — aun cuando se hayan exportado originalmente desde los integrados. Los archivos corruptos o anticuados se ignoran en silencio; el log de consola muestra el motivo. La multiselección está deshabilitada en el diálogo, así que un archivo por clic.

I7 Fila de ajuste preestablecido (activación por clic)

DÓNDE

Inspector → Sección Presets → cada fila de ajuste preestablecido en cada categoría.

TÉCNICO

Al hacer clic en una fila se reemplazan todos los campos de la TrainingConfig por los valores del ajuste preestablecido, se recuerda el ID del ajuste activo y se restablece el estado Modified. El check de activo delante de la fila solo aparece cuando el ajuste está seleccionado Y no modificado. En cuanto se cambia un valor en TrainingConfig (deslizador, stepper, conmutador en las otras secciones del Inspector), aparece una insignia naranja «Modified» tras el nombre. Los ajustes preestablecidos integrados no pueden sobrescribirse — ante modificación tienes que crear una copia mediante el botón Save (I1).

EN POCAS PALABRAS

Hacer clic en una fila activa el ajuste preestablecido y toma todos los valores allí guardados en los ajustes de entrenamiento actuales. El check delante del nombre muestra qué ajuste está activo. En cuanto luego mueves cualquier deslizador, stepper o conmutador en las otras secciones, aparece una insignia naranja «Modified» tras el nombre — porque tu setup se desvía ya del ajuste preestablecido. Los integrados no se pueden sobrescribir; si quieres conservar los cambios, crea una copia con el botón Save... (I1) o duplica el ajuste (I9).

I8 Menú contextual «Export...»

DÓNDE

Clic derecho en cualquier fila de ajuste preestablecido → primera entrada «Export...».

TÉCNICO

Funcionalidad idéntica a I5 (botón Export...), pero accesible más cómodamente — sin que el ajuste preestablecido tenga que seleccionarse primero. Exporta directamente el ajuste clicado en la fila. Funciona igual para todas las categorías (integrada o custom), sin restricción. La exportación contiene el flag de integrado y la categoría original, pero en la reimportación la categoría se mapea a Custom como se describe en I6.

EN POCAS PALABRAS

Vía rápida para exportar — clic derecho sobre el ajuste deseado y elegir «Export...». Ahorra el rodeo de clic previo y luego pulsar el botón Export.... Funciona igual con todas las categorías, también con integrados. El archivo `.radiancepreset` creado es idéntico al de I5; en la reimportación posterior aterriza automáticamente en la categoría Custom.

I9 Menú contextual «Duplicate»

DÓNDE

Clic derecho en cualquier fila de ajuste preestablecido → segunda entrada «Duplicate».

TÉCNICO

Clona el ajuste preestablecido en la categoría Custom. Crea un nuevo ID interno, añade « Copy» al nombre y guarda la copia. También funciona con ajustes integrados — el clon es entonces editable. El original queda intacto. La TrainingConfig se copia valor por valor (round-trip JSON), así que no quedan bindings de referencia entre original y copia.

EN POCAS PALABRAS

Crea una copia editable de un ajuste preestablecido en la categoría Custom. Útil si, p. ej., quieres el ajuste integrado «Quality» como punto de partida y luego solo quieres mover un poco el deslizador SSIM. Flujo: duplicar, renombrar (menú contextual o nueva ejecución de Save...), ajustar, listo. El original queda intacto — puedes volver a él cuando quieras. También funciona con integrados, que es la única manera de usar sus valores como base y a la vez hacerlos editables.

I10 Menú contextual «Delete»

DÓNDE

Clic derecho en filas de ajustes preestablecidos propios → última entrada «Delete» (rojo, destructivo).

TÉCNICO

Solo visible para ajustes personalizados. Los integrados no se pueden eliminar. La entrada está marcada como destructiva, aparece en rojo en el menú contextual y queda separada tras un divisor para que no se haga clic por accidente. **No** hay diálogo de confirmación — un clic elimina el ajuste de inmediato. El ajuste eliminado no es recuperable (Cmd-Z no funciona aquí — el Undo existe en la build actual solo para la lista de imágenes, no para operaciones de ajustes preestablecidos). Si el ajuste eliminado estaba activo, la TrainingConfig actual no cambia, solo se anula la selección de ajuste activo.

EN POCAS PALABRAS

Elimina tus propios ajustes preestablecidos. Para los integrados (Quick, Preview, Balanced, Quality, Ultra Detail, Drone / Aerial, 360° Walkaround, Photo / Object, etc.) «Delete» ni siquiera es visible — no puedes cargártelos por accidente. Atención: no hay aviso de seguridad ni undo, un clic y el ajuste desaparece. Si no estás seguro, primero saca una copia de seguridad a disco mediante Export... (I5/I8) — la puedes reimportar cuando quieras. Si el ajuste estaba activo, tu TrainingConfig no cambia, solo desaparece el check.

I11 Cabecera de categoría (expandir/colapsar)

DÓNDE

Inspector → Sección Presets → cada cabecera de categoría (Capture Class, Classic, MCMC, Hybrid, Custom).

TÉCNICO

Estado de colapsado por categoría con predeterminados distintos: el grupo curado Capture Class arranca **expandido**, Classic, MCMC, Hybrid y Custom arrancan **colapsadas**. El estado no se persiste — al reiniciar la app, todas las categorías vuelven al estado por defecto. La flecha chevron rota animada. El número de la derecha en la cabecera muestra el número de ajustes en esa categoría. El área de impacto de clic abarca toda la región de la cabecera.

EN POCAS PALABRAS

Expande y colapsa categorías para mantener la lista de ajustes preestablecidos ordenada. Al iniciar la app, el grupo Capture Class está abierto, Classic, MCMC, Hybrid y Custom están cerrados. Haz clic en la cabecera (toda la región es clicable) y la lista se despliega o pliega con una breve animación de chevron. El pequeño número de la derecha muestra cuántos ajustes hay en la categoría. Tras reiniciar la app, vuelve el estado por defecto — la app deliberadamente no persiste este ajuste de colapso.

Sección Training Configuration (I12–I22)

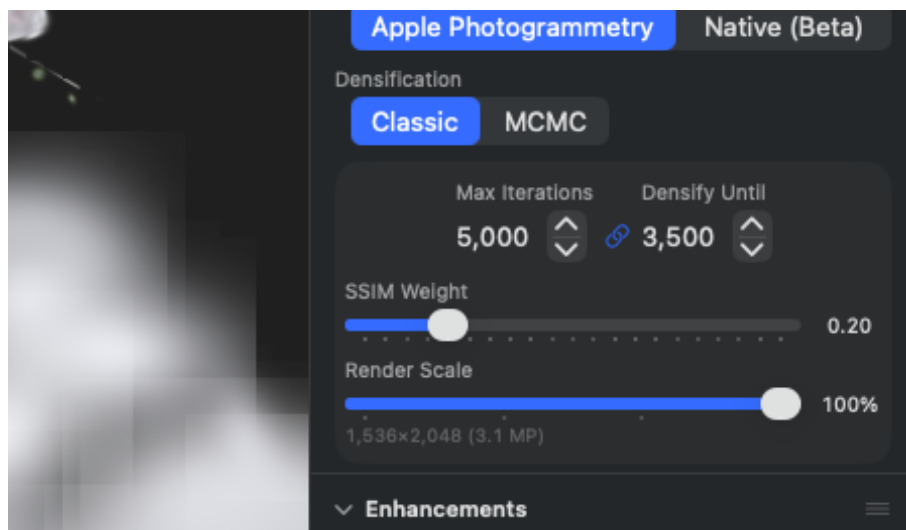


Figura 11: Recorte de solo la sección Training Configuration — Camera Alignment (Apple Photogrammetry activo, Native (Beta) inactivo), Densification (Classic activo), Max Iterations 5.000 / Densify Until 3.500 con icono de enlace, deslizador SSIM Weight 0,20, deslizador Render Scale al 100 % (1.536×2.048 = 3,1 MP)

Aquí viven las palancas centrales: qué backend de SfM debe calcular, cómo funciona la densificación, cuántas iteraciones, cuánto peso SSIM. Con la estrategia MCMC aparecen dos conmutadores adicionales («MCMC Quality» y «Auto-scale by scene») que quedan ocultos en modo Classic. Con el backend Native SfM se añade el campo de override de FOV, necesario solo para fotogramas de vídeo sin distancia focal EXIF.

I12 Selector Camera Alignment

DÓNDE

Inspector → Training Configuration → Camera Alignment (selector segmentado arriba).

TÉCNICO

Selector segmentado con dos opciones: Apple Photogrammetry y Native (Beta). La selección determina el backend de SfM usado en la próxima reconstrucción de cámara. A la vez influye en qué otros elementos del Inspector son visibles: Native muestra además el FOV override (I13), necesario solo con fotogramas de vídeo sin EXIF. Nota: para capturas exteriores muy grandes puedes importar el resultado de una herramienta externa (Metashape o COLMAP) mediante workspace import — véase Capítulo 1 (M5) y Capítulo 9 (Q3, Q6).

EN POCAS PALABRAS

Aquí eliges cómo se reconstruyen las posiciones de cámara — el conmutador más importante para la calidad final. Apple Photogrammetry es el estándar rápido y basta para la mayoría de los escaneos de objetos. Native (Beta) es el desarrollo propio compatible con App Store, bueno para órbitas y escenas de plato giratorio, y necesita el FOV override (I13) con fotogramas de vídeo sin EXIF. Con sets exteriores muy grandes puedes alternatively calcular las cámaras en Metashape o COLMAP y cargar el resultado mediante workspace import. Los detalles y recomendaciones por tipo de escena se encuentran en el Capítulo 9.

I13 Campo FOV Override (Native SfM)

DÓNDE

Inspector → Training Configuration → FOV Override (solo visible con Camera Alignment = Native).

TÉCNICO

Campo de texto numérico (rango 0-170°), predeterminado 0 = determinación automática a partir de EXIF o heurística. La entrada manual se necesita cuando las imágenes de entrada se extrajeron de un vídeo que no contiene metadatos de focal. Valores típicos: iPhone Wide ≈ 73°, DJI Mavic Wide-Crop ≈ 70°, dron con sensor de cuadro completo ≈ 84°. El valor se hace clamp a [0, 170] — los valores fuera se empujan de vuelta. Solo afecta a la pipeline nativa de SfM (Q4/Q5); Apple Photogrammetry ignora por completo este valor.

EN POCAS PALABRAS

Si tus imágenes no tienen EXIF (típico con fotogramas extraídos de vídeo), introduce aquí el campo de visión horizontal de la cámara en grados. Valores aproximados: iPhone Wide ≈ 73°, DJI Mavic Wide-Crop ≈ 70°, dron con sensor de cuadro completo ≈ 84°. Un 0 deja que la app adivine sola — a menudo va bien, pero puede salir mal con lentes raras. Los valores por encima de 170° se hacen clamp automáticamente. El campo solo es visible y solo surte efecto si has elegido Native como Camera Alignment (I12) — Apple Photogrammetry lo ignora por completo.

I15 Selector Densification

DÓNDE

Inspector → Training Configuration → Densification (selector segmentado, siempre visible).

TÉCNICO

Cambia entre las dos estrategias de densificación: Classic (procedimiento original 3DGS con clone/split/prune y umbral de gradiente) y MCMC (Stochastic Gradient Langevin Dynamics con Relocation, NeurIPS 2024). Al cambiar de Classic a MCMC, la app fija los campos específicos de MCMC automáticamente a valores por defecto probados (pesos de reg = 0, multiplicador de tope MCMC 3,0, schedule de sample/noise). Sin esa inicialización automática, las sesiones con ajustes preestablecidos antiguos sufrían el bug de colapso MCMC de la 1.4.4 (460K→5 Gaussianos, watchdog kill). La selección del selector determina además qué elementos del Inspector son visibles — con MCMC aparecen I16/I17. Efecto detallado por campo en el Capítulo 6, T11–T16 (Classic) y T61–T73 (MCMC).

EN POCAS PALABRAS

La elección central de estrategia para hacer crecer el número de Gaussianos. Classic está bien tunada a partir de 459 experimentos, produce resultados rápidos y de alta calidad, y no requiere conocer ningún campo de MCMC. MCMC es el enfoque más reciente (NeurIPS 2024), más reproducible y prescinde del ajuste manual de umbrales — pero calcula alrededor de 6× más para una calidad comparable. Al cambiar a MCMC, la app fija defaults seguros automáticamente para que el entrenamiento no caiga en el colapso de 1.4.4. Los detalles de los campos de estrategia están en el Capítulo 6 (T11–T16 Classic, T61–T73 MCMC).

I16 Conmutador MCMC Quality

DÓNDE

Inspector → Training Configuration → MCMC Quality (solo con Densification = MCMC).

TÉCNICO

Cambia la acumulación de gradientes a 2 pasos (activo) o 1 paso (inactivo). Acumula los gradientes de dos vistas de cámara consecutivas antes de ejecutar el paso del optimizador. Empíricamente (Sesión 33, V544a) reduce el error L1 final en torno al 6 % (0,0246 con Quality vs 0,0261 sin, en promedio de 3 trials sobre Horse-Full-MCMC). El precio: el doble de tiempo de entrenamiento. Con entrenamientos muy largos (200K iteraciones) esto añade más de 10 minutos de espera — así que solo merece la pena si los últimos puntos porcentuales de calidad son realmente necesarios. Solo afecta al entrenamiento, no al formato de exportación ni a la visualización en la vista.

EN POCAS PALABRAS

Modo de calidad para MCMC con acumulación de gradientes entre dos vistas. Hace el resultado final empíricamente alrededor de un 6 % mejor (L1 0,0246 en lugar de 0,0261 en la prueba Horse), pero cuesta el doble de tiempo. Si ya estás ejecutando un entrenamiento MCMC de 200K (fácilmente 2 horas), se añade casi otra hora. Merece la pena para renderizados de exhibición final o al cierre de una sesión de barrido de calidad, menos en el flujo diario. Solo visible cuando Densification está en MCMC (I15).

I17 Conmutador Auto-scale by scene

DÓNDE

Inspector → Training Configuration → Auto-scale by scene (solo con MCMC).

TÉCNICO

Cuando está activo, escala el techo efectivo de max-Gaussians con el número de puntos init de SfM \times multiplicador de tope MCMC (predeterminado 3,0). Ejemplo: SfM da 250K puntos init, tope base = 150K, multiplicador 3,0 → techo efectivo = $\max(150K, 750K) = 750K$. Cuando está desactivado, solo se aplica estrictamente el base. Se introdujo en v1.4.5 porque las grandes capturas exteriores con más de 1000 fotogramas y, consecuentemente, alta densidad de puntos SfM, mataban de hambre la densificación con el tope rígido por defecto de 150K — los puntos sobrantes se quedaban, los nuevos no podían surgir. Predeterminado OFF en ajustes personalizados, ON en los integrados de MCMC. Solo afecta al tiempo de entrenamiento, no a la exportación.

EN POCAS PALABRAS

Deja que el número máximo de Gaussians crezca con el tamaño de la escena (más exactamente: con el número de puntos init de SfM). Con escenas pequeñas apenas notas diferencia, con escenas exteriores grandes a menudo es crucial para la calidad — de otro modo el entrenamiento «se asfixia» porque el techo predeterminado de 150K es muy bajo para la escena. Se introdujo en la v1.4.5 después de que sets exteriores muy grandes (más de 1000 fotogramas) tocaran visiblemente el tope. Con los ajustes preestablecidos integrados de MCMC está activado de antemano; en los propios, está desactivado por defecto.

I18 Stepper Max Iterations

DÓNDE

Inspector → Training Configuration → GroupBox → Max Iterations.

TÉCNICO

Stepper con rango 1.000–100.000, paso 1.000. Determina el número total de iteraciones del optimizador. Linealmente correlacionado con el tiempo de entrenamiento (mitad \approx 50 % del tiempo). Sweet spots empíricos: 20K (Classic Balanced, $L1 \approx 0,028$), 40K (Classic Quality, $L1 \approx 0,023$), 200K (MCMC Full, $L1 \approx 0,0246$). Por encima de 40K con Classic apenas aporta mejoras en promedio — diminishing returns. Al cambiar, si la función link (I19) está activa, Densify Until se arrastra de forma proporcional (ratio predeterminado: 0,5, es decir, Densify-Until = $\text{Max}/2$).

EN POCAS PALABRAS

Cuántos pasos de entrenamiento se ejecutan — más es mejor, pero también cuesta linealmente más tiempo. Regla práctica: 20.000 para buena calidad, 40.000 para el óptimo con la estrategia Classic (por encima de eso en promedio apenas aporta nada más). MCMC necesita bastante más, 200.000 es aquí el estándar. Duplicar las iteraciones duplica aproximadamente el tiempo de entrenamiento. Con el botón de enlace (I19) activo, Densify Until se arrastra proporcionalmente — prácticamente siempre lo que quieres.

I19 Botón Link/Unlink (Densify ↔ Iterations)

DÓNDE

Inspector → Training Configuration → GroupBox → pequeño botón de enlace entre Max Iterations y Densify Until.

TÉCNICO

Botón conmutador que congela el ratio de Densify Until respecto a Max Iterations. Cuando está activo (icono de enlace resaltado), en cada cambio de Max Iterations el Densify Until se arrastra proporcionalmente. Cuando está desenchazado (icono link-plus), los valores permanecen independientes. El predeterminado está enlazado, porque eso refleja la correlación típica — cuando subes el entrenamiento al doble de iteraciones, normalmente quieres dejar correr también la densificación proporcionalmente más. El ratio se calcula a partir del valor actual al establecer el botón de enlace; un ratio típico es 0,5 (Densify-Until = la mitad del número de iteraciones).

EN POCAS PALABRAS

Pequeño botón clip entre Max Iterations y Densify Until. Cuando está activo (icono de enlace resaltado), los dos valores se mueven juntos — si duplicas las iteraciones, Densify Until también se duplica en el mismo ratio. Cuando no (icono `link.badge.plus`), puedes ajustarlos de forma independiente. El predeterminado está enlazado, porque eso refleja la correlación típica — un entrenamiento más largo suele querer una fase de densificación más larga. Para el 99 % de los casos déjalo bloqueado.

I20 Stepper Densify Until

DÓNDE

Inspector → Training Configuration → GroupBox → Densify Until.

TÉCNICO

Stepper con rango 500–50.000, paso 500. Determina el índice de iteración a partir del cual ya no se añaden nuevos Gaussianos mediante clone/split (Classic) o relocalización (MCMC). Tras alcanzarlo, solo se refinan posición y color. Valores más altos = más Gaussianos = archivo más grande, más tiempo de GPU por iteración (+30–60 % de tiempo de GPU por paso). Valores típicos: 15K (para 30K max-iter), 20K (para 40K), 100K (para 200K MCMC). Con link activo (I19), escalado automáticamente. Actúa de forma distinta con Classic vs MCMC: Classic detiene del todo el crecimiento, MCMC detiene la lógica de relocalización, pero la adaptación de sample/noise sigue corriendo.

EN POCAS PALABRAS

Hasta qué iteración pueden añadirse nuevos Gaussianos — con Classic mediante clone/split, con MCMC mediante relocalización. Después, solo se trata de refinar color y forma de los puntos existentes. Más alto = más detalle, pero también archivo más grande y +30–60 % de tiempo de GPU por paso. Valores típicos: 15K (para 30K max-iter), 20K (para 40K), 100K (para 200K MCMC). Normalmente cuelga vía link (I19) de Max Iterations — rara vez tiene sentido desacoplarlo manualmente.

I21 Deslizador SSIM Weight DÓNDE

Inspector → Training Configuration → GroupBox → SSIM Weight.

 TÉCNICO

Deslizador 0,0–1,0 en pasos de 0,05, mostrado como «0.20». Mezcla loss L1 (0,0) y loss SSIM (1,0). L1 ajusta la luminosidad por píxel, SSIM ajusta la similitud estructural (bordes, estadísticas locales). El predeterminado 0,2 es el valor del paper 3DGS original (Kerbl 2023) y se ha verificado como compromiso robusto en numerosas sesiones. Valores más altos (0,5+) favorecen la preservación de detalle, pero pueden ignorar errores locales de luminosidad. Valores más bajos (< 0,1) llevan a pérdida de detalle en bordes nítidos. El cómputo de SSIM corre en el shader con una ventana Gaussiana 11×11. Rendimiento: en 0,0 (solo L1), el entrenamiento es aprox. un 8-12 % más rápido, porque el cómputo SSIM se omite en el shader.

 EN POCAS PALABRAS

Con qué fuerza se pondera la similitud estructural de imagen (bordes, patrones locales) frente a la comparación pura de luminosidad. 0,2 es el estándar del paper 3DGS original y basta para casi todas las escenas. Más alto (0,5+) para estructuras finas como pelo, pelaje o vegetación — ahí más peso estructural ayuda. Más bajo (0,0) hace el entrenamiento alrededor de un 8-12 % más rápido, porque se omite el cómputo SSIM en el shader, pero cuesta detalle en bordes nítidos. Si no tienes una buena razón para un cambio, deja 0,2 como está.

I22 Deslizador Render Scale **DÓNDE**

Inspector → Training Configuration → GroupBox → Render Scale.

 **TÉCNICO**

Deslizador 0,25–1,0 en pasos de 0,25, mostrado como «100%». Escala la resolución de renderizado del entrenamiento relativa al tamaño de la imagen fuente. La mayor palanca de rendimiento: 50 % reduce el tiempo de GPU aprox. un 75 % (porque 4x menos píxeles), 25 % aprox. un 94 %. El umbral de gradiente se escala junto automáticamente. Bajo el deslizador aparece una visualización de resolución en vivo en MP (p. ej., «2304×1296 (3,0 MP)»). Si el valor actual se desvía del recomendado, «— recomendado: 50 %» se muestra en texto naranja. La recomendación apunta a ~3 MP de resolución efectiva — el rango más eficientemente procesado por las GPUs de Apple Silicon. Las imágenes fuente 4K reciben, p. ej., una recomendación de 25 % automáticamente; las imágenes FullHD, 100 %. Un cambio también dispara la realocación de buffers.

 **EN POCAS PALABRAS**

A qué resolución renderiza el entrenamiento — una de las mayores palancas de rendimiento. Completa (100 %) da la mejor calidad, pero cuesta mucho tiempo de GPU con imágenes grandes. Media (50 %) ahorra alrededor del 75 % de tiempo de GPU, porque se calculan cuatro veces menos píxeles — perfecto para fuentes 4K. Bajo el deslizador ves la resolución efectiva en megapíxeles; la app apunta a unos 3 MP, porque eso corre con la mayor eficiencia en Apple Silicon. Si tu valor se desvía de eso, la app muestra un aviso naranja de «recomendado» — normalmente vale la pena seguirlo.

Sección Enhancements (I26–I29, I42–I44)

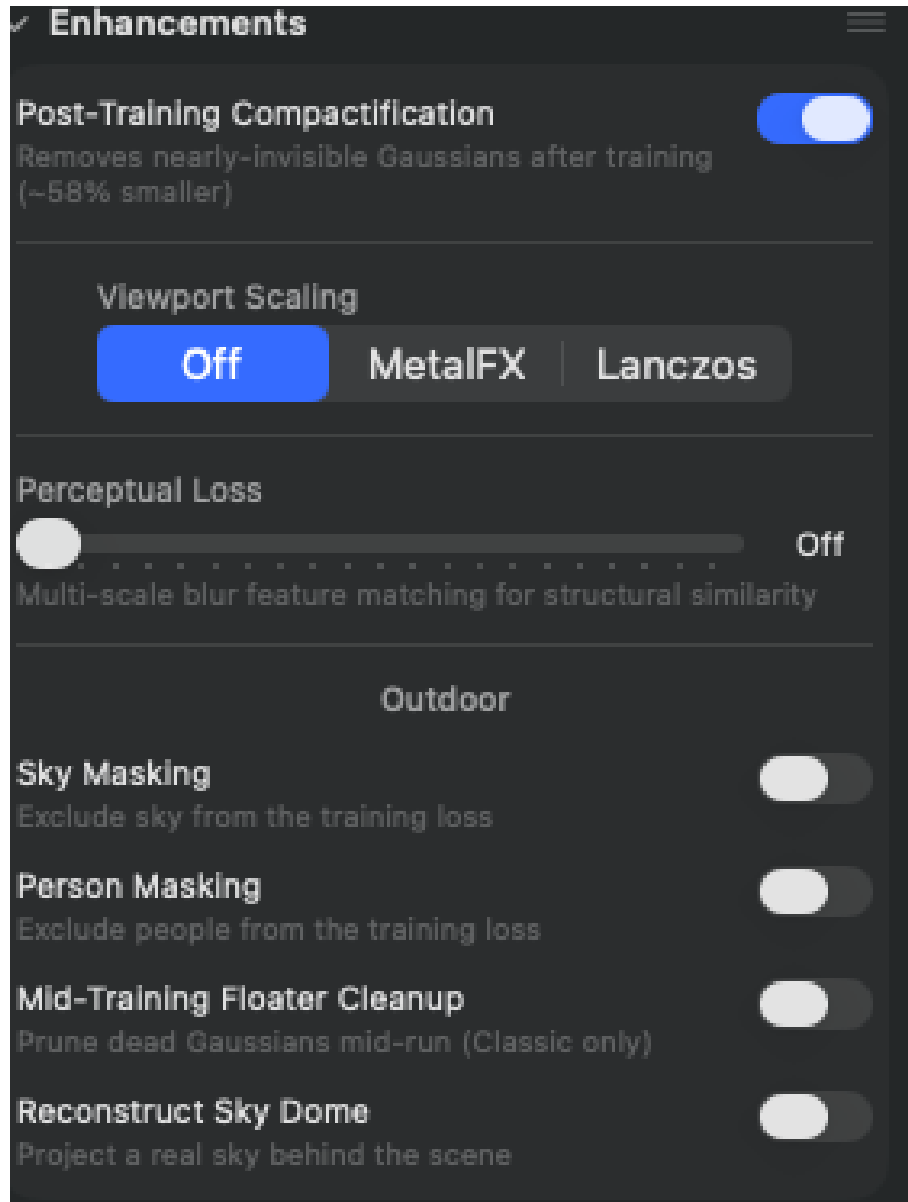


Figura 12: Recorte de solo la sección Enhancements — tres filas: Post-Training Compactification (toggle on), Viewport Scaling (selector segmentado Off/MetalFX/Lanczos), Perceptual Loss (deslizador en «Off»). Cada fila con subtítulo explica su función

La sección Enhancements agrupa tres funciones que mejoran la calidad de imagen sin cambiar en sí el bucle de entrenamiento. Las dos primeras (I26–I27) son etapas **post-entrenamiento** o **de vista**: Compactification limpia tras terminar el entrenamiento, el Viewport Scaling es un renderer puro de vista que no influye en el entrenamiento en curso. El Perceptual Loss (I29), pese a pertenecer a la sección, es un componente del entrenamiento — se activa como término de loss adicional durante el entrenamiento; de ahí la separación de los reguladores de vista mediante un divisor. A partir de la v1.6, la sección también tiene un grupo Outdoor (I42–I44: Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome) — opciones de entrenamiento contra los floaters del cielo que antes vivían en la ventana de Ajustes y que ahora residen aquí por proyecto.

I26 Conmutador Post-Training Compactification **DÓNDE**

Inspector → Enhancements → Post-Training Compactification.

 **TÉCNICO**

Activa el post-procesado V443: tras terminar las iteraciones de entrenamiento, los Gaussianos con opacidad por debajo de 0,01 (1 % de visibilidad) se eliminan. Empíricamente, esto reduce el tamaño de archivo en ~55-58 % sin pérdida visible de calidad — porque esos Gaussianos visualmente no aportan de todos modos. La compactificación corre como una pasada compact GPU y tarda fracciones de segundo a unos segundos según el número de Gaussianos. No afecta al rendimiento del entrenamiento. Si este conmutador está desactivado, también se exportan los Gaussianos invisibles — solo relevante si quieres usar el formato para otra etapa de entrenamiento (Continue Training); en caso contrario, almacenamiento desperdiciado.

 **EN POCAS PALABRAS**

Limpia tras el entrenamiento los Gaussianos que de todos modos no puedes ver (opacidad por debajo del 1 %). Hace los archivos de exportación aproximadamente la mitad de grandes (~55-58 % de reducción de tamaño) sin pérdida visible de calidad. Corre como una breve pasada de GPU tras la última iteración, tarda solo fracciones de segundo a unos segundos. Debería estar prácticamente siempre activado — la única razón para desactivarlo es si quieres continuar el entrenamiento más tarde mediante Continue Training y necesitas conservar también los Gaussianos invisibles. Con flujos normales de exportación déjalo simplemente activado.

I27 Selector Viewport Scaling DÓNDE

Inspector → Enhancements → Viewport Scaling (selector segmentado con tres opciones: Off, MetalFX, Lanczos).

 TÉCNICO

Un único selector segmentado que elige el upscaler de la vista — las tres opciones son **mutuamente excluyentes**. Cuando la resolución de entrenamiento (vía I22 Render Scale) es menor que el tamaño de la vista, el modo elegido escala el fotograma renderizado hasta el tamaño de visualización. **Off** = simple estirado bilineal. **MetalFX** = el Spatial Upscaler basado en ML de Apple, la opción más nítida (el modelo ML está optimizado para bordes nítidos), sobrecarga aprox. 1-2 ms por fotograma en GPUs M3. **Lanczos** = los Metal Performance Shaders de Apple con resampling sinc de 8 taps, clásico sin ML, sobrecarga mínima (< 0,5 ms), calidad inferior a MetalFX, pero sin el «alisado» típico del ML de estructuras finas de líneas. La pipeline del render se reconfigura en vivo al conmutar — visible al instante, sin reinicio. **Contexto:** antes eran dos conmutadores separados (MetalFX + Lanczos) que podían estar activos a la vez — un estado contradictorio en el que MetalFX pasaba en silencio por encima de Lanczos. El selector elimina ese estado; un estado «ambos-activos» heredado eventualmente de sesiones antiguas se cura solo a MetalFX en el siguiente cambio. Actúa **solo** en la vista en vivo, no en las exportaciones renderizadas (vídeo orbital, capturas) — esas se renderizan a la resolución completa de la fuente.

 EN POCAS PALABRAS

Afila la imagen en vivo en la vista — especialmente útil si trabajas con resolución de entrenamiento reducida (Render Scale 50 %, véase I22). Tres niveles, de los que siempre hay solo uno activo: «Off» simplemente estira los píxeles, «MetalFX» usa el aprendizaje automático de Apple para los bordes más nítidos (prácticamente siempre la mejor elección), «Lanczos» es el filtro clásico sin ML — tómallo como fallback en caso de que MetalFX te alise líneas o muestre artefactos en una escena. Surte efecto en vivo, sin reinicio. Solo actúa en la vista en vivo, no en vídeos orbitales exportados o capturas — esos se renderizan a la resolución completa de la fuente. A diferencia de antes, ya no puedes elegir por error dos modos a la vez.

I29 Deslizador Perceptual Loss

DÓNDE

Inspector → Enhancements → Perceptual Loss.

TÉCNICO

Deslizador 0,0–0,2 en pasos de 0,01, mostrado en 0,0 como «Off», en otro caso como «0.05», etc. Activa un término de loss adicional que compara el blur Gaussiano multiescala del renderizado con la imagen ground truth (3 escalas de blur). Captura diferencias estructurales que L1+SSIM por sí solos no detectan. Implementación V460. Empíricamente, un valor de 0,05–0,1 mejora la puntuación L1 en las sesiones unos pocos puntos porcentuales, pero cuesta ~5 % de tiempo de entrenamiento (forward pass adicional por los kernels de blur). Por encima de 0,15, el entrenamiento se vuelve inestable y L1 vuelve a empeorar (el término de loss domina la optimización). Actúa **durante** el entrenamiento, no en post-procesado — pese a su posición en la sección «Enhancements», no es un acabado posterior puro.

EN POCAS PALABRAS

Un término de loss adicional que comprueba la similitud estructural de imagen sobre tres etapas distintas de blur. Ayuda especialmente en escenas con estructuras finas como pelo, tela o vegetación, porque captura patrones que L1+SSIM por sí solos no ven. Los valores más pequeños son más seguros — 0,05 a 0,1 es el sweet spot; por encima de 0,15 el entrenamiento se vuelve inestable y el loss vuelve a empeorar. En 0 (Off), la función está totalmente apagada y no cuesta nada; activa, come alrededor de un 5 % del tiempo de entrenamiento por el forward pass extra a través de los kernels de blur. Actúa, pese a la sección «Enhancements», directamente durante el entrenamiento, no solo en post-procesado.

I42 Enmascaramiento de cielo **DÓNDE**

Inspector → Enhancements (grupo Outdoor) → Conmutador «Sky Masking». Bound: `AppState.trainingConfig.skyMaskingEnabled` (por proyecto, `@DefaultFalse`). Predeterminado: `false` .

 **TÉCNICO**

Activa la segmentación de píxeles de cielo basada en Apple Vision previa al entrenamiento. Antes de iniciarlo, para cada cámara de entrada se extrae la región de cielo mediante la máscara de primer plano de Apple Vision (Sky = Background) y se asigna como máscara por píxel a la respectiva cámara. Durante el entrenamiento, la contribución del loss por píxel se multiplica por el complemento del enmascaramiento de cielo — los píxeles de cielo contribuyen 0 al gradiente, por lo que los Gaussianos que se proyectan al cielo no reciben señales de optimización y por tanto no se vuelven más «densos» ni más «brillantes». Reduce significativamente los floaters (grupos oscuros en el cielo) en escenas de exteriores o dron. Cuesta una regresión de L1 del ~3 % en el entrenamiento clásico de 40K (véase [memory/dev_outdoor-floater-reduction.md](#)). Solo es útil para escenas exteriores con cielo claramente reconocible; en escenas interiores o con fondos blancos, la segmentación de cielo identifica zonas equivocadas y bloquea señales válidas de loss. El valor ahora se almacena por proyecto (ya no es global de la app) y sigue al ajuste preestablecido / archivo de escena.

 **EN POCAS PALABRAS**

En tomas exteriores con cielo en el encuadre, a menudo se forman grupos negros o coloreados en el cielo — los llamados «floaters». Esta opción detecta automáticamente dónde está el cielo y le dice al entrenamiento: «Deja el cielo en paz.» Funciona muy bien para vuelos de dron y escenas de paisaje. En interiores o con fondos oscuros puede empeorar la imagen — así que solo actívalo cuando hay cielo real visible. Detalles: [memory/dev_outdoor-floater-reduction.md](#).

I43 Limpieza de floaters a mitad del entrenamiento

DÓNDE

Inspector → Enhancements (grupo Outdoor) → Conmutador «Mid-Training Floater Cleanup». Bound: `AppState.trainingConfig.floaterCleanupEnabled` (por proyecto, `@DefaultFalse`). Predeterminado: `false`.

TÉCNICO

Habilita dos pasadas adicionales de control de densidad durante el entrenamiento Classic de 40K (ajuste preestablecido «P4 Quality»): en la iteración 20,000 y en la 30,000. Ambas pasadas examinan todos los Gaussianos según tres criterios: (a) opacidad muy baja (predeterminado 0,005), (b) tamaño en espacio de pantalla diminuto, (c) sin contribuciones al loss en las últimas 1000 iteraciones. Los Gaussianos que cumplen las tres condiciones se purgan. Efecto: ~5–15 % menos Gaussianos al final del entrenamiento, visiblemente menos grupos oscuros en el cielo en escenas de dron/exteriores. Cuesta una regresión de L1 del ~1–3 % en escenas interiores de primer plano, por eso no está activa por defecto. Las dos iteraciones de limpieza (20K, 30K) están codificadas y de momento no se pueden cambiar desde la UI; para entrenamientos más cortos (p. ej. P2 Preview 5K), el conmutador no tiene efecto, porque nunca alcanza los hitos de iteración. **Nuevo:** el conmutador solo se puede manejar cuando el ajuste preestablecido activo usa el densificador **Classic** (`densificationStrategy == .classic`). Con MCMC o Hybrid queda **deshabilitado** y aparece un aviso inline, porque esas estrategias gestionan de todos modos los Gaussianos muertos por sí mismas (MCMC vía relocalización, Hybrid vía una lógica combinada de reloc/ruido) — las pasadas manuales de limpieza serían ahí inútiles o contraproducentes. Referencia de código: `RadianceKitApp.swift`, pestaña General. Detalles: `memory/dev_outdoor-floater-reduction.md`.

EN POCAS PALABRAS

Durante el entrenamiento, a veces se forman puntos Gaussianos «muertos» que ya no contribuyen a la calidad de la imagen pero ocupan memoria. Esta opción hace una limpieza dos veces durante un entrenamiento largo (en 20K y 30K iteraciones) y elimina esos cadáveres. En escenas exteriores con cielo es especialmente útil, porque la mayoría de los floaters se reúnen allí. Para entrenamientos pequeños o primeros planos de muebles, no es realmente necesario. El conmutador solo se puede activar cuando tu ajuste preestablecido usa el densificador Classic — con ajustes MCMC o Hybrid queda atenuado (junto con una breve explicación), porque esos limpian sus puntos muertos por su cuenta.

I44 Reconstruir cúpula de cielo **DÓNDE**

Inspector → Enhancements (grupo Outdoor) → Conmutador «Reconstruct Sky Dome». Bound: `AppState.trainingConfig.skyDomeEnabled` (por proyecto, `@DefaultFalse`). Predeterminado: `false`.

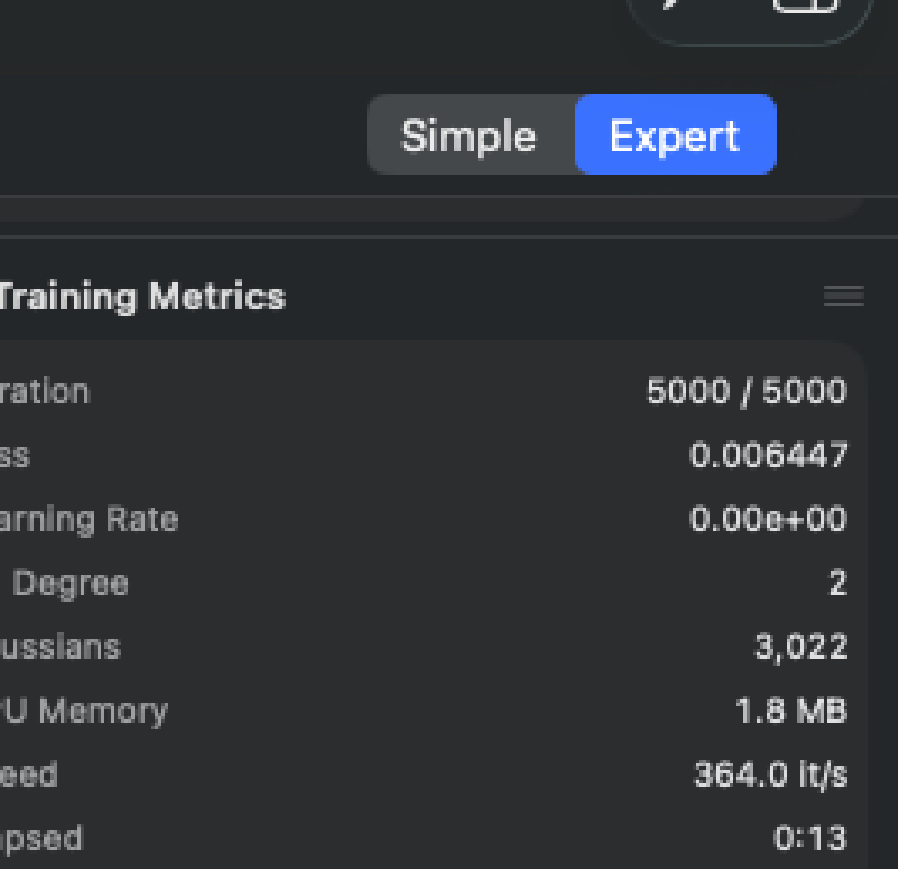
 **TÉCNICO**

Activa la proyección de Sky Dome previa al entrenamiento (V549e MVP). Tras el SfM y antes del inicio del entrenamiento, se extrae de la imagen para cada cámara de entrada la máscara de cielo de Apple Vision compartida con I42, los píxeles de cielo se desproyectan usando los intrínsecos de la cámara sobre una superficie esférica virtual (radio predeterminado 8x el radio de la escena). Sobre esa esfera se inicializan ~5000 nuevos Gaussianos con medias de color de los píxeles de cielo proyectados, escala muy grande (1,0 en unidades de escena) y opacidad inicial 0,95. Estos 5000 Gaussianos no son una máscara de cielo en sentido clásico — se entrenan como todos los demás, pero la alta opacidad inicial los mantiene en una capa delgada. Resultado: en vistas novel de 360° en escenas exteriores/dron aparecen el color y las estructuras de nube reales en lugar de los grupos de confeti oscuro. El valor se recuerda entre reinicios. Solo útil para escenas exteriores con al menos cobertura de cámara de 360°; en capturas puras de objeto sin vista del cielo no tiene efecto. Estado: experimental, validación A/B más amplia sobre más sets exteriores aún pendiente.

 **EN POCAS PALABRAS**

En lugar de que el entrenamiento intente «adivinar» el cielo a partir de los pocos píxeles visibles (lo que lleva a floaters), RadianceKit proyecta los píxeles de cielo directamente sobre una esfera virtual alrededor de la escena antes de iniciar el entrenamiento. Cuando luego rotas la escena terminada en 360°, ves cielo real en lugar de grupos negros. Solo funciona en tomas exteriores en las que de hecho hay cielo en el encuadre. Para escaneos de salones o tomas de estudio no aporta nada.

Sección Metrics (I30–I38)



Metric	Value
Iteration	5000 / 5000
Loss	0.006447
Learning Rate	0.00e+00
Degree	2
Gaussians	3,022
GPU Memory	1.8 MB
Speed	364.0 It/s
Elapsed	0:13

Figura 13: Recorte de solo la sección Training Metrics tras un entrenamiento completado en el ramo (5K iteraciones, 2.991 Gaussianos finales) — tabla con métricas de entrenamiento (Iteration, Loss, SSIM Loss, Combined Loss, Gaussian Count, Learning Rate, Elapsed, ETA)

Mientras un entrenamiento está en curso, la sección Metrics muestra nueve valores en vivo del bucle de entrenamiento. Antes de iniciar un entrenamiento, la sección está vacía («Start training to see live metrics»). Todos los valores se actualizan cada ~30 iteraciones (frecuencia de actualización del stream). La sección es de solo lectura — ningún elemento es clicable o modificable. Para un análisis más profundo, consulta los logs de entrenamiento JSONL en `~/Documents/RadianceKit/Logs/` (`script python3 scripts/analyze_logs.py best 5`).

I30 Iteration **DÓNDE**

Inspector → Metrics → Iteration. Solo lectura.

 **TÉCNICO**

Se muestra en el formato «4523 / 40000» — iteración actual sobre el total de iteraciones planificadas. Cuenta de forma sincronizada con el bucle de entrenamiento, que empuja los valores cada ~30 iter. El segundo número corresponde al valor de Max Iterations en el momento del inicio; ya no cambia, aun cuando el usuario ajuste después el stepper — el run en curso usa su propia copia snapshot. Si la app añade iteraciones adicionales mediante el menú Training (Continue Training +5K/+10K/+20K), el denominador aumenta.

 **EN POCAS PALABRAS**

Dónde está el entrenamiento ahora mismo. «4523 / 40000» significa: 4523 de 40.000 pasos hechos, así que aproximadamente el 11 % completado. El número de la izquierda cuenta hacia arriba en tiempo real; si se queda parado durante minutos, el entrenamiento está colgado — normalmente indicio de throttling de GPU o una app competidora. El número de la derecha corresponde al valor de Max Iterations (I18) al inicio del entrenamiento y ya no cambia, aun cuando ajustes después el stepper. Con Continue Training (+5K/+10K/+20K) crece según los pasos adicionales.

I31 Loss **DÓNDE**

Inspector → Metrics → Loss. Solo lectura.

 **TÉCNICO**

Valor float con seis decimales (p. ej., «0.024385»). Mide el loss combinado L1+SSIM (mezcla controlada vía I21 SSIM Weight) más el Perceptual Loss opcional (I29) y otros regularizadores. La escala no es absoluta, sino dependiente de la escena — exige el mismo dataset para la mayoría de las comparaciones. Valores finales típicos con buenas configuraciones: - Classic Quality 40K iter: 0,022–0,025 (Horse, Truck, Garden) - MCMC Full 200K iter: 0,024–0,028 - Outdoor dron 30K: 0,030–0,060 (peor por razones de geometría) - Apartamentos indoor: 0,018–0,025

Valores por encima de 0,10 tras 5K iteraciones indican problemas de SfM (malas poses de cámara) — aborta y recalcula el SfM.

 **EN POCAS PALABRAS**

Cuánto se desvía la imagen renderizada aún del original — combinado de L1, SSIM y posiblemente Perceptual Loss. Más pequeño es mejor. Por debajo de 0,03 normalmente es muy bueno, por debajo de 0,05 aún aceptable; las escenas exteriores están, por razones de geometría, más bien en 0,03–0,06. Por encima de 0,10 tras varios miles de iteraciones es una señal de aviso — normalmente es la reconstrucción de cámara (el SfM no funcionó limpiamente). La escala no es absoluta, sino dependiente de la escena; haz comparaciones solo dentro del mismo dataset. Si el número de repente sube a saltos, normalmente ha ocurrido un evento de explosión de gradiente.

I32 Learning Rate

DÓNDE

Inspector → Metrics → Learning Rate. Solo lectura.



TÉCNICO

Visualización en notación científica (p. ej., «1.60e-04»). Tasa de aprendizaje actual para los parámetros de posición (3DGS tiene seis LRs independientes para posición, SH-DC, SH-Rest, opacidad, escala, rotación — aquí se muestra la LR de posición como valor representativo). Valor inicial predeterminado 1,6e-4, que decae mediante decay exponencial a ~1,6e-6 al final del entrenamiento. El decay es ajustable mediante el campo LR schedule en la configuración de entrenamiento (campo T en el Cap. 6). Si la LR se mantiene inusualmente alta (p. ej., 1e-3 o más después de 10K iteraciones), podría indicar una configuración cargada incorrectamente.

EN POCAS PALABRAS

Lo grandes que son ahora mismo los pasos de optimización — concretamente la tasa de aprendizaje para las posiciones Gaussianas. Empieza en 1.60e-04 y cae exponencialmente hasta cerca de 1.60e-06 al final del entrenamiento («1.60e-06» = 0,0000016). La curva corre automáticamente, no tienes que ajustar nada aquí. Si el valor tras más de 10.000 iteraciones sigue siendo mayor que 1e-3, probablemente se cargó una config incorrecta — aborta el entrenamiento y elige un nuevo ajuste preestablecido. Internamente, 3DGS tiene seis tasas de aprendizaje independientes (posición, SH-DC, SH-Rest, opacidad, escala, rotación); aquí solo ves la LR de posición como sustituto.

I33 SH Degree

DÓNDE

Inspector → Metrics → SH Degree. Solo lectura.



TÉCNICO

Entero 0-3. Grado de spherical harmonics para la representación de color. Empieza en 0 (solo el componente DC, es decir, color independiente de la dirección por Gaussiano — es decir, solo una constante RGB) y sube progresivamente hasta 3 a lo largo del entrenamiento. El schedule estándar sube el grado en 1000/2000/3000 iteraciones en 1 cada vez. SH-3 corresponde a 48 coeficientes de color por Gaussiano (3 canales RGB × 16 funciones base SH). Mayor grado SH = más reflejos dependientes de la dirección (las superficies brillantes se ven correctamente distintas bajo distintos ángulos de visión), pero también más memoria y entrenamiento más lento.

EN POCAS PALABRAS

Lo compleja que es ahora mismo la representación de color por Gaussiano. Empieza en 0 (solo un color independiente de la dirección por punto) y se sube por pasos a 3 — típicamente en la iteración 1000, 2000 y 3000. La etapa 3 significa 48 coeficientes de color por Gaussiano y permite reflejos dependientes de la dirección, es decir, que las superficies brillantes se ven correctamente distintas desde diversos ángulos de visión. No necesitas tocar esto activamente, el schedule corre automáticamente. Un grado más alto cuesta más memoria y ralentiza un poco el entrenamiento — pero es el precio para brillos realistas.

I34 Gaussians **DÓNDE**

Inspector → Metrics → Gaussians. Solo lectura.

 **TÉCNICO**

Número actual de Gaussianos en el modelo, formateado con separador local (p. ej., «524.318»). Crecimiento: - Classic: empieza en los puntos init de SfM (típicamente 50K-300K), crece mediante clone/split hasta justo antes de Densify Until, después estático hasta el final del entrenamiento (módulo pruning) - MCMC: los puntos sample se añaden hasta el tope MCMC, después solo relocalización
Valores finales sanos: - Classic Quality: 400K-700K (Horse 524K, Garden 800K) - MCMC Full: exactamente en el tope (predeterminado 150K, con multiplicador auto-scale × número de SfM, según escena 500K-1,5M)

Con MCMC, si el número cae a < 60 % del tope → anomalía (indicador de colapso, sugiere regularizadores demasiado agresivos).

 **EN POCAS PALABRAS**

Cuántos puntos Gaussianos tiene actualmente el modelo 3D. Crece durante el entrenamiento hasta alcanzar Densify Until (I20); después, el número se mantiene prácticamente constante. Más puntos = más detalle, pero también archivo más grande y renderizado más lento en la vista. 500.000 Gaussianos es una media típica para Classic Quality en una escena media; MCMC Full aterriza, según el auto-scale (I17), entre 500K y 1,5M. Si el número con MCMC de repente cae por debajo del 60 % del tope, es un indicador de colapso — normalmente regularizadores demasiado agresivos.

I35 GPU Memory **DÓNDE**

Inspector → Metrics → GPU Memory. Solo lectura.

 **TÉCNICO**

Estimación del consumo de memoria del buffer de Gaussianos como número de Gaussianos × 616 bytes (formateado en estilo memoria). 616 bytes es el tamaño empírico de un Gaussiano totalmente equipado (posición, escala, rotación, opacidad, coeficientes SH grado 3, acumulador de gradiente). La visualización **no** captura el overhead del renderer (buffer de tiles, buffer de sort, buffer backward) — el requisito real de memoria GPU es típicamente 2-3× este valor. Con 500K Gaussianos: mostrado ~290 MB, real ~700 MB. Con 1,5M Gaussianos: mostrado ~880 MB, real ~2,5 GB. En M3 Max con 64+ GB de memoria unificada es no crítico, en M3 Pro con 18 GB ya es un límite.

 **EN POCAS PALABRAS**

Una estimación de cuánta memoria GPU ocupan los propios Gaussianos — alrededor de 616 bytes por punto. El consumo real de GPU es 2-3× más alto que el mostrado, porque el renderer añade sus propios buffers de tile, sort y backward. En un MacBook con 16-18 GB de memoria unificada deberías mantenerte por debajo de 500K Gaussianos; con M3 Max o Studio (64+ GB) puedes ejecutar fácilmente 1,5M y más. Si el entrenamiento se cae de repente o el sistema hace swap, normalmente se ha alcanzado el límite aquí — baja Render Scale (I22) o reduce Densify Until (I20).

I36 Speed **DÓNDE**

Inspector → Metrics → Speed. Solo lectura.

 **TÉCNICO**

Iteraciones por segundo con un decimal («24.3 it/s»). Calculado por el entrenador como media móvil sobre las últimas ~100 iteraciones. Valores típicos:

- ajuste preestablecido Quick (1K iter): 80-120 it/s (corto, sin estado estable)
- Classic 20K @ 1,0 Render Scale (escena Truck, M3 Max): 25-35 it/s
- Classic 20K @ 0,5 Render Scale: 80-120 it/s
- MCMC 200K @ 0,5 Render Scale: 25-50 it/s (más lento por la relocalización)
- Con 1M+ Gaussianos y resolución completa: < 10 it/s

Una velocidad decreciente a lo largo del entrenamiento es normal — más Gaussianos = más cómputo por iteración. Las caídas repentinas (p. ej., de 30 → 5 it/s) indican thermal throttling de GPU o apps competidoras.

 **EN POCAS PALABRAS**

Lo rápido que está corriendo el entrenamiento, en iteraciones por segundo. Típicamente está en 20-50 it/s, con Render Scale reducido (50 %) y escenas pequeñas también 80-120 it/s. Cae a lo largo del entrenamiento de forma natural, porque más Gaussianos = más trabajo por iteración. Las caídas repentinas (p. ej., 30 → 5 it/s) indican thermal throttling de GPU o apps competidoras — pestañas de navegador con vídeo, copia de Time Machine, indexado de Fotos. Mantener la app en primer plano y cerrar programas en segundo plano suele ayudar. Con 1M+ Gaussianos y resolución completa, por debajo de 10 it/s es normal.

I37 Elapsed **DÓNDE**

Inspector → Metrics → Elapsed. Solo lectura.

 **TÉCNICO**

Tiempo ya transcurrido como «4:23» (m:ss) o «1:23:45» (h:mm:ss). Cambio de formato a partir de 1 hora. Mide solo el tiempo de entrenamiento puro, no las fases previas (cómputo de SfM, importación de imágenes). Con pause/resume el reloj sigue corriendo — es decir, es wall-clock, no tiempo de CPU.

 **EN POCAS PALABRAS**

Cuánto tiempo lleva corriendo el entrenamiento, como cronómetro puro (tiempo wall-clock). Formato es «m:ss» hasta una hora, después «h:mm:ss». No es «tiempo de CPU», sino «cuánto tiempo llevamos esperando» — así que los tiempos de pausa también cuentan. Mide solo la fase pura de entrenamiento, no el cómputo previo de SfM ni la importación de imágenes. Útil para comparar con el ETA (I38) — si Elapsed supera claramente el ETA original, el entrenamiento se ha vuelto más lento de lo previsto en algún punto.

I38 ETA DÓNDE

Inspector → Metrics → ETA. Solo lectura.

 TÉCNICO

Tiempo restante estimado como «17:42» o «1:12:35». Cálculo: (Max Iterations – iteración actual) / iteraciones por segundo. Muestra «-» cuando la velocidad es actualmente cero (al principio o durante una pausa). La estimación **no** se ajusta al ralentizamiento típico hacia el final del entrenamiento — especialmente con MCMC y Classic con valores grandes de Densify-Until, el entrenamiento tiende a volverse más lento porque entran cada vez más Gaussianos. En la realidad suele quedar 10-20 % por encima del ETA inicial.

 EN POCAS PALABRAS

Cuánto tiempo se espera que quede — calculado a partir de las iteraciones restantes y la velocidad actual (I36). Una estimación aproximada: si el Mac de repente se vuelve más lento (más Gaussianos de la fase de densify, thermal throttling, otras apps), puede durar más de lo mostrado. La app no tiene en cuenta el ralentizamiento típico hacia el final del entrenamiento, así que el final real suele caer un 10-20 % por encima del ETA inicial. Si le sumas un 15 % extra, suele encajar. Muestra «-» cuando la velocidad es ahora mismo 0 (inicio del entrenamiento o pausa).

Sección Loss Chart (I39–I41)

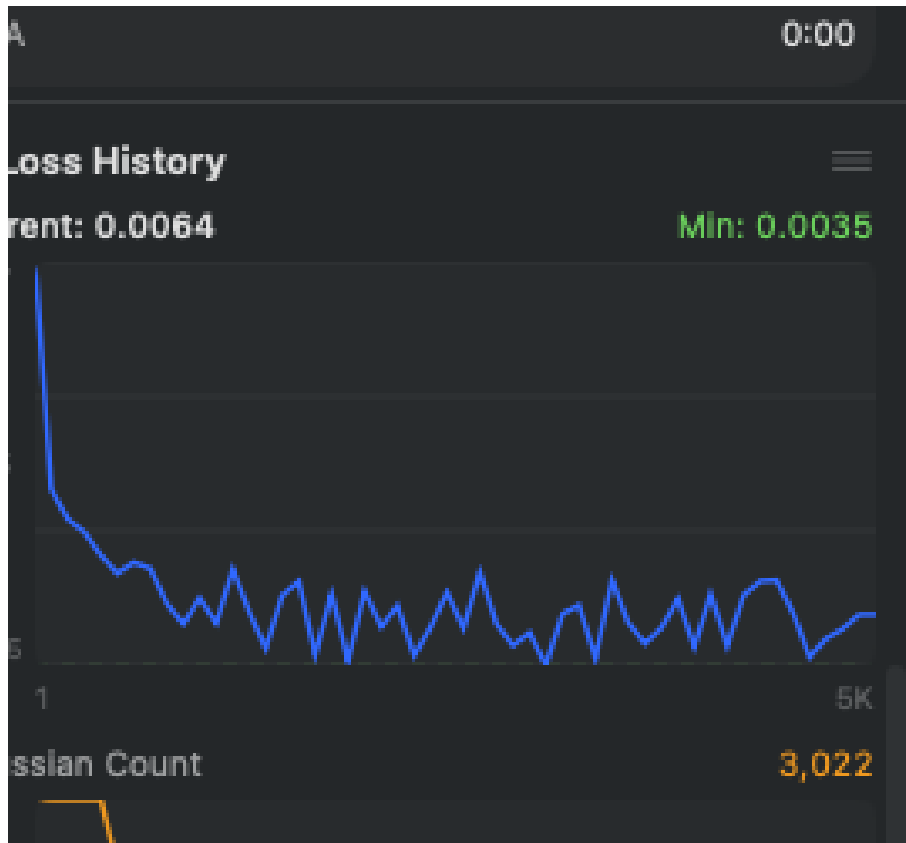


Figura 14: Recorte de solo la sección Loss History tras un entrenamiento completado — Current 0,0064, Min 0,0035 (verde), curva azul de 0,027 (iteración 1) a 0,0035 (iteración 5K) con codo característico hacia la iter 200, debajo gráfico Gaussian Count naranja

La sección Loss Chart visualiza el transcurso del entrenamiento a lo largo del tiempo. Consiste en dos gráficos: un gráfico Loss Curve (grande, arriba, azul) y un gráfico Gaussian Count (más pequeño, abajo, naranja). Ambos se construyen en vivo durante el entrenamiento y persisten hasta el siguiente inicio de entrenamiento. Antes del primer entrenamiento, el área está vacía («Loss curve will appear during training»). Los gráficos son dibujos de path SwiftUI puros (no el framework Swift Charts), así que renderizan con fluidez incluso con 100K+ puntos.

I39 Current Loss (visualización)**DÓNDE**

Inspector → Loss Chart → área de etiqueta izquierda «Current: 0.0287». Solo lectura.

TÉCNICO

Valor float del último sample point de loss, formateado con cuatro decimales. Idéntico a I31 (Loss en la sección Metrics), solo que aquí formateado de forma más compacta. La fuente es el historial de loss — una lista que recibe una entrada cada ~30 iteraciones. Solo se añaden valores finitos a la lista — NaN/Infinity (muy raros, en el caso de un bug de explosión de gradiente) se filtran.

EN POCAS PALABRAS

El valor de loss actual en notación más corta que en la sección Metrics (cuatro decimales). De contenido idéntico a I31, pero aquí la visualización está justo en el gráfico de loss y te da el valor numérico exacto al observar la curva. Como todas las métricas en vivo, se actualiza cada ~30 iteraciones. Los valores NaN o Infinity (extremadamente raros con bugs de explosión de gradiente) la app los filtra automáticamente. Útil al mirar el gráfico para no tener que saltar a la otra sección.

I40 Min Loss (visualización)**DÓNDE**

Inspector → Loss Chart → área de etiqueta derecha «Min: 0.0245» (verde). Solo lectura.

TÉCNICO

Mínimo de todos los valores de loss vistos jamás del run de entrenamiento actual. Recalculado en vivo a partir del historial de loss — sin persistencia separada. Visualizado con texto verde, porque «Min» = «Best so far». La línea verde discontinua en el borde inferior del gráfico marca esta posición y visualmente. Con sesiones de continue-training, el seguimiento del min se reinicia — el historial antiguo es reemplazado en la UI por el nuevo (no anexado). Si el entrenamiento actual corre peor que el previo, la visualización del min puede ser mayor que el resultado final anterior.

EN POCAS PALABRAS

El valor de loss más bajo que este entrenamiento ha visto hasta ahora — mostrado en verde, porque «el mejor hasta el momento». La línea verde discontinua en el borde inferior del gráfico también marca esta posición visualmente. Si la curva actual está ahora claramente por encima, con suerte aún hay una mejora; normalmente Min es el indicador de resultado final que te importa después. Con sesiones de continue-training, el seguimiento del min se reinicia, porque el historial antiguo es reemplazado en la UI por el nuevo — el valor min puede por tanto verse peor que el resultado final anterior.

I41 Gráfico Gaussian Count

DÓNDE

Inspector → Loss Chart → segundo gráfico debajo (naranja). Solo lectura.

TÉCNICO

Gráfico de línea del número de Gaussianos sobre las iteraciones de entrenamiento. Fuente: el historial del número de Gaussianos (lista de pares (iter, count), rellenado por el entrenador cada ~30 iter). Escala Y dinámica entre mínimo y máximo del historial. Con la estrategia Classic la curva típicamente tiene este aspecto: subida sostenida hasta Densify Until, luego plana (con pequeñas fluctuaciones de pruning). Con MCMC: subida pronunciada hasta el tope, después línea horizontal (la relocalización mantiene el número constante). Si la curva **cae** pese a entrenamiento activo, la densificación está podando con demasiada agresividad — indicio de defaults incorrectos o un bug conocido de colapso MCMC (tema del hotfix v1.4.4).

EN POCAS PALABRAS

Cómo se desarrolla el número de Gaussianos a lo largo del tiempo de entrenamiento — el gráfico naranja más pequeño bajo la curva de Loss. Con la estrategia Classic, la línea sube sostenidamente hasta alcanzar Densify Until (I20), después se mantiene plana con pequeñas fluctuaciones de pruning. Con MCMC dispara con fuerza hasta el tope y después se mantiene horizontal, porque la relocalización mantiene el número constante. Si la curva, pese a entrenamiento activo, de repente cae hacia abajo, la densificación está siendo demasiado agresiva con el pruning — señal clásica del bug de colapso MCMC de la v1.4.4. Entonces ayuda la actualización de la app o volver a Classic.

¿Cómo leer la curva de loss?

El gráfico de loss es la herramienta de diagnóstico más importante del Inspector — ningún otro indicador muestra de manera tan directa si el entrenamiento avanza de forma útil o está estancado. La forma sana típica es una caída rápida en las primeras 1000-3000 iteraciones (de ~0,15 a ~0,05), seguida de un descenso lento y sostenido hasta el final del entrenamiento (hasta 0,020-0,030). De forma logarítmica, la curva parece una diagonal suave.

¿Qué significa una meseta en el loss? Si la curva permanece plana durante varios miles de iteraciones, hay dos lecturas posibles: (a) el entrenamiento ha «convergiado» — el loss ya no puede bajar de forma significativa, porque el modelo es tan bueno como puede ser con los datos y ajustes dados. Eso es lo deseado; está «listo». (b) El entrenamiento «se cuelga» — el loss aún podría bajar, pero la optimización se estanca (mínimo local, learning rate demasiado bajo, densificación desactivada). Para distinguir: si el valor de loss cae en un rango típicamente bueno (0,020-0,030 con indoor/objeto, 0,040-0,060 con outdoor) y la curva está plana desde las 5K iteraciones, ha convergiado. Si el valor es claramente más alto que con escenas comparables (p. ej., 0,08), se ha colgado.

Nota: meseta de Gaussianos ≠ meseta de loss. Una meseta en el número de Gaussianos **no** significa «el entrenamiento está hecho». Solo significa que la densificación ha dejado de añadir nuevos puntos — ya sea porque se ha alcanzado (Classic) o porque el tope MCMC está lleno. El entrenamiento continúa después y solo refina los puntos

existentes. La señal real de «listo» la lees en la curva de loss y en la visualización de Iteración (I30), no aquí.

Regla práctica para abortar: Si la curva de loss tras 5000+ iteraciones está por encima de 0,08 y apenas baja, con alta probabilidad la reconstrucción de SfM está torcida. Aborta el entrenamiento, mira en el Capítulo 9 si el backend de SfM elegido encaja con la escena, posiblemente cambia a COLMAP/Native, después reinicia. Mejor invertir 10 minutos en un SfM mejor que 2 horas de entrenamiento con una mala alineación de cámara.

¿Cuándo recurrir al Inspector?

Referencia rápida: ¿qué sección + qué controles para qué caso de uso típico?

Tarea habitual	Sección	IDs de control
Desaturar los colores del splat terminado	Look	L1 (Saturation)
Redondear splats de aguja/confeti	Look	L2 (Splat length)
Rellenar nube agujereada / agrandar splats	Look	L3 (Splat size)
Desvanecer el «confeti lejano» en órbitas	Look	L4 (Fade far region)
Descartar los ajustes de Look	Look	L5 (Reset finishing)
Cargar un setup prefabricado	Presets	I7 (clic en fila)
Guardar un setup propio	Presets	I1 → I2 → I4
Compartir un setup con un compañero	Presets	I5 (Export) o I6 (Import)
Cambiar backend de SfM (p. ej., porque Apple-PG es muy inestable)	Training Configuración	I12 (véase Cap. 9)
Procesar fotogramas de vídeo sin focal EXIF	Training Configuración	I13 (FOV Override)
Rendimiento de COLMAP: GLOMAP en lugar de Classic	Training Configuración	I14
Cambiar de Classic a MCMC	Training Configuración	I15
Dejar correr el entrenamiento más tiempo	Training Configuración	I18 (Max Iter) + I20 (Densify Until) — acoplados vía I19
Reducir a la mitad el tiempo de GPU	Training Configuración	I22 (Render Scale al 50 %)
Calidad de entrenamiento +6 % (MCMC)	Training Configuración	I16 (MCMC Quality)
Escena exterior con muchos puntos SfM	Training Configuración	I17 (Auto-scale by scene)
Configurar / cambiar la ruta de COLMAP	Training Configuración	I23 / I24 / I25
Hacer los archivos de exportación más pequeños	Enhancements	I26 (déjalo siempre activo)
Vista más nítida sin más tiempo de entrenamiento	Enhancements	I27 (Viewport Scaling → MetalFX)
MetalFX suaviza demasiado → alternativa	Enhancements	I27 (Viewport Scaling → Lanzos)
El último ápice de detalle con estructuras finas	Enhancements	I29 (Percentual Loss 0.05-0.1)

El último ápice de detalle con estructuras finas

Enhancements
Metrics Line Chart

67 / 242

I29 (Percentual Loss 0.05-0.1)
I29 (Percentual Loss 0.05-0.1)
(tendrás que tenerlo activo)

CAPÍTULO

Capítulo 3 — Ajustes

La ventana de ajustes se abre con `RadianceKit` → `Settings...` o con el atajo de teclado estándar `⌘,`. Contiene dos pestañas: **General** y **AI Helpers**. A diferencia de los valores del Inspector del Capítulo 2, los ajustes de esta ventana tienen efecto **a nivel de aplicación** (en todos los proyectos) — se persisten y sobreviven al reinicio de la app. La pestaña General agrupa tres secciones de contenido: Interfaz, Vista y Entrenamiento. (Los tres conmutadores de reducción de floaters en exteriores — Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome — que antes vivían aquí se trasladaron, a partir de la v1.6, a la sección Enhancements del Inspector experto, donde ahora se almacenan por proyecto; véase el Capítulo 2, 142–144.) La pestaña AI Helpers habilita los asistentes de aprendizaje automático on-device (Vision, CoreML) para el preprocesado de SfM y entrenamiento.

Los controles anteriores para habilitar o deshabilitar conjuntamente todos los AI Helpers ya no existen en la versión actual — por eso no se documentan aquí. La antigua zona «Coming Soon» para asistentes aún no entregados también se ha eliminado y no se referencia aquí.

Pestaña General

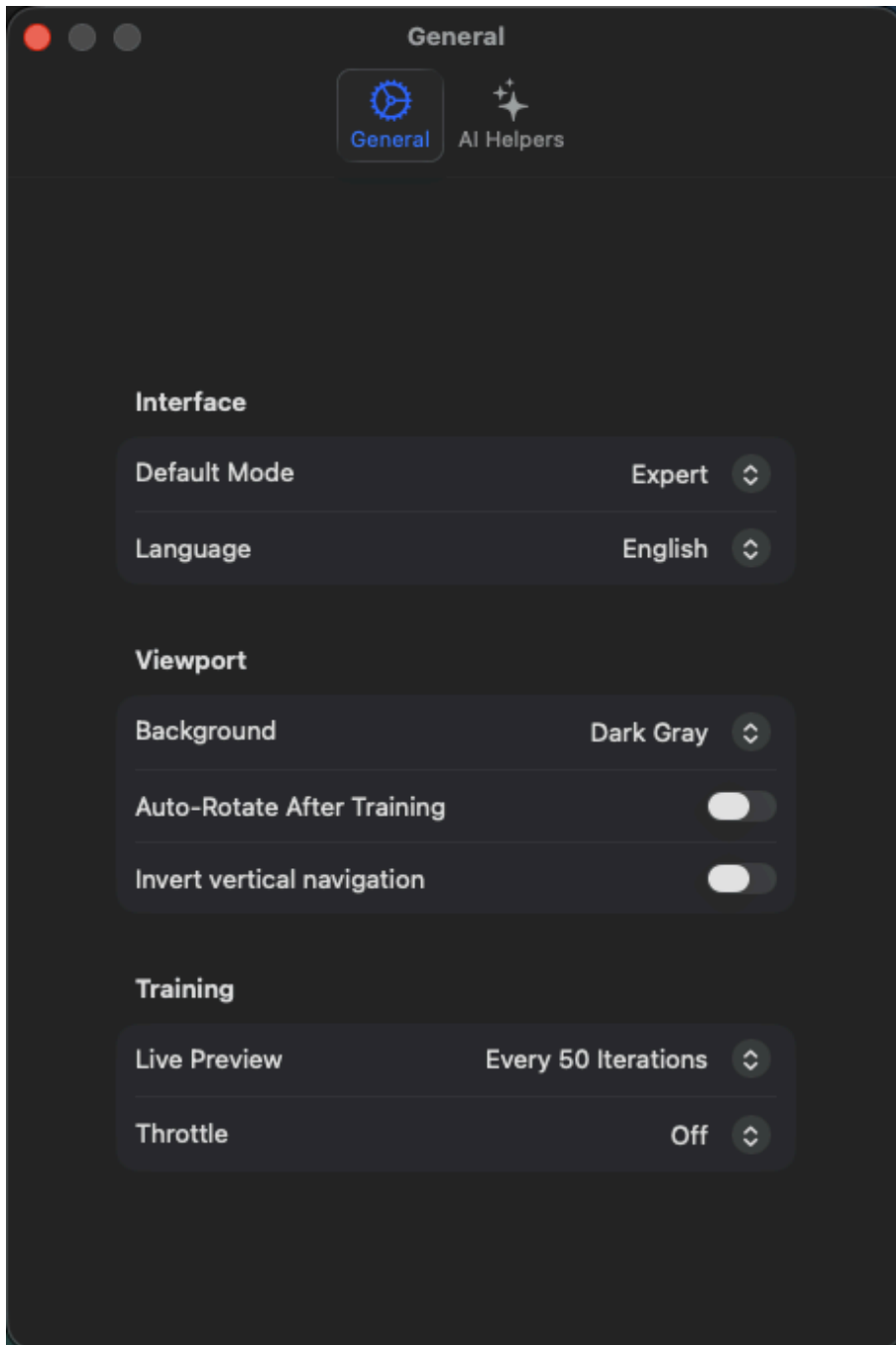


Figura 15: Ajustes → pestaña General con las secciones Interfaz, Vista, Entrenamiento y Experimental

S1 Modo predeterminado

DÓNDE

Settings → General → Interface → Selector Default Mode. Bound.: Predeterminado: `.simple`.

TÉCNICO

Controla en cuál de los dos modos de UI se abre la app tras el próximo arranque. «Simple Mode» es el flujo de asistente guiado en 4 pasos (Importar → Procesar → Previsualizar → Exportar, documentado en el Capítulo 10 bajo Z1–Z4), «Expert Mode» el diseño clásico de tres paneles con Navegador, Vista 3D e Inspector experto del Capítulo 2. El valor se recuerda entre reinicios. Efecto idéntico al menú Mode → Simple Mode (⌘1) / Mode → Expert Mode (⌘2), salvo que el menú cambia la sesión en curso mientras este selector establece el predeterminado para sesiones futuras. Ambos modos acceden al mismo estado del proyecto — los proyectos, las cámaras y la configuración de entrenamiento se conservan al cambiar de modo. Los botones de la barra de herramientas específicos de cada modo se renderizan de inmediato.

EN POCAS PALABRAS

Aquí eliges con qué interfaz arranca RadianceKit en el próximo lanzamiento. «Simple Mode» es el modo principiante: cuatro pasos claros, ajustes preestablecidos predefinidos, apenas opciones. «Expert Mode» es el layout completo de caja de herramientas con todos los controles que ves en el Capítulo 2. Puedes alternar en cualquier momento mediante el menú «Mode» sin perder imágenes ni progreso de entrenamiento.

S2 Idioma

DÓNDE

Settings → General → Interface → Selector Language. Bound:.. Predeterminado: `.system` (sigue el idioma de macOS).

TÉCNICO

Selecciona el idioma de visualización de toda la UI de la app, independientemente del idioma del sistema macOS. RadianceKit está localizado en 17 idiomas (`de` , `en` , `pl` , `en-AU` , `ar-SA` , más 12 adicionales). Cuando se ajusta a «System», la app sigue al idioma de macOS. Con una elección explícita, el ajuste de idioma se recuerda entre reinicios; el efecto completo normalmente requiere reiniciar la app, porque los paquetes de localización solo se cargan al inicio. Se tienen en cuenta las 298 claves de localización documentadas en el proyecto, incluidos todos los textos en sub-vistas y tooltips de ayuda.

EN POCAS PALABRAS

Si tu Mac está en inglés pero prefieres la interfaz de RadianceKit en alemán (o al revés), aquí lo configuras. La mayoría de los textos cambian de inmediato. Algunos diálogos solo aparecen en el nuevo idioma tras reiniciar la app.

S3 Fondo de la vista

DÓNDE

Settings → General → Viewport → Selector Background. Bound:.. Predeterminado: `.darkGray` (RGB 0,1, 0,1, 0,1).

TÉCNICO

Establece el color de fondo predeterminado para la vista 3D. Tres opciones: «Dark Gray» (RGB 0,1, 0,1, 0,1 — predeterminado), «Black» (0, 0, 0) y «White» (1, 1, 1). El ajuste persiste el predeterminado para nuevos proyectos y sesiones entre reinicios y al mismo tiempo actualiza el renderer Metal en ejecución de inmediato. Las mismas opciones se encuentran en el menú Viewport → Background (M21, M22, M23), pero el selector de ajustes define el predeterminado, mientras que el menú cambia la visualización en ejecución. Importante para capturas y vídeos de demostración: los fondos blancos resaltan más los floaters verdes/azules, los fondos oscuros son mejores para capturas de render limpias.

EN POCAS PALABRAS

El color detrás de tus modelos 3D en la ventana de previsualización. Gris oscuro es el predeterminado y va bien con la mayoría de las escenas. El blanco es bueno para capturas de pantalla, el negro queda más elegante para capturas de render. Puedes cambiar el color en cualquier momento desde el menú «Viewport → Background» para la escena en curso — este ajuste solo define qué color vuelve a estar activo la próxima vez que la abras.

S4 Auto-rotar tras el entrenamiento

DÓNDE

Settings → General → Viewport → Conmutador «Auto-Rotate After Training». Bound.: Predeterminado: `false`.

TÉCNICO

Inicia una rotación de plato giratorio continua de la cámara de la vista alrededor del centroide de la escena inmediatamente tras terminar el entrenamiento (tasa de rotación predeterminada $\sim 0,3$ rad/s). Útil en la práctica para sesiones de demostración, comparaciones A/B y para valorar al instante desde una vista 360° si se han formado «floaters» en el borde de la escena. El efecto es visualmente idéntico al menú Viewport → Toggle Auto-Rotation (M16, $\text{⌘} \backslash T$), salvo que este conmutador dispara el comportamiento de forma automática tras el final del entrenamiento, en vez de manualmente. Se puede interrumpir en cualquier momento desde el menú o haciendo clic en la vista (lo que pausa la rotación). No influye en el rendimiento de entrenamiento — la rotación solo se ejecuta cuando termina el entrenamiento.

EN POCAS PALABRAS

Cuando está activado, la escena 3D gira automáticamente en cuanto termina el entrenamiento — como un tiovivo. Resulta agradable si lanzas un entrenamiento durante la noche y por la mañana quieres ver el resultado ya en movimiento sin tener que hacer clic. Para sesiones largas en las que solo monitorizas el entrenamiento, déjalo desactivado.

S5 Intervalo de previsualización en vivo

DÓNDE

Settings → General → Training
→ Selector Live Preview. Bound:
AppState.trainingConfig.livePreviewInterval.
Predeterminado: 0 (Off).

TÉCNICO

Determina el intervalo de iteraciones al que se renderiza la instantánea de entrenamiento en curso en la vista 3D. Cuatro valores discretos: 0 («Off»), 50, 250, 1000 iteraciones. Con la previsualización en vivo activa, el entrenador copia el buffer de Gaussianos de la GPU a un buffer de render separado y dispara un redibujado de la vista. Con «Off», la vista solo se actualiza tras completarse el entrenamiento. Coste de rendimiento: cada 50 iteraciones, ~5–10 % más lento en M3 Ultra; cada 250, ~1–2 % más lento; cada 1000, no medible. Sobrecarga de memoria constante ~2 GB para el buffer de instantánea, independientemente del intervalo. El valor sirve como predeterminado para nuevos entrenamientos; tras el inicio del entrenamiento, el Inspector de entrenamiento muestra el valor real en vivo del entrenamiento en curso. Con un intervalo de 50 la impresión visual es un «crecimiento» fluido de la nube de puntos; con 1000 se ve a saltos.

EN POCAS PALABRAS

Mientras el entrenamiento se está ejecutando, puedes escoger con qué frecuencia se actualiza la vista 3D. «Off» significa: sin actualización durante el entrenamiento (lo más rápido). «Every 50 Iterations» muestra casi en tiempo real cómo se va formando tu escena (ligeramente más lento). Para mirar cómodamente entrenamientos pequeños, «Every 250» es un buen compromiso.

S6 Retardo de throttle



DÓNDE

Settings → General → Training → Selector Throttle.
Bound: `AppState.trainingConfig.throttleDelayMs`.
Predeterminado: 0 (Off).



TÉCNICO

Inserta un retardo artificial en milisegundos entre iteraciones de entrenamiento. Cuatro valores discretos: 0 («Off»), 2 («Light»), 5 («Moderate»), 10 («Eco»). Propósito: durante entrenamientos largos (varias horas), la GPU estaría si no al 100 % de uso, lo que provoca una UI del sistema notablemente más lenta (el cursor se entrecorta, otras apps van pesadas). El retardo de throttle da a la GPU pausas en las que pueden ejecutarse otras tareas. El coste de rendimiento es considerable: con 5 ms de throttle, un entrenamiento típico de 40K tarda alrededor de un 50–80 % más que sin throttle. En modo de rendimiento «Eco» (10 ms), el retardo por iteración es mayor que la propia iteración — factor de 2–3× más lento. Con throttle activo, aparece bajo el selector el aviso: «Throttle is on. Training will be slower than usual.» La propia app no reacciona notablemente mejor — solo se benefician otras apps.



EN POCAS PALABRAS

Si tu Mac se calienta demasiado durante un entrenamiento largo u otros programas se vuelven demasiado pesados, activa aquí un freno. «Off» da gas a fondo a la GPU (lo más rápido). «Light» hace una pequeña pausa entre cada paso (un poco más lento, pero el sistema reacciona mejor). «Eco» es el freno más fuerte — bueno para entrenamientos nocturnos en un MacBook que no debe calentarse demasiado.

Pestaña AI Helpers

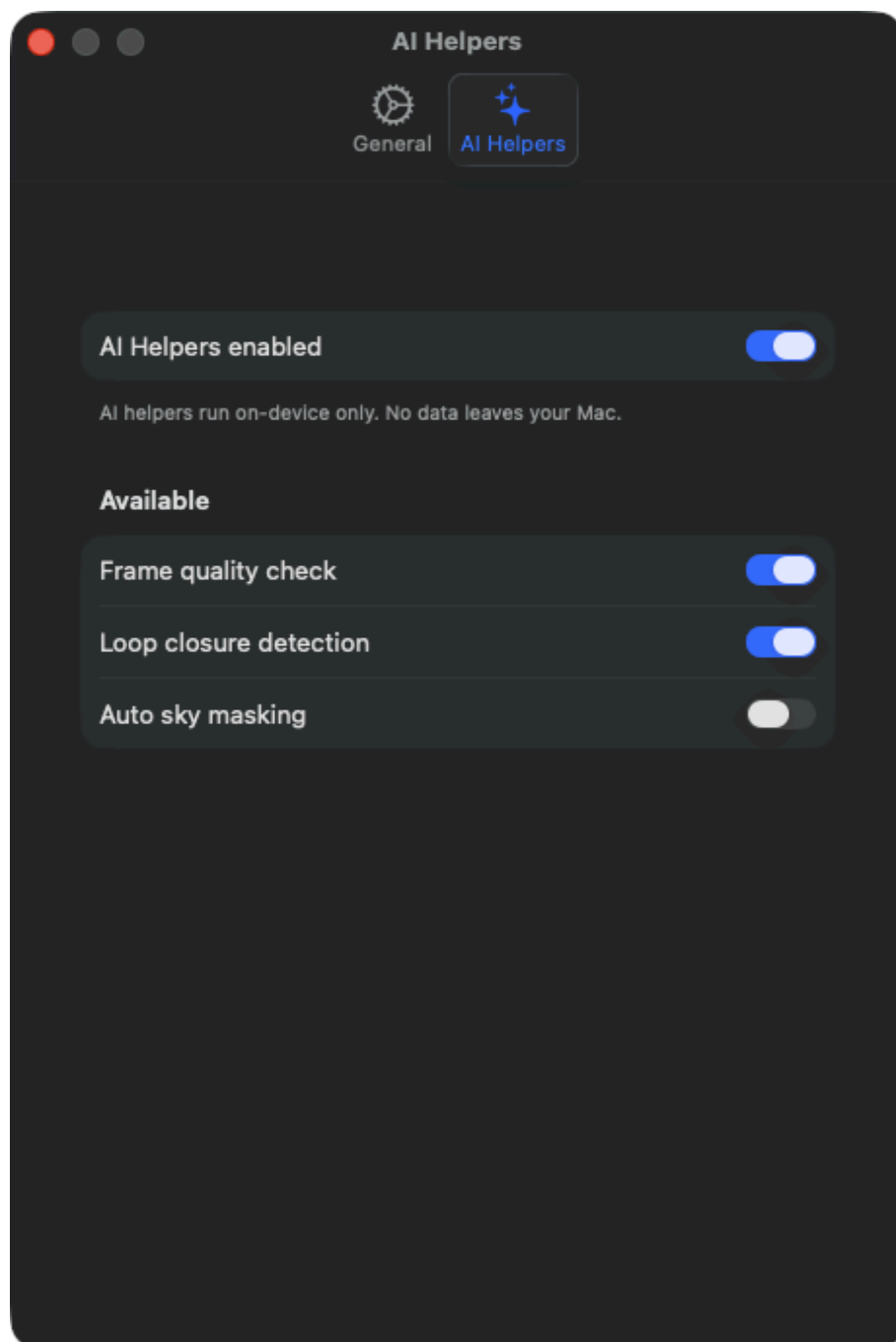


Figura 16: Ajustes → pestaña AI Helpers con el interruptor maestro y los sub-conmutadores

S11 AI Helpers habilitados (maestro) **DÓNDE**

Settings → AI Helpers → primera sección → Conmutador «AI Helpers enabled». Bound: Predeterminado: `true`.

 **TÉCNICO**

Interruptor maestro sobre todas las funciones AI Helpers en la pipeline. Cuando está desactivado, la pipeline de importación y SfM omite por completo todas las etapas de preprocesado basadas en ML — sin llamada a Apple Vision, sin carga de modelo CoreML, sin despertar al NPU. Cuando está activado, se consultan los sub-conmutadores individuales (S12–S13). El valor se recuerda entre reinicios. Afecta a las siguientes etapas: (a) comprobación previa de calidad de fotograma antes de SfM (S12), (b) detección de cierre de bucle (S13). Importante: cuando está desactivado, los dos sub-conmutadores se inhabilitan y aparecen visualmente atenuados. La nota al pie subraya que todos los AI Helpers se ejecutan estrictamente on-device — sin subida de imágenes, sin procesamiento en la nube. La garantía de privacidad proviene del uso exclusivo del framework Apple Vision (localmente en el Neural Engine) y de modelos CoreML que se entregan directamente en el bundle de la app.

 **EN POCAS PALABRAS**

El interruptor principal para todas las funciones que internamente usan IA / aprendizaje automático. El predeterminado es «activado», porque los asistentes ahorran mucho tiempo sin que tus imágenes salgan del Mac. Si los quieres totalmente desactivados (p. ej., para ahorrar energía o porque tu Mac no tiene NPU), desactívalos aquí — las dos subopciones de abajo se atenuan automáticamente y dejan de hacer nada.

S12 Comprobación de calidad de fotograma **DÓNDE**

Settings → AI Helpers → Sección Available → Conmutador «Frame quality check». Bound: Predeterminado: `true`.

 **TÉCNICO**

Activa el screener de calidad de fotograma (Fase 3.11), que analiza cada fotograma importado antes de la llamada a SfM. Pasos de la pipeline por fotograma: (a) filtro de varianza laplaciana de Apple Vision (detección de desenfoque — umbral ~150), (b) comprobación basada en histograma de sobre/sub-exposición (umbral: >5 % de píxeles en 0 o 255), (c) detección de fotograma en blanco (desviación estándar < 5 sobre todos los píxeles). Los fotogramas que pasan las tres comprobaciones siguen adelante directamente. Los que fallan al menos una comprobación disparan un diálogo modal de confirmación que enumera cada fotograma problemático con miniatura y motivo y pregunta si debe eliminarse. Importante: sin borrado automático — el diálogo siempre es obligatorio, el usuario conserva la decisión final. Rendimiento: ~50 ms por fotograma en M3 Ultra, ejecutado en paralelo. Cuando está desactivado, todos los fotogramas se reenvían a SfM sin comprobar. Con el maestro (S11) desactivado, este conmutador aparece atenuado y no tiene efecto. Estado entregado según la memoria: SHIPPED 2026-05-23.

 **EN POCAS PALABRAS**

Antes del entrenamiento propiamente dicho, la app mira cada foto: ¿está borrosa?, ¿totalmente oscura o blanca?, ¿en blanco? Si es así, te pregunta si quieres descartar la imagen — nunca elimina nada automáticamente. Esto ahorra muchas horas más adelante, porque una sola imagen totalmente borrosa puede arruinar a veces todo el entrenamiento. El predeterminado es «activado», porque el coste es casi cero y el beneficio, grande.

S13 Detección de cierre de bucle

DÓNDE

Settings → AI Helpers → Sección Available → Conmutador «Loop closure detection». Bound: Predefinido: `true`.

TÉCNICO

Activa la detección de cierre de bucle basada en feature print de Apple Vision. Para cada fotograma importado se calcula un vector de características de ~768 dimensiones que representa un embedding neuronal del contenido de la imagen. Todos los feature prints se comparan luego por pares mediante similitud coseno. Los pares con similitud $> 0,85$ y distancia de índice de fotograma > 50 (es decir, fotogramas no adyacentes) se identifican como «candidatos a cierre de bucle» y se escriben en un archivo sidecar JSONL dentro de la carpeta del proyecto. Solo informativo — la secuencia de imágenes importada no se modifica. Propósito: da al solver de SfM (especialmente COLMAP) una pista de que esos fotogramas se agrupan en el espacio 3D. En el SfM nativo, la información del sidecar es actualmente solo documental; COLMAP usa las pistas internamente mediante un archivo custom matches (integración manual posible, no cableada automáticamente). Rendimiento: ~200 ms por fotograma en M3 Ultra, en paralelo. Cuando está desactivado, no se generan feature prints. Con el maestro (S11) desactivado, aparece atenuado.

EN POCAS PALABRAS

Cuando caminas alrededor de un objeto fotografiando y acabas otra vez en el punto de partida, le ayuda enormemente al ordenador saberlo. Esta opción detecta automáticamente cuáles de tus fotos se tomaron «casi desde el mismo sitio» y lo escribe en un pequeño archivo auxiliar. Las herramientas de SfM (especialmente COLMAP) pueden usar esa información para entregar una reconstrucción 3D más limpia. El predeterminado es «activado», porque corre sin tu intervención y no cambia nada en tus imágenes.

Ajustes espejo del Inspector

Las entradas de ajustes restantes (S17–S33) de la tabla de inventario son espejos del Inspector experto y están documentadas en el Capítulo 2 (controles del Inspector I12–I29). No aparecen físicamente en la ventana de Ajustes, sino que se listaron en el inventario solo porque se ejecutan a través de propiedades de `TrainingConfig` que se persisten mediante y, por tanto, tienen formalmente carácter de ajuste. Para explicaciones de fondo, véase allí.

¿Cuándo qué?

Ajuste	Ámbito	Persistencia
S1 Modo predeterminado	Global de la app	Reinicio de la app
S2 Idioma	Global de la app	Reinicio de la app
S3 Fondo de la vista	Global de la app (predeterminado) + runtime	Reinicio de la app
S4 Auto-rotar tras entrenamiento	Global de la app	Reinicio de la app
S5 Intervalo de previsualización en vivo	Predeterminado para nuevos entrenamientos	Reinicio de la app
S6 Retardo de throttle	Predeterminado para nuevos entrenamientos	Reinicio de la app
S11 Maestro AI Helpers	Global de la app	Reinicio de la app
S12 Comprobación de calidad de fotografía	Global de la app	Reinicio de la app
S13 Detección de cierre de bucle	Global de la app	Reinicio de la app

Global de la app = afecta a todos los proyectos. Predeterminado para nuevos entrenamientos = solo afecta al siguiente entrenamiento creado; las sesiones en curso no cambian. Entrenamiento actual = se aplica de inmediato a la configuración de entrenamiento en curso, pero no se persiste sin reimportación explícita.

CAPÍTULO

Capítulo 4 — Ventanas auxiliares

Además de la ventana principal (vista 3D más Inspector), RadianceKit gestiona siete ventanas adicionales, todas abribles desde el menú Help. La lista de arriba a abajo: User Guide (⌘?), Keyboard Shortcuts (⌘/), Open Training Logs... (no abre una ventana de la app sino el Finder; por eso no se trata más aquí), Manage Storage..., Pareto Dashboard... (⇧⌘D), Holdout Analysis... (⇧⌘H), BayesOpt Console... (⇧⌘B). Tres de ellas — Dashboard, Holdout, BayesOpt — son herramientas de análisis autónomas. Cada una tiene su propia pila view-model, lee o escribe archivos JSON en disco, y existe un argumento CLI por cada una que permite apuntar la ventana a un archivo concreto directamente al arrancar la app (`--dashboard-dir` , `--holdout-file` , `--bayesopt-autorun`).

Las cuatro ventanas simples (User Guide, Keyboard Shortcuts, Manage Storage, más las entradas del submenú Open Training Logs / Open Exports Folder) reciben una entrada corta por control. Las tres ventanas de análisis se documentan con más detalle — cada una con una introducción que explica qué ves en la ventana, cuándo deberías abrirla y cómo interpretar la imagen mostrada.

Al final del capítulo hay una sección de referencia cruzada al Inspector de la ventana principal: qué puedes leer de forma significativa del gráfico de loss en vivo y de la visualización del número de Gaussianos durante un entrenamiento en curso.

User Guide (W1–W4)

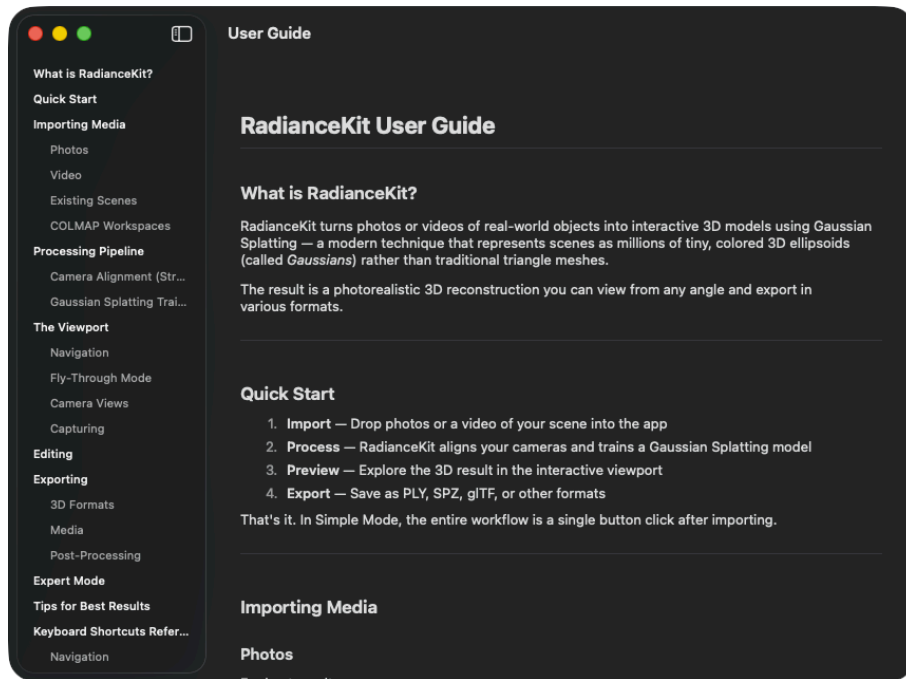


Figura 17: Ventana User Guide con barra lateral a la izquierda y contenido Markdown renderizado a la derecha

Qué es: Una ventana de ayuda integrada que renderiza el `guide_<language>.md` entregado con la app. El idioma se deriva de los Ajustes (pestaña General → Idioma) o, si allí está «System», de las preferencias de idioma de macOS. El layout es clásico: barra lateral con todos los títulos a la izquierda, texto del cuerpo a la derecha.

CUÁNDO ABRIRLA Cuando necesitas un recordatorio rápido de un único punto — como sustituto de palabra clave. La referencia completa es este manual; la ventana de ayuda integrada se parece más a lo que sería un `--help` en la línea de comandos. Se actualiza con cada lanzamiento de la app, pero está mantenida deliberadamente más superficial en contenido.

W1 NavigationSplitView (Sidebar + Detail)



Help → User Guide (⌘?).



Layout de dos columnas con una barra lateral estrecha (al menos 180 pt de ancho) para el árbol de contenido y un área de detalle desplazable para el contenido Markdown real. La ventana tiene un tamaño mínimo de 700 × 500 pt. En la primera apertura, la ventana carga el `guide_<lang>.md` correspondiente del bundle de la app (fallback `guide_en.md`), lo parsea en registros de bloque (títulos H1–H4, párrafos, listas, tablas, separadores) y extrae por separado la estructura de títulos para la barra lateral. El formato inline (negrita, cursiva, code-span) se renderiza mediante el motor Markdown integrado. El idioma se lee de los ajustes de la app, con el caso especial del chino (`zh-Hans`) y el portugués brasileño (`pt-BR`), que se mantienen como tags de locale completos porque esas variantes difieren respectivamente del zh y del pt.

EN POCAS PALABRAS

El texto de ayuda integrado, con la lista de temas a la izquierda y el contenido a la derecha. El idioma se ajusta automáticamente según los ajustes del sistema. Funciona offline, pero está pensado deliberadamente como versión corta — la referencia completa es este manual.

W2 List (barra lateral de títulos)



Columna izquierda en la ventana User Guide.



Lista de todos los títulos H2 y H3 del documento Markdown actual. Las entradas H2 aparecen sin sangría con peso Medium, las H3 con 16 pt de sangría a la izquierda y un estilo de foreground reducido. H4 y más profundos se ignoran, porque la profundidad haría la barra lateral confusa. Los IDs de anchor se generan a partir del texto del título mediante slugificación (minúsculas + espacios a guiones + filtrado de letras/números/guiones — el mismo algoritmo que usa GitHub para sus anchors Markdown, así que URLs externas a la doc también podrían aterrizar en el mismo anchor). La lista usa el estilo nativo de macOS.

EN POCAS PALABRAS

La barra de navegación a la izquierda. Toca una entrada y saltas a la sección.

W3 Button (título → salto a anchor)

DÓNDE

Un botón por fila de la barra lateral.

TÉCNICO

Cada entrada de la barra lateral es un botón que fija el anchor actual, pero visualmente se ve como una entrada de lista. Una variable observadora dispara entonces el salto de scroll al anchor correspondiente con una animación suave de 0,3 s. Tras el salto, el valor del anchor se restablece para que el siguiente clic sobre el mismo anchor dispare otra vez (de lo contrario el observador no volvería a dispararse porque el valor no habría cambiado).

EN POCAS PALABRAS

Hacer clic te lleva al punto correspondiente del texto a la derecha.

W4 ScrollView (contenido de detalle)

DÓNDE

Columna derecha.

TÉCNICO

Área de contenido desplazable, apilada verticalmente, con renderizado lazy, porque guías más largas pueden tener fácilmente más de 200 bloques Markdown — una variante no lazy los instanciaría todos a la vez. Cada bloque recibe su propio ID, ya sea el anchor del título (para H1–H3 saltables) o un placeholder de índice. El ancho máximo es de 720 pt, padding 32 horizontal / 24 vertical, así que las líneas largas mantienen un layout bien legible. Las tablas se renderizan celda a celda con stacks horizontales y separadores; el código inline mediante el motor Markdown integrado. Los bloques de código reales se tratan actualmente como párrafos — una limitación conocida de la ventana Help.

EN POCAS PALABRAS

El texto de ayuda en sí. Desplazable, ancho bien legible, tipografía clara.

Keyboard Shortcuts (W5–W6)

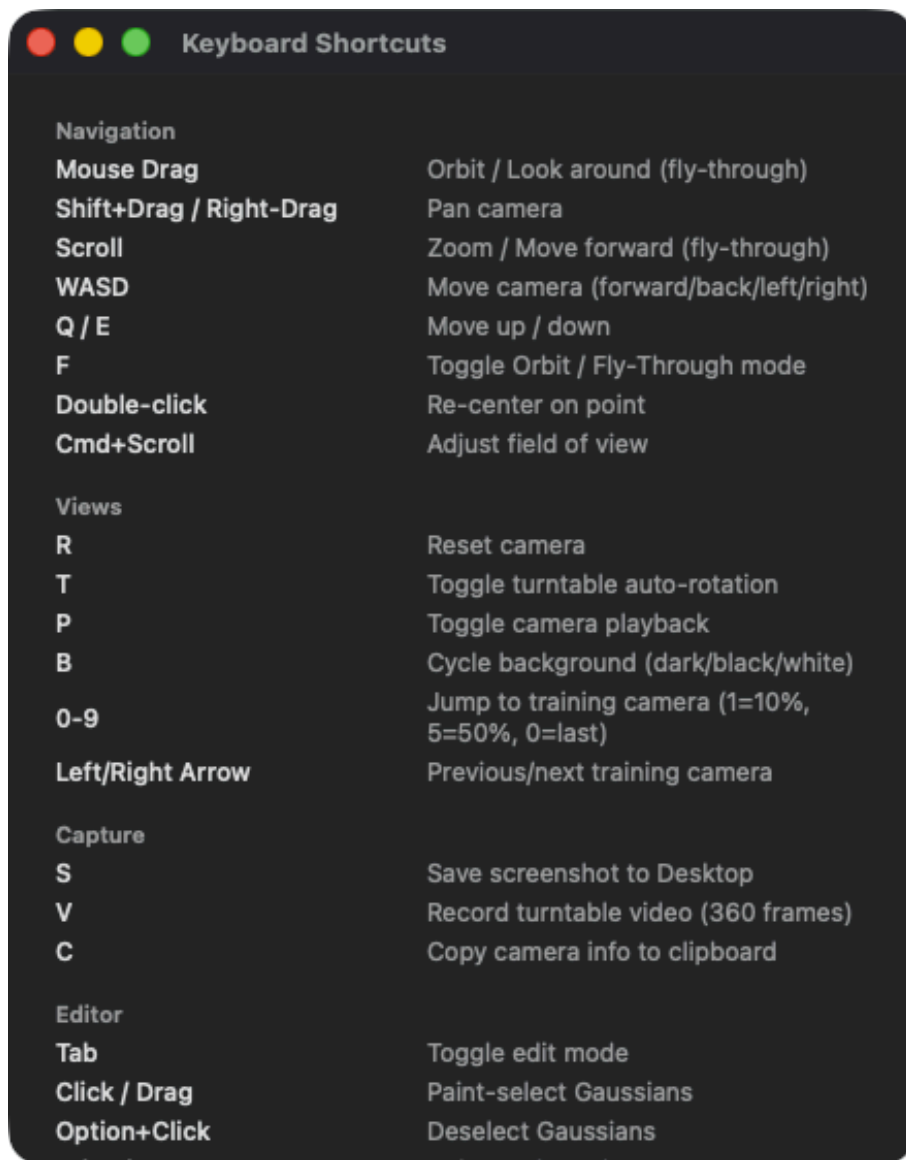


Figura 18: Ventana Keyboard Shortcuts — cinco grupos Navigation/Views/Capture/Editor/Training con columna de hotkey a la izquierda y descripción a la derecha

LO QUE SE VE EN LA IMAGEN Lista de referencia estática en cinco secciones. **Navigation:** Mouse Drag (Orbit/Fly), Shift+Drag/Right-Drag (Pan), Scroll (Zoom), WASD (movimiento fly-through), Q/E (Up/Down), F (Toggle Orbit/Fly), doble clic (re-centrar), Cmd+Scroll (ajuste FoV). **Views:** R (Reset Camera), T (Auto-Rotation), P (Camera Playback), B (ciclo de fondo), 0–9 (saltar a cámara de entrenamiento 1=10%/5=50%/0=última), Flecha Izquierda/Derecha (Prev/Next Cam). **Capture:** S (captura al escritorio), V (Turntable Video), C (Copy Camera Info). **Editor:** Tab (modo edición), Click/Drag (Paint-Select), Option+Click (Deselect), X / Delete (eliminar selección), Cmd-Z (deshacer último borrado), [/] (tamaño de pincel menor/mayor), Esc (limpiar selección). **Training:** Start, Pause/Resume, Cancel, Continue +5K/+10K/+20K vía los atajos de menú en M9–M14.

Qué es: Una visión general estática simple de todos los atajos de teclado — Navigation, Views, Capture, Editor, Training. El contenido está codificado en duro, sin carga de Markdown.

CUÁNDO ABRIRLA Cuando buscas la forma más rápida de hacer algo en la vista. WASD fly-through, R para reset de cámara, B para ciclo de fondo — están todos aquí.

W5 ScrollView (área de contenido)

DÓNDE

Help → Keyboard Shortcuts (⌘/).

TÉCNICO

Un área de scroll simple con una lista vertical dentro. Padding 20 alrededor, sin árbol de navegación en barra lateral (la lista es lo bastante corta). El contenido se agrupa en cinco secciones (Navigation, Views, Capture, Editor, Training). Una fila por combinación de teclas con texto traducible en ambas columnas. La columna izquierda (código de tecla) tiene ancho fijo de 180 pt, así las descripciones a la derecha se mantienen alineadas verticalmente. Sin interacción más allá del scroll — hacer clic en una fila no dispara nada; los atajos son modificadores de teclado reales en el menú y la vista.

EN POCAS PALABRAS

Tabla con todas las teclas. Cheat sheet estática para consulta rápida.

W6 VStack (secciones de atajos)

DÓNDE

Dentro del ScrollView.

TÉCNICO

Secciones apiladas alineadas a la izquierda con espaciado de 16 pt. Dentro de las cinco secciones, cada una contiene una cabecera + secuencia de filas. Las cabeceras usan un estilo secondary subheadline — deliberadamente no un formato de título, porque las secciones no necesitan ser navegables. El contenido es deliberadamente plano (sin disclosure, sin búsqueda, sin filtro) para que el componente corra sin cambios en cualquier versión de macOS y el archivo se mantenga legible.

EN POCAS PALABRAS

La agrupación de teclas por función (Navigation, Views, Editor, etc.).

Manage Storage (W7–W12)

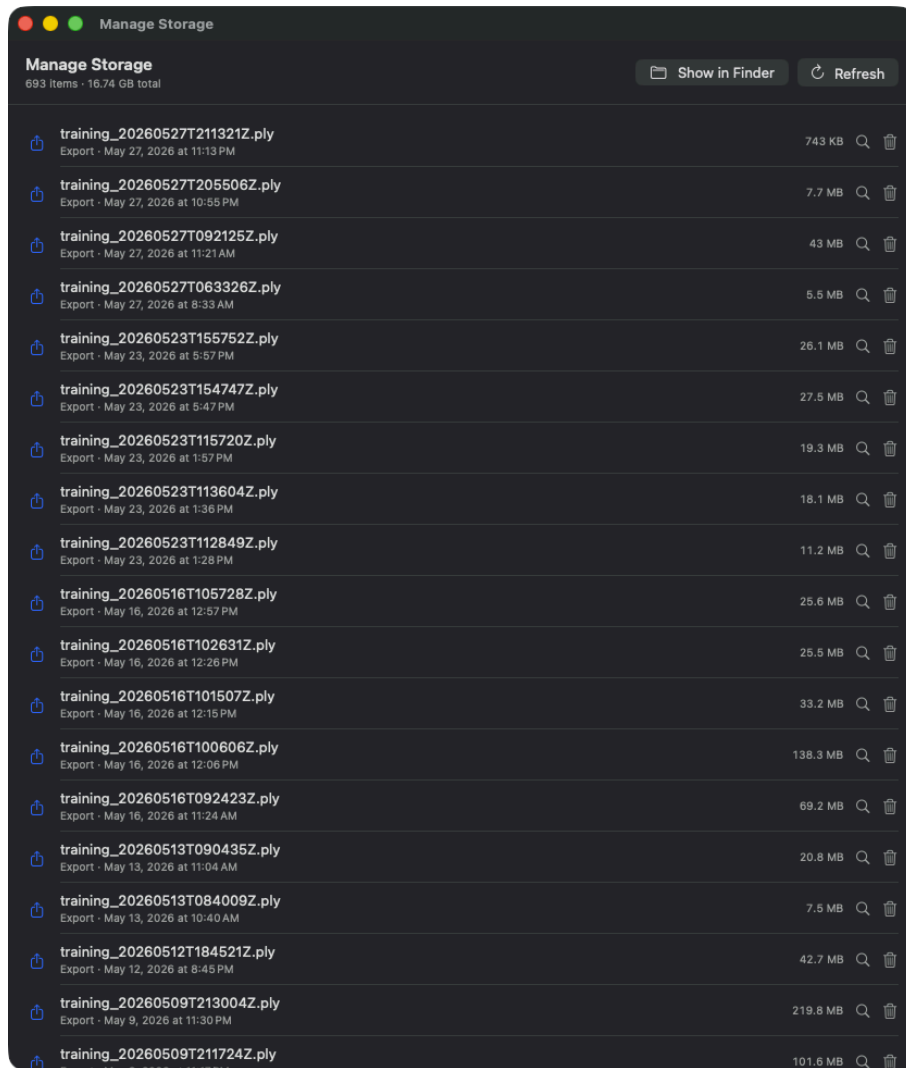


Figura 19: Ventana Manage Storage — la cabecera muestra «693 ítems · 16,74 GB total», tabla con archivos PLY de exportación ordenados por fecha, cada uno con insignia de formato + nombre + tamaño + fecha

LO QUE SE VE EN LA IMAGEN Vista de tabla de todos los archivos gestionados por RadianceKit. La cabecera cuenta 693 ítems, 16,74 GB de tamaño total. Toolbar arriba: «Show in Finder» + «Refresh». Cada fila: icono PLY, nombre de archivo (p. ej., `training_20260527T211321Z.ply`), fecha de exportación, tamaño (varía de 7 KB a 218 MB), icono de lupa (Reveal) e icono de papelera (Move to Trash). Los archivos están ordenados por fecha, los más nuevos arriba. En esta grabación demo, las exportaciones PLY dominan porque se trabajó mucho con `--benchmark`.

Qué es: Una visión general del uso de disco de todo lo que RadianceKit almacena en `~/Documents/RadianceKit/` — Logs, Exports, Scenes, paquetes de Capture (de la companion iOS), Imports (copias de staging de las imágenes de entrada). Un tamaño en bytes por entrada y dos botones: «Show in Finder» y «Move to Trash». Esto NO es una limpieza automática — la app no elimina nada por sí misma; decides por entrada.

CUÁNDO ABRIRLA Cuando el disco se llena. Los logs en particular se acumulan (un JSONL por intento de entrenamiento, más el `_qualityMetrics.json`); las exportaciones también, claro (PLY 100 % datos crudos, una por exportación). También útil tras un crash, cuando el directorio de staging de imports aún tiene copias viejas de las imágenes de entrada (véase «Incidente de presión de disco» en `dev_v549f-needle-reduction.md`).

W7 Botón «Show in Finder»

DÓNDE

Arriba a la derecha de la cabecera en la ventana del navegador de almacenamiento.

TÉCNICO

Abre todo el directorio de RadianceKit (`~/Documents/RadianceKit/`) en el Finder, así puedes ver directamente la estructura de carpetas y también manipularla con el propio Finder. La acción abre una nueva ventana del Finder y no cambia al contenedor sandbox de la app — `~/Documents/RadianceKit/` es el dominio normal de Documentos accesible a las apps, no una ruta de contenedor sandbox.

EN POCAS PALABRAS

Abre el directorio en el Finder para que puedas trabajar con los archivos por tu cuenta.

W8 Botón «Refresh»

DÓNDE

Cabecera, junto al botón Finder.

TÉCNICO

Dispara un escaneo en segundo plano que corre como tarea asíncrona iniciada por el usuario para que escanear árboles de directorios grandes no bloquee la UI. La caminata real recorre cada subcarpeta conocida (Logs, Exports, Scenes, Captures, Imports) y produce una entrada de almacenamiento por hijo directo. Por entrada se determina el tamaño recursivo — preferentemente el uso real de disco (incluido el sharing de hardlinks de APFS) con fallback al tamaño de archivo lógico.

EN POCAS PALABRAS

Relee la lista por si has eliminado o añadido algo en el Finder entretanto.

W9 List (entradas de almacenamiento) **DÓNDE**

Contenido principal bajo la cabecera.

 **TÉCNICO**

Lista con este layout por fila: icono SF Symbols específico de categoría (documento para Logs, flecha de subida para Exports, cubo para Scenes, bandeja para Imports), nombre + subtítulo (etiqueta Kind + fecha de modificación formateada), contador de bytes a la derecha (alineado a la derecha, monoespaciado), botón Reveal (icono de lupa), botón papelera (papelera). Orden: primario por Kind (Scenes primero, luego Exports, Logs, Captures, Imports, Other), secundario por fecha de modificación descendente (lo más nuevo arriba). Si el escaneo sigue en curso, el área muestra en su lugar un indicador de progreso «Scanning...». Si no se encontró nada, una visualización empty-state con un icono de bandeja.

 **EN POCAS PALABRAS**

Lista de todos tus datos de RadianceKit, ordenados por tipo y recencia. Por entrada ves el tamaño y puedes eliminar directamente.

W10 Botón de fila «Reveal in Finder» **DÓNDE**

Por fila, icono de lupa a la derecha.

 **TÉCNICO**

Abre el Finder y selecciona el ítem específico (archivo o carpeta). Diferencia con W7: W7 abre el directorio raíz; W10 resalta exactamente esta entrada. Flujo práctico: identificar una entrada grande, hacer clic en la lupa, después copiarla, por ejemplo, a un volumen externo.

 **EN POCAS PALABRAS**

Salta a esta entrada directamente en el Finder para que la encuentres rápido.

W11 Botón de fila «Move to Trash»

DÓNDE

Por fila, icono de papelera a la derecha junto a la lupa.

TÉCNICO

Dispara el diálogo de confirmación (W12). Solo tras la confirmación corre la operación estándar de macOS «move to trash» (es decir, reversible, sin borrado directo). Tras tirar a la papelera con éxito, la entrada se elimina de la lista y el contador total de bytes se actualiza. En caso de errores, se muestra un diálogo modal de error.

EN POCAS PALABRAS

Mueve la entrada a la papelera. Un diálogo pregunta primero.

W12 ConfirmationDialog (confirmación de borrado)

DÓNDE

Disparado por W11, mostrado como sheet de macOS.

TÉCNICO

Diálogo estándar de confirmación con un título dinámico «Delete <name>?» y una línea de mensaje que indica explícitamente que la entrada va a la papelera y puede restaurarse desde allí (hasta que se vacíe la papelera). Dos botones: «Move to Trash» como acción destructiva (mostrada en rojo) y «Cancel» con vínculo automático a Esc. El diálogo es no modal en el sentido de que solo bloquea esta ventana, no toda la app — es el estándar de macOS para borrados reversibles.

EN POCAS PALABRAS

Aviso de seguridad antes de borrar. «Move to Trash» es reversible — mientras no se vacíe la papelera.

Pareto Dashboard (W13–W22)

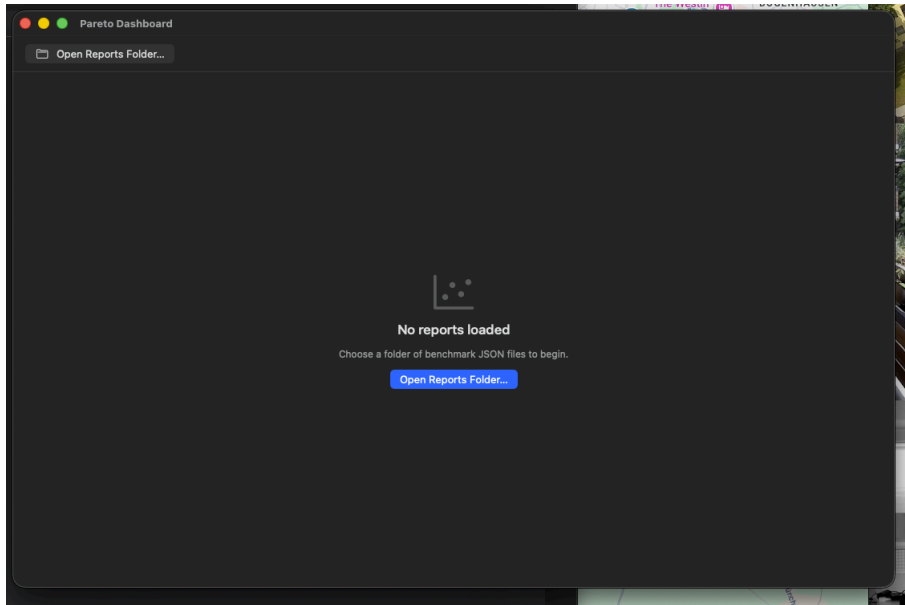


Figura 20: Pareto Dashboard — estado vacío antes de la importación de reportes

Estado vacío (tras la primera apertura) — estado vacío con call-to-action «Open Reports Folder...». Los puntos de datos aparecen en cuanto se cargan los reportes de entrenamiento, véase la siguiente captura.

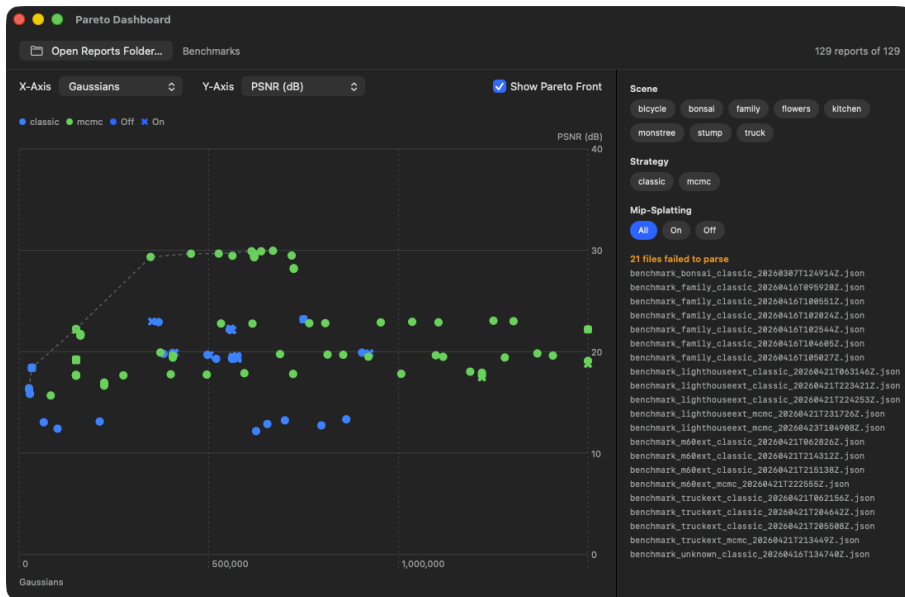


Figura 21: Pareto Dashboard con 129 reportes de benchmark cargados — Gaussianos vs PSNR con frente de Pareto, filtros Scene/Strategy/Mip

LO QUE SE VE EN LA IMAGEN La barra de herramientas de la cabecera muestra «129 reports of 129» (todos los reportes de la carpeta seleccionada parseados con éxito — 21 archivos adicionales no pudieron parsearse por formato antiguo, véase la lista de avisos a la derecha). Ejes: selector de eje X en Gaussians , selector de eje Y en PSNR (dB) . Scatter plot: puntos verdes = estrategia Classic, puntos azules = MCMC.

La línea discontinua del frente de Pareto corre a lo largo de los mejores valores de PSNR alcanzados y se aplana hacia PSNR≈30 dB a partir de unos 500K Gaussianos. Chips de filtro a la derecha: 7 escenas (bicycle, bonsai, family, flowers, kitchen, stump, truck), 2 estrategias (classic, mcmc), 3 opciones de Mip-Splatting (All, On, Off). En este momento todos los filtros están abiertos, de ahí el denso cluster de puntos.

Qué es: Una herramienta de comparación multi-run. En el pasado entrenaste varias escenas, o la misma escena con distintos ajustes preestablecidos — cada uno de esos runs produce (si pasaste `--benchmark` o lo llamaste mediante la función Benchmark) un archivo de reporte JSON que contiene, entre otras cosas, PSNR final, SSIM, LPIPS, número de Gaussianos y tiempo wallclock. El Dashboard lee toda una carpeta de esos reportes a la vez y los representa como scatter 2D con ejes seleccionables. Además, se dibuja el frente de Pareto (el conjunto de puntos no dominados) como línea discontinua.

CUÁNDO ABRIRLA Después de haber producido al menos tres o cuatro reportes de entrenamiento. Con menos puntos, la línea de frente no es significativa. Caso de uso típico: intentaste reconstruir una escena exterior y ejecutaste sucesivamente P3 Balanced (Classic), P4 Quality (Classic), P7 MCMC Quality y P9 Outdoor (tuned) — ahora quieres saber qué configuración entrega el mejor PSNR por segundo de tiempo de entrenamiento, o cuál requiere menos Gaussianos para un PSNR dado.

CÓMO INTERPRETARLA Ambos ejes son libremente seleccionables (eje X: `psnr`, `ssim`, `lpips`, ...; eje Y lo mismo). La lógica de frente de Pareto en `ParetoFront2D.indices` sabe para cada métrica si «menor = mejor» (p. ej., LPIPS, Loss, Time) o «mayor = mejor» (PSNR, SSIM) — así que, según la elección de eje, la línea va de abajo-izquierda a arriba-derecha o de arriba-izquierda a abajo-derecha, siempre a lo largo de la mejor combinación alcanzada. Un punto es Pareto-óptimo si NINGÚN otro punto es al menos tan bueno en AMBAS dimensiones (es decir, ningún otro punto lo domina). Los puntos Pareto-óptimos están sobre la línea, los demás a la derecha/encima (según la orientación del eje). Los puntos SOBRE la línea son los candidatos reales para «mejor ajuste preestablecido»; los puntos LEJOS de la línea son tiempo de entrenamiento desperdiciado.

CHIPS DE FILTRO Puedes restringir la selección a una escena concreta (si solo quieres comparar runs de exteriores, p. ej.), a una estrategia concreta (Classic o MCMC) o a Mip-Splatting on/off (relevante tras la Fase Q1.5, donde Mip permanece como flag avanzado opt-in).

FLUJO DE EJEMPLO Tienes tres reportes para la escena «truck» en `~/Documents/RadianceKit/Reports/`: Run A (P4 Quality, 40K iter, 524K Gs, 105 s, PSNR 23,4), Run B (P7 MCMC, 200K iter, 150K Gs, 693 s, PSNR 24,6), Run C (P9 Outdoor, 100K iter, 1,25M Gs, 312 s, PSNR 25,8). Pon el eje X en `trainingTime`, el eje Y en `PSNR`. El Run B queda arriba a la derecha, el Run C aún más arriba a la derecha, el Run A abajo a la izquierda. El frente de Pareto conecta A y C — ambos no dominados. El Run B está «perdido» (C es mejor en Time Y PSNR). Conclusión: para «truck» el default MCMC no compensa; o rápido+ok (A) o largo+muy bueno (C). Guarda la configuración de C como ajuste preestablecido propio (Inspector → I1 Save Preset).

Siguiente acción: Guardar la mejor configuración como ajuste preestablecido. Concretamente: mira los puntos de Pareto (al hacer hover se muestra PSNR/SSIM/LPIPS/Gs/Time en el tooltip), decide cuál te encaja mejor en el trade-off tiempo-vs-calidad, abre

el reporte correspondiente (el nombre de archivo contiene la marca de tiempo del run), copia su configuración de entrenamiento a un nuevo run, o guárdala tras la siguiente sesión de entrenamiento como ajuste preestablecido mediante el Inspector.

W13 Botón «Open Reports Folder...»

DÓNDE

Toolbar, arriba a la izquierda.

TÉCNICO

Abre un diálogo selector de carpeta con el prompt «Select a folder containing benchmark .json reports». Tras la confirmación, corre una tarea en segundo plano que parsea secuencialmente todos los archivos `.json` de la carpeta. Los reportes defectuosos (JSON roto, schema incorrecto) se recopilan y se muestran al pie de la barra lateral como «N file failed to parse» — sin crash. Si ocurre un segundo clic mientras una primera carga sigue en curso, la tarea anterior se cancela para que dos resultados no escriban a la vez en el estado.

También vía CLI: `--dashboard-dir /ruta/a/reports` carga la carpeta directamente al arrancar la app.

EN POCAS PALABRAS

Selecciona la carpeta donde están tus reportes de benchmark. La ruta predeterminada es `~/Documents/RadianceKit/Reports/`. Después carga todos los JSONs a la vez.

W14 Selector «X-Axis»

DÓNDE

Encima del gráfico, a la izquierda.

TÉCNICO

Selector tipo menú con todos los ejes de métrica disponibles del módulo de dashboard (PSNR, SSIM, LPIPS, número de Gaussianos, tiempo de entrenamiento, etc.). El predeterminado es número de Gaussianos. Al cambiar, el punto sobre el que estaba el hover se restablece, porque una posición resaltada previamente en el viejo sistema de coordenadas ya no tiene sentido tras el cambio de eje. El selector está constreñido a ancho de contenido para que no abarque todo el ancho.

EN POCAS PALABRAS

Qué métrica debe estar en el eje horizontal. Normalmente «tiempo de entrenamiento» o «número de Gaussianos» porque son los «costes» que quieres comparar.

W15 Selector «Y-Axis» **DÓNDE**

Encima del gráfico, junto a X-Axis.

 **TÉCNICO**

Idéntico a W14, salvo que el predeterminado es PSNR. La elección de eje se almacena de forma independiente, así que el usuario también puede elegir combinaciones sin sentido (X=PSNR, Y=PSNR — pondría todos los puntos en una diagonal). Esas combinaciones no se atrapan, sin embargo; una decisión deliberada, porque una comparación «SSIM vs PSNR» es bastante interesante para ver lo consistentes que son las métricas.

 **EN POCAS PALABRAS**

Qué hay en el eje vertical. Normalmente «PSNR» o «SSIM» como medida de calidad.

W16 Conmutador «Show Pareto Front» **DÓNDE**

A la derecha de los selectores de ejes.

 **TÉCNICO**

Conmutador estándar de macOS. Si está activo, se dibuja en el gráfico de Pareto una línea con el frente de Pareto 2D calculado, además de la nube de puntos. Estilo: discontinua (patrón de guiones 4–4), gris semitransparente, ancho de línea 1,5 pt. El cálculo de Pareto corre en el hilo principal — con el número típico de reportes ($\leq \sim 50$) es rápido sin problemas. Si el conmutador está apagado, la línea se omite y solo quedan los puntos.

 **EN POCAS PALABRAS**

Muestra la línea que pasa por los puntos «mejores hasta el momento». Si la línea estorba (p. ej., porque solo quieres comparar trades individuales), apágala.

W17 Chips de filtro «Scene»

 **DÓNDE**

Barra lateral derecha en la ventana del Dashboard.

 **TÉCNICO**

Chips de filtro para cada escena que aparece en los reportes cargados. Layout flow personalizado que envuelve automáticamente chips en varias líneas en cuanto se agota el ancho. Los chips activos reciben el fondo de acento, los inactivos un fondo material estándar neutro. La multiselección es posible (semántica de conjunto); si no hay chip seleccionado, todas las escenas cuentan como «pasadas» — es decir, la lógica de conjunto es «selección vacía = todo», no «selección vacía = nada».

EN POCAS PALABRAS

Hacer clic en un nombre de escena filtra los puntos solo a esa escena. Multiselección posible. Vacío = todas las escenas.

W18 Chips de filtro «Strategy»

 **DÓNDE**

Bajo el filtro Scene en la barra lateral.

 **TÉCNICO**

Exactamente como W17, pero para estrategias de entrenamiento — típicamente los dos valores «classic» y «mcmc», derivados del campo strategy de los JSONs de reportes de benchmark. Útil si tienes reportes mezclados de ambas estrategias y solo quieres ver un tipo (p. ej., «solo mostrar runs de MCMC porque ya he excluido Classic»).

EN POCAS PALABRAS

Filtrar por Classic o MCMC. Por defecto ambos están activos.

W19 Chips de filtro «Mip-Splatting» **DÓNDE**

Bajo el filtro Strategy en la barra lateral.

 **TÉCNICO**

Filtro de tres valores (en lugar de un conjunto como W17/W18): «All» / «On» / «Off». Trasfondo: Mip-Splatting se evaluó en la Fase Q1.5 como mejora multi-escala experimental y el veredicto final fue «no hay una victoria clara a través de todo; mantener como flag opt-in». Cuando haces comparaciones Mip on/off a menudo quieres separar muy nítidamente. De ahí el filtro ternario dedicado con los estados «deja pasar todo», «solo Mip on», «solo Mip off». La sección de la barra lateral solo se muestra si en el conjunto de datos hay al menos un reporte Mip Y al menos un reporte no-Mip (de lo contrario, filtrar no tendría sentido).

 **EN POCAS PALABRAS**

Si quieres comparar Mip-Splatting on/off, aquí hay un filtro de tres vías. En otro caso, ignorar.

W20 ChipButton (conmutador de filtro, all/on/off) **DÓNDE**

Componente auxiliar, usado en W17/W18/W19.

 **TÉCNICO**

Wrapper de botón minimalista. Contenido: texto de etiqueta con tamaño de fuente caption y padding 10 horizontal / 5 vertical. Fondo condicional: si está activo → color de acento de la app con texto blanco; en otro caso, fondo material estándar neutro con texto negro. La forma es de cápsula (tipo pill). Estilo plain-button para que el material de la cápsula no quede solapado por un borde del sistema.

 **EN POCAS PALABRAS**

Los propios botones de filtro redondeados. Visualmente como una etiqueta de iOS.

W21 Chart (scatter de Pareto) **DÓNDE**

Área central del Dashboard.

 **TÉCNICO**

Diagrama Swift Charts con dos capas: 1. un punto por reporte — posición a partir de las métricas X e Y elegidas, color por estrategia, símbolo por estado Mip. Tamaño de símbolo normal 80, resaltado 200 (si el ID coincide con el reporte sobre el que está actualmente el hover). 2. una línea para el frente de Pareto, solo si el conmutador está activo.

Overlay del chart: un rectángulo transparente registra el movimiento del ratón; por fotograma se determina la posición del punto más cercano en distancia euclídea en el plot frame y el reporte hovered se actualiza si la distancia es menor de 24 px (en otro caso se resetea). Así obtienes el tooltip sin hacer clic — basta con pasar por encima.

 **EN POCAS PALABRAS**

El scatter plot real. Cada punto es un run de entrenamiento. Pasa por encima para el tooltip de detalle.

W22 Tooltip (detalle por hover) **DÓNDE**

Bajo el gráfico, mostrado al hacer hover.

 **TÉCNICO**

Stack horizontal: nombre de escena (headline), tag de estrategia (caption), separador, después métricas PSNR/SSIM/LPIPS/Gs/Time cada una en un pequeño grupo vertical (etiqueta + valor monoespaciado). Si Mip estaba activado, adicionalmente una etiqueta cápsula «Mip» en color de acento. Fondo semitransparente con blur, rectángulo redondeado con radio de 8 pt. Solo se muestra cuando el ratón está realmente sobre un punto. Desaparece automáticamente al salir.

 **EN POCAS PALABRAS**

La tarjeta de detalle al pie cuando pasas con el ratón por encima de un punto. Muestra a la vez todas las métricas de calidad y la configuración del run.

Holdout Analysis (W23–W29)

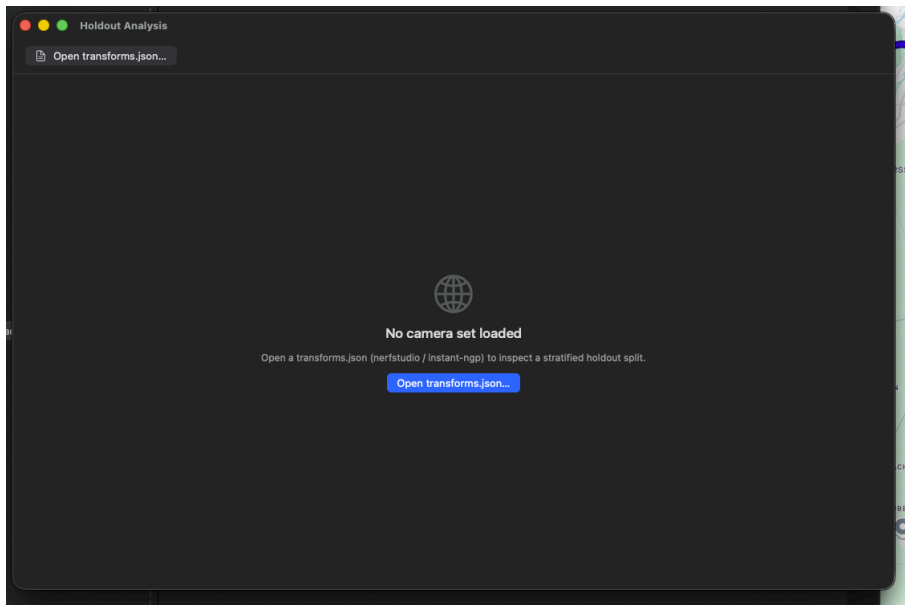


Figura 22: Holdout Analysis — estado vacío antes de cargar un `transforms.json`

Estado vacío con empty state y call-to-action «Open transforms.json...». Acepta el formato de NeRF Studio e Instant-NGP.

Estado vacío (tras la primera apertura) — los marcadores de cámara aparecen en cuanto se carga un `transforms.json`, véase la siguiente captura.

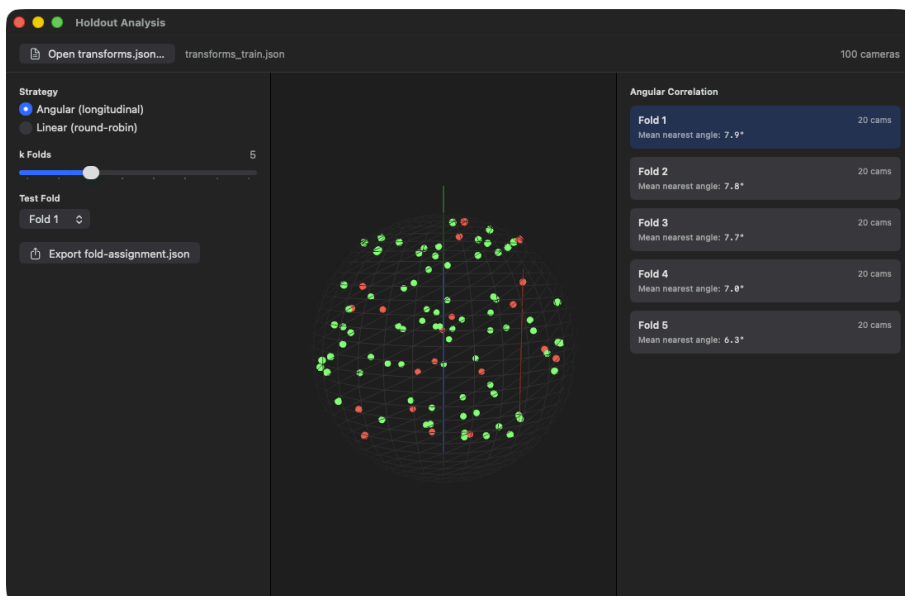


Figura 23: Globo de Holdout con 100 cámaras NeRF Blender mic, 5 folds de 20 cámaras cada uno, estrategia angular activa

LO QUE SE VE EN LA IMAGEN La cabecera muestra el archivo cargado (`transforms_train.json`) y el número de cámaras («100 cameras»). Barra lateral izquierda: selector de estrategia con dos opciones — Angular (longitudinal) activo (alineamos los folds a lo largo de sectores longitudinales/latitudinales en la esfera, así cada fold de test es

geométricamente denso) vs Linear

(round-robin) (basado en orden, cada k-ésimo fotograma como conjunto de test). El deslizador de k-folds está en 5, el selector de fold de test en Fold 1. El botón Export produce un `fold-assignment.json` para Nerfstudio/Instant-NGP. Panel central: proyección de globo 3D de las 100 cámaras — puntos verdes = Train, puntos rojos = fold de test actual (Fold 1 con 20 cámaras). Barra lateral derecha (Angular Correlation): por fold 20 cámaras + Mean Nearest Angle (Fold 1: 7,9°, Fold 2: 7,8°, Fold 3: 7,7°, Fold 4: 7,0°, Fold 5: 6,3°) — un valor más pequeño significa que las cámaras dentro de ese fold están juntas, así que el split de Holdout es espacialmente coherente.

Qué es: Un visualizador 3D para tu disposición de cámaras con lógica de cross-validation. Cargas un `transforms.json` (el formato estándar de Nerfstudio / Instant-NGP para poses de cámara), la app lee todas las cámaras, proyecta sus direcciones de visión sobre una esfera unidad y las muestra como pequeños marcadores esféricos sobre un globo virtual. Después divide las cámaras en `k` folds (según la estrategia seleccionada: angular o lineal), marca la porción de entrenamiento en verde y la de test en rojo (Holdout), y calcula por fold una puntuación de Angular Correlation que te dice lo lejos que está el fold de test del fold de entrenamiento en el espacio de ángulos de visión.

CUÁNDO ABRIRLA Cuando quieres hacer evaluación Holdout — es decir, ¿lo bien que tu modelo generaliza a puntos de vista no vistos? El default en el entrenamiento es «cada 8ª vista como Holdout» (convención Mip-NeRF360), pero es un split muy lineal. Si tus imágenes se agrupan en el tiempo, por ejemplo (un lado del objeto primero, después el otro), entonces «cada 8ª» no es representativo — una posición aleatoria de secuencia acaba en test, pero todos sus vecinos están en entrenamiento, lo que es demasiado fácil. Con «angular» estratificas a lo largo del espacio de ángulos de visión: cada fold contiene cámaras de todas las zonas de la órbita, así que el test sondea de verdad los huecos de generalización.

CÓMO INTERPRETARLA Angular vs Linear: - Angular (default): divide las cámaras por ángulo longitudinal (coordenada ϕ alrededor del eje Y) en `k` sectores iguales. El Fold 0 contiene cámaras con $\phi \in [0^\circ, 360/k^\circ)$, el Fold 1 las siguientes, y así sucesivamente. Ventaja: cada fold cubre una porción de la órbita; el fold de test es espacialmente compacto pero ampliamente distribuido por el dataset global. Bueno para grabaciones orbitales clásicas. - Linear (round-robin): índice de fold = $(\text{image_index} \bmod k)$. Ese es el simple split «cada k-ésimo». Funciona si el orden de imagen NO tiene sesgo espacial (p. ej., tomas de dron ordenadas aleatoriamente). Funciona mal si las imágenes se agrupan en el tiempo.

En el globo 3D ves de inmediato: puntos verdes (entrenamiento) y puntos rojos (test). Si los puntos rojos se agrupan todos en una esquina, el Holdout es malo (sin buen test de generalización). Si están repartidos entre los verdes, es bueno. La puntuación de Angular Correlation por fold (barra lateral derecha, en grados) dice además: valor más pequeño = el test está cerca del entrenamiento (cada cámara de test tiene una cámara de entrenamiento cercana, test fácil); valor más grande = el test está lejos del entrenamiento (generalización más difícil).

FLUJO DE EJEMPLO Capturaste tu escena de truck con 251 imágenes, exporta mediante la entrada de menú M33 (Export SfM transforms.json) un archivo Nerfstudio. Abre la

ventana Holdout (⇧⌘H), carga el JSON con «Open transforms.json...», mira el globo. $k=5$ (default) te da 5 folds. Haz clic en «Fold 3» — comprueba si los marcadores rojos están razonablemente repartidos. Si sí: «Export fold-assignment.json», pon el archivo exportado en la carpeta de reportes, y en el siguiente run de entrenamiento con `--benchmark` (o los ajustes correspondientes del Inspector) se usa exactamente esa asignación de fold como Holdout de test — en lugar del default «cada 8ª».

W23 Botón «Open transforms.json...»

DÓNDE

Toolbar, arriba a la izquierda.

TÉCNICO

Abre un diálogo selector de archivo restringido a archivos JSON. Tras la confirmación, el módulo Holdout carga el archivo. El loader parsea tanto el formato Nerfstudio (intrínsecos de cámara más lista de frames con ruta de imagen y matriz transform) como el formato Instant-NGP (misma estructura). Para cada frame se extrae la dirección de visión de la matriz transform (eje z de la base local de la cámara) y se almacena. Si el parseo falla, se muestra un mensaje de error en el área de estado.

También vía CLI: `--holdout-file /ruta/a/transforms.json` abre la ventana directamente con el archivo cargado.

EN POCAS PALABRAS

Carga tu JSON de poses de cámara. Lo estándar son las exportaciones de Nerfstudio e Instant-NGP. RadianceKit mismo puede exportar transforms.json vía Menú → Export → SfM.

W24 Selector «Strategy» (angular/linear)

DÓNDE

Barra lateral izquierda, arriba.

TÉCNICO

Selector tipo radio con dos opciones: Angular y Linear. El cambio de estrategia dispara automáticamente un nuevo cálculo de los folds. Las direcciones de visión son una lista de vectores unidad 3D en la esfera; la estrategia angular las proyecta sobre el ángulo longitudinal ϕ y ordena, la estrategia lineal simplemente hace un split modulo sobre el índice de frame.

EN POCAS PALABRAS

Angular para tomas orbitales bien distribuidas (default, seguro), Linear solo si tus imágenes no se agrupan espacialmente.

W25 Deslizador «k Folds» **DÓNDE**

Barra lateral izquierda, en el medio.

 **TÉCNICO**

Deslizador de 3 a 10, paso 1. Al cambiar, el cálculo de folds se reinicia automáticamente para que la lista de folds, los índices de entrenamiento/test y la puntuación por fold se recalculen de inmediato. El valor seleccionado se muestra como texto monospaced-digit junto a la etiqueta de la derecha.

Regla práctica: $k=5$ es el estándar (te da un 20 % de test por fold, lo común para cross-validation). $k=10$ si tienes muchos datos y necesitas más folds para significancia estadística. $k=3$ si tienes pocos datos.

 **EN POCAS PALABRAS**

Cuántos folds tiene el split. 5 es el default y encaja casi siempre.

W26 Selector «Test Fold» **DÓNDE**

Barra lateral izquierda, bajo el deslizador k.

 **TÉCNICO**

Selector tipo menú. Las opciones son dinámicamente $0..<k$, etiquetas «Fold 1» a «Fold N» (es decir, 1-indexado en la UI, 0-indexado internamente). Si el índice seleccionado previamente es $\geq k$ (p. ej., porque has bajado k de 10 a 5), se restablece automáticamente a 0. El fold de test seleccionado se muestra en rojo en el globo, todos los demás en verde.

 **EN POCAS PALABRAS**

Qué fold es el actual de test. Puedes clicar entre ellos y ver cómo se ve cada uno en el globo.

W27 Botón «Export fold-assignment.json» DÓNDE

Barra lateral izquierda, abajo.

 TÉCNICO

Abre un diálogo de guardar con nombre predeterminado `fold-assignment.json`. Tras la confirmación, el módulo Holdout codifica el split actual en un schema JSON (asignación de fold por frame más bloque meta de estrategia). Ese archivo se puede pasar luego al siguiente run de entrenamiento con `--benchmark`, así se usa el mismo Holdout para la evaluación final de métricas. Los errores de escritura se muestran como texto de error; el éxito en texto verde como «Saved to (filename)».

 EN POCAS PALABRAS

Guarda el split actual train/test como JSON. Después puedes pasar ese archivo directamente al entrenamiento para que se use de nuevo el mismo conjunto de test.

W28 SCNView (globo de cámaras 3D) DÓNDE

Panel central en la ventana Holdout.

 TÉCNICO

Vista de globo SceneKit. La escena consta de: una esfera wireframe (radio 1,0, 36 segmentos, gris oscuro), tres stubs de eje coloreados (rojo/verde/azul para X/Y/Z, cada uno de 1,2 de largo) y, por cámara, una pequeña esfera marcadora (radio 0,03) en la posición correspondiente de la dirección de visión sobre la esfera unidad (un poco por fuera para que no desaparezca DENTRO de la esfera wireframe). Los marcadores NO se reconstruyen en cada cambio de fold — la reconstrucción solo es necesaria cuando cambia la lista de frames (es decir, se carga un nuevo JSON). En su lugar, por actualización corre una actualización in-place de los colores del material: rojo para los índices de test, verde para los de entrenamiento, gris claro si ninguno. Así los ticks del deslizador siguen siendo rápidos incluso con $N > 1000$ cámaras.

El control de cámara está activado — puedes rotar, hacer zoom y desplazar el globo con el ratón. La iluminación se asegura de que los marcadores no se vean planos. El fondo es gris oscuro.

 EN POCAS PALABRAS

El globo 3D con las posiciones de cámara. Verde = entrenamiento, rojo = test, gris claro = sin asignar (no ocurre, todas las cámaras pertenecen a algún sitio). Con el ratón puedes rotar y hacer zoom al globo.

W29 FoldCard (tocar para seleccionar fold)**DÓNDE**

Barra lateral derecha, sección «Angular Correlation».

TÉCNICO

Una vista de tarjeta por fold — rectángulo redondeado con radio de 6 pt, padding 10, layout vertical con dos filas (arriba «Fold N» + número de cámaras, abajo «Mean nearest angle:»

1. valor en grados). Color de fondo condicional: fold activo =

color de acento semitransparente, inactivo = material estándar neutro. Tocar selecciona el fold y el globo recolora en vivo.

La puntuación «Mean nearest angle» es el ángulo más pequeño medio por cámara de test a la cámara de entrenamiento más cercana (calculado internamente en radianes, mostrado en grados en la UI).

EN POCAS PALABRAS

Una pequeña tarjeta por fold a la derecha con el número de cámaras y la distancia media a la cámara de entrenamiento más cercana. Hacer clic la selecciona como test.

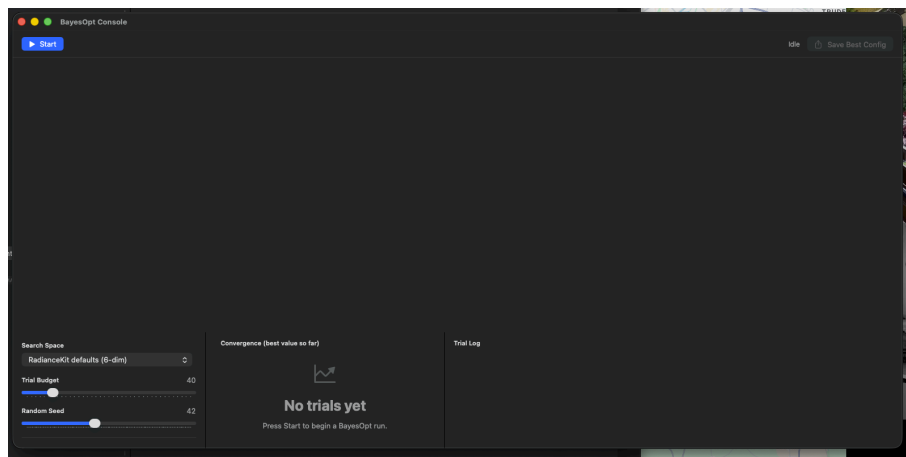
BayesOpt Console (W30–W39)

Figura 24: Consola BayesOpt — estado vacío antes del inicio del trial

Estado vacío con selector de espacio de búsqueda (RadianceKit defaults (6-dim)), deslizador de trial budget (default 40), random seed (42) y tres paneles vacíos para gráfico de convergencia, log de trials y lista de parámetros del espacio de búsqueda.

Estado vacío (tras la primera apertura) — el gráfico de convergencia y la tabla de trials se llenan en cuanto se inicia un run, véase la siguiente captura.



Figura 25: Consola BayesOpt tras 40 trials — el gráfico de convergencia sube con fuerza hasta el trial 15, mejor valor 0,9943, log de trials con tags init/bo/restart

LO QUE SE VE EN LA IMAGEN Estado arriba a la derecha «Finished — best 0.9943 after 40 trials». Barra lateral izquierda: selector de espacio de búsqueda en RadianceKit defaults (6-dim), trial budget 40, random seed 42. La lista de parámetros muestra los seis hiperparámetros a tunear con sus rangos de valor: mipSmoothing3DScale [0.05, 0.5], mipFilter2DVariance [0.1, 0.6], densifyGradThreshold [5e-07, 5e-06], ssimWeight [0.05, 0.5], mcmcNoiseScale [1e-05, 0.0001], mcmcRelocationInterval [50, 200]. Centro: gráfico de convergencia (X = índice de trial 1-40, Y = valor objetivo 0-1) — puntos grises = muestras iniciales (LHS), puntos azules = adquisición BayesOpt, puntos naranjas = trials restart (#22 y #31). La línea de mejor valor sube con fuerza hasta el trial ~7, después solo mejora marginal hasta el trial 15, a partir de ahí una meseta plana en 0,99+. Barra lateral derecha: log de trials #1-#34 con score + tag (init/bo/restart). El botón Save Best Config arriba a la derecha escribe `bayesopt-best.json`.

Qué es: Una consola de optimización bayesiana para búsqueda de hiperparámetros. La optimización bayesiana es un método automático que intenta encontrar el óptimo de una función desconocida con el menor número posible de experimentos — típicamente: «¿qué combinación de mcmcMaxGaussians, capMultiplier, ssimWeight y gradThreshold entrega el mejor PSNR para mi clase de escena?» En lugar de una rejilla de $6^4 = 1296$ trials, la optimización bayesiana prueba alrededor de 40-100 trials informados y se acerca así al óptimo.

Importante: La versión actualmente entregada en la app no ejecuta la optimización contra runs de entrenamiento reales (eso tardaría días) sino contra un objetivo demo sintético — un paisaje multi-modal con carácter de hill-climbing más ruido suave. Es deliberado: la ventana está pensada para mostrarte el comportamiento del optimizador (curva de convergencia, puntos de muestra, mejor hasta el momento) y dejarte entender las definiciones del espacio de búsqueda. Para runs reales de BayesOpt dirigidos por entrenamiento (como se hicieron en la Fase Q7 para los ajustes preestablecidos scene-class) se usa un flujo CLI offline separado; la ventana es la variante UI en vivo.

CUÁNDO ABRIRLA Tres casos de uso: 1. Quieres entender cómo funciona BayesOpt — entonces inicia un run demo y observa el gráfico de convergencia. 2. Estás planificando una nueva clase de escena (como «acuarios» o «muebles antiguos») para la que los

10 ajustes preestablecidos integrados no encajan a la perfección. Define mentalmente un espacio de búsqueda, pruébalo aquí con el «Bowl demo» o el preset «Densify», después exporta la mejor config como JSON y úsala como punto de partida para un run de entrenamiento real. 3. Quieres inspeccionar los espacios de búsqueda por defecto definidos en el paquete RKBayesOpt (subconjunto Mip, RadianceKit defaults) — se listan en el panel de parámetros de la barra lateral izquierda.

CÓMO INTERPRETARLA - **Gráfico de convergencia** (columna central): Y = mejor valor de función objetivo alcanzado hasta el momento. X = índice de trial. Sube inicialmente con fuerza (BayesOpt prueba las muestras iniciales aleatoriamente, algunas con suerte), luego se aplana cada vez más porque se agota la región cercana al óptimo. Si la línea se mantiene plana durante 20+ trials, puedes detener el run — trials adicionales no aportarán nada. Los puntos individuales en el gráfico son los valores de trial individuales (es decir, no «mejor hasta el momento»), coloreados por fase: gris = muestra inicial, azul = adquisición BayesOpt, naranja = restart. - **Tabla de trials** (columna derecha): #1, #2, #3, ... cada uno con valor y tag de fase. El mejor trial hasta el momento está marcado con una estrella amarilla. Desde la tabla puedes identificar el mejor trial e inspeccionar sus valores de parámetro después durante la exportación. - **Inspector de espacio de búsqueda** (barra lateral izquierda): muestra para el preset seleccionado todos los nombres de parámetro y sus rangos de búsqueda `[lo, hi]`. Si estás en el preset «RadianceKit defaults (6-dim)», ves, p. ej., «densifyGradThreshold [5e-7, 5e-6]» — es decir, log-uniforme entre esos dos valores.

FLUJO DE EJEMPLO Escoge el preset «RadianceKit defaults (6-dim)», trial budget 40, seed 42. Haz clic en «Start». Observa: los primeros 8 trials son grises (muestras iniciales, LHS Latin hypercube), los siguientes azules (adquiridos por BayesOpt). El gráfico de convergencia sube con fuerza hasta el trial ~15, después se aplana. En el trial ~30–40 el mejor valor se estabiliza. Haz clic en «Save Best Config» — se guarda un `bayesopt-best.json` con el nombre de preset, índice de trial, valor y los valores de parámetro decodificados. Después puedes copiar manualmente ese JSON a tu definición de ajuste preestablecido.

W30 Botón «Start» **DÓNDE**

Toolbar a la izquierda, en estado idle/finished.

 **TÉCNICO**

Restablece la lista de trials, cambia al estado running, genera un nuevo run ID (para detección de stale en clics múltiples de Start) y crea una nueva pause gate. Después arranca una tarea en segundo plano que ejecuta el optimizador como stream asíncrono. El tamaño de muestras iniciales es $\min(8, \text{budget} / 4 + 1)$ — así, típicamente 8 muestras Latin hypercube con $\text{budget} \geq 28$, menos con budget pequeño. Los updates de trial se reciben de forma incremental y se anexan a la lista. Protección stale-run: si entretanto un segundo clic de Start fija un nuevo run ID, los updates del run antiguo se descartan.

Estilo de acción primaria para el look prominente del botón.

 **EN POCAS PALABRAS**

Inicia un run de optimización nuevo con el espacio de búsqueda, budget y seed actuales.

W31 Botón «Pause» **DÓNDE**

Toolbar a la izquierda, en estado running.

 **TÉCNICO**

Activa la pause gate y cambia al estado paused. El efecto real: el runner espera en un bucle de polling de 50 ms antes de evaluar la siguiente función objetivo. Eso significa que un trial actualmente en curso se ejecuta hasta completarse (es sintético y solo tarda microsegundos), pero no se inicia un trial más. En cuanto Resume corre, continúa donde lo dejó.

 **EN POCAS PALABRAS**

Pausa el run. El cálculo actual aún corre hasta completarse, después pausa.

W32 Botón «Stop» **DÓNDE**

Toolbar a la izquierda, en estado running y paused.

 **TÉCNICO**

Cancela la tarea del runner, nulea la referencia, libera la pause gate (si todavía estaba en pause) y cambia al estado finished (si existen trials) o idle (si no). Los trials ya calculados permanecen visibles en la lista — Stop no los elimina. El rol destructivo del botón muestra el botón en rojo porque cancela el run.

 **EN POCAS PALABRAS**

Cancela el run de forma permanente. Los trials se mantienen visibles; aún puedes exportar la mejor config.

W33 Botón «Resume» **DÓNDE**

Toolbar a la izquierda, en estado paused.

 **TÉCNICO**

Libera la pause gate y vuelve al estado running. La tarea del runner ya está en marcha (está esperando en el bucle de polling); en cuanto el bucle nota que la pausa se ha levantado, continúa e inicia el siguiente trial.

 **EN POCAS PALABRAS**

Reanuda un run en pausa.

W34 Botón «Save Best Config» **DÓNDE**

Toolbar a la derecha, siempre visible (pero deshabilitado si no hay bestTrial).

 **TÉCNICO**

Abre un diálogo de guardar con nombre predeterminado `bayesopt-best.json`, restringido a JSON. Tras la confirmación se construye un diccionario payload: nombre de preset, índice de trial, valor (puntuación objetivo), parámetros (diccionario de nombres de parámetros decodificados → valores). La decodificación proyecta de vuelta las coordenadas del espacio de búsqueda normalizadas en $[0,1]^d$ al rango de valor original (con escalas log-uniforme/lineal/integer según corresponda). La salida JSON está pretty-printed y con claves ordenadas. En errores de escritura (en la versión demo actual) se ignora en silencio — sin UI de error porque es una vía demo.

El botón se mantiene atenuado mientras no haya corrido ningún trial.

 **EN POCAS PALABRAS**

Guarda los valores de parámetro del mejor trial hasta el momento como JSON. Después puedes copiar manualmente esos valores a tu configuración de ajuste preestablecido.

W35 Selector «Search Space» preset**DÓNDE**

Barra lateral izquierda, arriba.

TÉCNICO

Selector tipo menú con cuatro opciones preset:
 - «RadianceKit defaults (6-dim)» — el espacio de búsqueda estándar completo con todos los hiperparámetros Q7. - «Mip subset (2-dim)» — solo `mipSmoothing3DScale` [0,05, 0,5] log-uniforme y `mipFilter2DVariance` [0,1, 0,6] lineal. Útil cuando quieres tunear Mip-Splatting para una clase de escena. - «densify-until + ssim-weight + grad-thresh» — tres parámetros relevantes para Densify (`densifyGradThreshold` log-uniforme, `ssimWeight` lineal, `densifyUntilIter` integer). - «Bowl demo (1-dim)» — espacio de búsqueda pedagógico de un solo parámetro para demos «así es como funciona BayesOpt».

Mientras hay un run activo, el espacio de búsqueda no se puede cambiar (confundiría al optimizador).

EN POCAS PALABRAS

Qué espacio de búsqueda de hiperparámetros explora BayesOpt. El default es «RadianceKit defaults». Para intentos de tuning Mip dirigido, «Mip subset». Para entender cómo funciona BayesOpt, «Bowl demo».

W36 Deslizador «Trial Budget»**DÓNDE**

Barra lateral izquierda, bajo el selector de espacio de búsqueda.

TÉCNICO

Deslizador de 10 a 200, paso 5. Default 40. Eso significa: BayesOpt puede hacer un máximo de N trials. De ellos, los primeros son muestras iniciales (Latin hypercube), el resto son trials BayesOpt reales. Reglas prácticas: un espacio de búsqueda con d dimensiones necesita alrededor de $10*d$ a $20*d$ trials para un buen óptimo. En los defaults de 6-dim son 60–120, en el subset Mip de 2-dim 20–40, en la Bowl demo de 1-dim 10–20.

Durante el run el deslizador está deshabilitado.

EN POCAS PALABRAS

Cuántos intentos de optimización como máximo. Más intentos = mejor solución, pero cuesta más tiempo. 40 es un buen default para el objetivo demo.

W37 Deslizador «Random Seed»**DÓNDE**

Barra lateral izquierda, bajo el deslizador de budget.

TÉCNICO

Deslizador de 1 a 100, paso 1. Default 42. La seed se pasa tanto a las muestras iniciales Latin hypercube como al componente de ruido del objetivo demo. Reproducibilidad: misma seed + mismo espacio de búsqueda + mismo budget da exactamente la misma secuencia de trials. Útil para «¿obtienen todos tus compañeros el mismo run cuando reconstruyen la demo?». Deshabilitado durante el run.

EN POCAS PALABRAS

Controla el generador aleatorio. Misma seed = mismo run — para reproducir.

W38 Chart (convergencia)**DÓNDE**

Columna central de la ventana.

TÉCNICO

Diagrama Swift Charts con dos capas: 1. una línea para «mejor valor hasta el momento» por trial — una curva monótonicamente creciente o constante en color de acento. 2. un punto por trial con el valor objetivo individual, coloreado por fase. Tamaño de símbolo 40. Tres etiquetas de fase: «init» (gris), «bo» (azul), «restart» (naranja).

Una pequeña leyenda muestra los colores de fase arriba a la izquierda. Si la lista de trials está vacía (antes del primer inicio), en su lugar se muestra una visualización empty-state con un icono de chart y el aviso «Press Start to begin a BayesOpt run.»

EN POCAS PALABRAS

El gráfico de progreso. La línea sólida es «la mejor solución encontrada hasta el momento»; los puntos son los intentos individuales. Si la línea se mantiene plana durante mucho tiempo, BayesOpt ha encontrado el óptimo.

W39 Table (log de trials)**DÓNDE**

Columna derecha de la ventana.

TÉCNICO

Área de scroll con filas de trial apiladas de forma lazy. Un stack horizontal por fila: número de trial (monoespaciado de 3 dígitos, a la izquierda), valor (monoespaciado, alineado a la derecha, 70 pt de ancho), tag de fase (cápsula, rellena con color de fase al 25 % de opacidad), opcionalmente una estrella amarilla si ese trial es ahora mismo el mejor. Un mecanismo de auto-scroll salta automáticamente al final en cuanto se añade un nuevo trial — así puedes seguir la progresión en vivo al pie de la pantalla sin desplazarte tú mismo.

EN POCAS PALABRAS

La tabla de todos los intentos. Valor, fase, estrella para el mejor. Auto-scrollea, los nuevos trials aparecen abajo.

Ventana principal: curva de loss y número de Gaussianos (I39–I41, referencia cruzada)

Tres de las visualizaciones del Inspector en la ventana principal merecen su propia explicación porque son constantemente visibles durante un entrenamiento en curso y hay reglas prácticas importantes para cuando la curva se ve sana. Las visualizaciones están en el Inspector bajo la sección «Loss Chart» (véase Capítulo 2 — Inspector) y complementan el análisis Holdout de la ventana auxiliar de arriba.

¿Cuándo es sana la curva de loss? Una curva de loss sana muestra tres fases: (1) **Warmup** — en las primeras 200–500 iteraciones, el loss cae con fuerza desde alto (típicamente 0,15–0,25 para L1+SSIM combinados según la escena) a aproximadamente la mitad. Si el loss NO cae en esta fase, la entrada suele ser incorrecta (imágenes rotas, malas poses SfM, muy pocos Gaussianos iniciales). (2) **Densification** — entre ~500 y densifyUntilIteration (classic 15K, MCMC hasta 20K o 25K) el loss sigue cayendo, a menudo con pequeños saltos hacia abajo cuando las operaciones densify insertan nuevos Gaussianos y el optimizador los aprovecha. El número de Gaussianos sube en esta fase. (3) **Refinement** — después de eso el loss entra en una cola que se aplana. Valores finales típicos: Tanks-&-Temples Truck con P4 Quality aterriza en $L1 \approx 0,023$, Horse con Full Classic V546 en $L1 \approx 0,0230$, escenas exteriores Mip-NeRF360 a menudo peor (0,04–0,07).

¿Qué significa una meseta? Una meseta (la curva de loss corre horizontal durante varios miles de iteraciones) tiene dos interpretaciones: (a) el modelo ha convergido, entrenar más no aportará nada — el buen caso. (b) el modelo está atascado (mínimo local, mala información de gradiente, tope al límite del buffer) — el mal caso. Ambos se ven idénticos en el gráfico. Distinción: mira el número de Gaussianos. Si también está plano Y cerca del tope MCMC (p. ej., 150K de 150K en `.fullMCMC`), estás en el límite —

o sube el tope o acepta la meseta. Si el número de Gaussianos sigue creciendo pero el loss no cae, está atascado.

¿Cuándo abortar vs continuar entrenando? Regla práctica: 10K iteraciones sin mejora del min loss → abortar, las iteraciones siguientes son tiempo perdido. Antes de eso: puedes añadir una extensión vía Cmd+T (menú Training → Continue Training → +5K iterations), si ves una mejora marginal. Atención: con MCMC la meseta es a menudo real — el tope es el límite natural.

Una meseta del número de Gaussianos NO es una señal de «hecho». Solo significa que MCMC ha alcanzado el tope o que la densificación Classic se ha agotado. La pregunta real de «hecho» solo la responde el análisis Holdout — PSNR/SSIM/LPIPS sobre un conjunto de test independiente, evaluado en la ventana Holdout (W23–W29) o mediante el flag `--benchmark`.

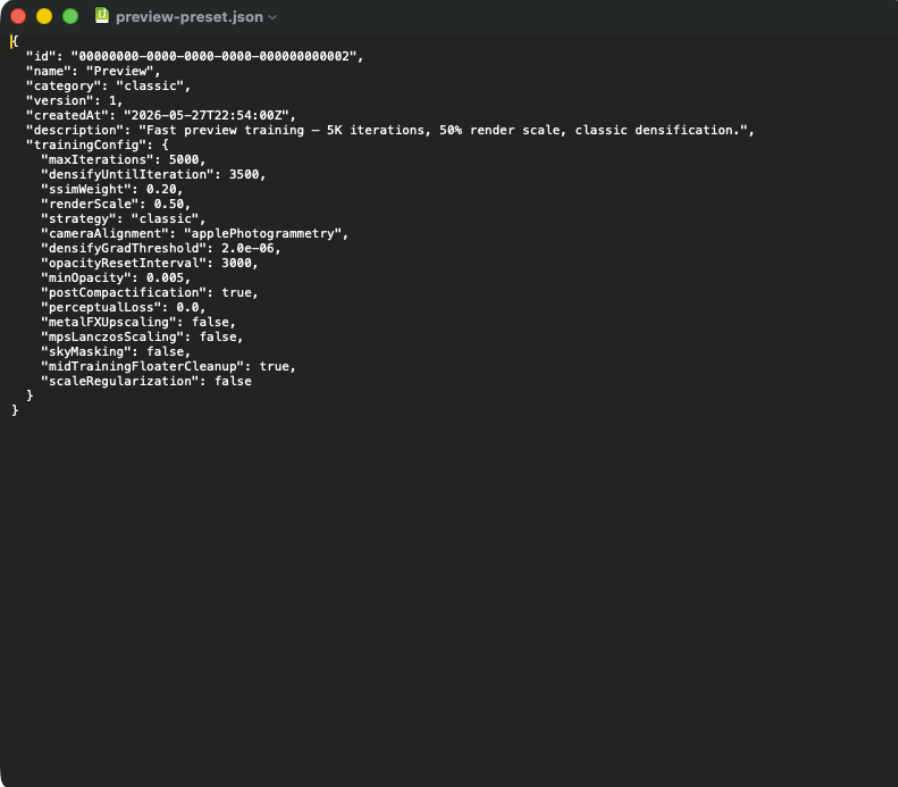
PSNR/Holdout es la verdad, loss es solo un proxy. El loss es una métrica relativa: cae a medida que tu modelo se ajusta a las vistas de entrenamiento. Un loss bajo no significa automáticamente un buen modelo — si el modelo ha memorizado las imágenes de entrenamiento (overfitting), el loss sería pequeño, pero el PSNR sobre vistas no vistas (Holdout) sería malo. Por tanto: para la evaluación final de calidad, siempre mira métricas Holdout, no solo el loss final.

Caja de reglas prácticas

- User Guide y Keyboard Shortcuts son ayuda estática — rápidos para preguntas de palabra clave, para profundidad ten este manual a mano.
- Abre Manage Storage en cuanto el disco caiga por debajo del 10 % de espacio libre. Los logs y el staging de imports son los sospechosos habituales.
- Pareto Dashboard solo es útil tras al menos tres o cuatro reportes de entrenamiento. Eje X = costes (Time / Gs), eje Y = calidad (PSNR / SSIM). El frente de Pareto muestra las combinaciones eficientes.
- Usa Holdout Analysis antes de publicar benchmarks de PSNR con otros — te asegura que tu conjunto de test es realmente representativo.
- BayesOpt Console es ante todo una herramienta de aprendizaje y de inspección para definiciones de espacio de búsqueda. Para un tuning real de hiperparámetros dirigido por entrenamiento, usa el flujo CLI offline.
- La meseta de loss y la meseta del número de Gaussianos deben interpretarse por separado. El límite del tope no es una señal de «hecho». La calidad real se mide solo por PSNR Holdout.
- 10K iteraciones sin mejora del min-loss → detener el entrenamiento.

CAPÍTULO

Capítulo 6 — Configuración del entrenamiento



```
{
  "id": "00000000-0000-0000-0000-000000000002",
  "name": "Preview",
  "category": "classic",
  "version": 1,
  "createdAt": "2026-05-27T22:54:00Z",
  "description": "Fast preview training - 5K iterations, 50% render scale, classic densification.",
  "trainingConfig": {
    "maxIterations": 5000,
    "densifyUntilIteration": 3500,
    "ssimWeight": 0.20,
    "renderScale": 0.50,
    "strategy": "classic",
    "cameraAlignment": "applePhotogrammetry",
    "densifyGradThreshold": 2.0e-06,
    "opacityResetInterval": 3000,
    "minOpacity": 0.005,
    "postCompactification": true,
    "perceptualLoss": 0.0,
    "metalFXUpscaling": false,
    "mpslanczosScaling": false,
    "skyMasking": false,
    "midTrainingFloaterCleanup": true,
    "scaleRegularization": false
  }
}
```

Figura 26: Ajuste preestablecido Preview exportado como JSON y visualizado en TextEdit — campos id/name/category/version/createdAt/ description, trainingConfig con todos los parámetros relevantes (maxIterations 5000, densifyUntilIteration 3500, ssimWeight 0,20, renderScale 0,50, strategy classic, cameraAlignment applePhotogrammetry, densifyGradThreshold 2.0e-06, opacityResetInterval 3000, minOpacity 0,005, seis conmutadores booleanos)

LO QUE SE VE EN LA IMAGEN Una exportación típica de un ajuste preestablecido como JSON. Campos de nivel superior: `id` (UUID), `name`, (`classic` | `mcmc` | `sceneClass` | `custom`), (`versión de schema`), (`timestamp`), (`texto libre`). El objeto anidado contiene los parámetros críticos para la reproducibilidad — en la importación, todo el bloque se deserializa en el struct `TrainingConfig`, y los defaults de la versión actual de la app rellenan los campos que falten en el JSON (p. ej., tras una actualización de la app). Para pasar un ajuste preestablecido a otro Mac, basta con enviar este archivo JSON.

El struct `TrainingConfig` es el corazón de cada run de entrenamiento en RadianceKit. Recopila todos los parámetros que influyen en el entrenamiento — desde el número máximo de iteraciones, pasando por las ocho learning rates, hasta los campos especiales para MCMC, Mip-Splatting, el currículum y la lógica de tope consciente de la escena. Lo editas en la barra lateral en la sección Training Configuration (Vista Experto), lo guardas como ajuste preestablecido o lo entregas como exportación JSON a otro Mac. Al iniciar el entrenamiento, este mismo objeto se congela y se entrega al backend de GPU.

Este capítulo es material de referencia para power users y autores de scripts. Lista los 81 campos públicos, los 9 ajustes preestablecidos estáticos y el único método público. El archivo fuente es `TrainingConfig.swift` — en caso de duda, el comentario de doc almacenado allí y el valor por defecto del inicializador son la fuente de verdad.

AVISO · UI VS. PRESET/CLI

Solo 12 de los 81 campos tienen un deslizador, conmutador o selector directo en el Inspector (build de App Store sandbox): **T1, T2, T17, T20, T22, T38, T56–T58, T60, T61, T73**. Los 69 campos restantes se fijan mediante el **ajuste preestablecido** elegido (Capítulo 7) y solo pueden sobrescribirse directamente mediante un **flag CLI** (véase Capítulo 5). Esta separación es a propósito: los defaults se mantienen estables y probados en producción, mientras que los power users siguen teniendo una vía de escape. Si un campo te interesa de verdad, comprueba primero el Capítulo 2 (Inspector) y el Capítulo 5 (CLI) para ver si llegas a él sin trastear el JSON.

Índice:

1. Iteración (T1–T2)
2. Learning Rates (T3–T10)
3. Densificación — Classic (T11–T16)
4. Loss (T17–T20)
5. Progresión de grado SH (T21)
6. Rendimiento (T22–T25)
7. Diagnóstico y preparación de la nube de puntos (T26–T30)
8. Regularización (T31–T37)
9. Refinamiento (T38–T44)
10. Sky Dome (T45–T48)
11. Adam + schedule de LR (T49–T55)
12. Post-procesado + Apple AI (T56–T60)
13. Densificación MCMC (T61–T73)
14. Mip-Splatting (Q1.5) (T74–T76)
15. Densificación adaptativa (Q5) (T77–T79)
16. Currículum (Q6) (T80–T81)
17. Ajustes preestablecidos estáticos (TP1–TP9)
18. Método:
19. ¿Qué campo para qué? (Cheat Sheet)
20. Campos peligrosos

Iteración (T1–T2)

T1 maxIterations

DETALLES

Predeterminado: 30 000 (initializer), 35 000 (`.full`), 200 000 (`.fullMCMC`) **Rango:** 1 000 – 500 000 (deslizador UI), sin límite superior duro en la lógica **Definido en:**

TÉCNICO

Número total de iteraciones de entrenamiento que ejecuta el backend. Una iteración significa un forward render de una sola cámara de entrenamiento, un backward pass sobre todos los componentes del loss (L1 + SSIM + regularizaciones opcionales + máscara de cielo) y un paso del optimizador Adam. Este número controla directamente los demás schedules: la learning rate de posición sigue una curva cosine annealing de 0 a $T1$ mismo o a $T49$ `positionLRScheduleEndIteration`; la densificación se detiene en $T2$ `densifyUntilIteration`; el decay de ruido MCMC termina en $T69$ `mcmcNoiseDecayEnd`; los upgrades de grado SH ocurren en las tres marcas de $T21$. Para la densificación clásica, el sweet spot empírico está en 20 000–35 000 iteraciones (Sesiones 1–32, tests V546), para MCMC en 60 000–200 000 (V534). Empujar mucho más allá de los valores almacenados en el ajuste preestablecido rara vez aporta calidad adicional — el momentum de Adam se satura, y sin un final de decay de LR el loss se estanca. Al contrario, bajar de ~5 000 lleva a geometría incompletamente convergida (la density control tiene muy poco tiempo para clone/split).

EN POCAS PALABRAS

Cuánto tiempo entrena la app. Más iteraciones = mejor resultado, pero a partir de cierto punto ya no se nota mejor, solo dura mucho más. Los ajustes preestablecidos están elegidos para que, sin pensar, obtengas un buen valor: Quick 1 000, Preview 5 000, Balanced 20 000, Quality 35 000, MCMC Quality 200 000. Si mueves la perilla tú: con MCMC, sube tranquilamente (100 000–200 000); con Classic, no por encima de 40 000 — a partir de ahí no ayuda más.

T2 densifyUntilIteration

DETALLES

Predeterminado: 15 000 (initializer), 5 000 (`.full`), 160 000 (`.fullMCMC`) **Rango:** 0 – **Definido en:**

TÉCNICO

Iteración en la que se detiene la densificación. Hasta este punto, los Gaussianos se clonan, dividen y podan según las reglas parametrizadas en T11–T16 (Classic) o T67–T70 (MCMC); después, el número de Gaussianos se mantiene constante y solo se optimizan posiciones, rotaciones, escalas, opacidades y coeficientes SH (fase de refinamiento). En el paper original de 3DGS, el valor está en el 50 % de T1, en el preset `.full` de RadianceKit solo en el ~14 % (5 000 de 35 000) — consecuencia de los experimentos V310/V338 que mostraron que tras 5 000 iteraciones más densificación empeora el resultado (más floaters, mayor uso de memoria, sin ganancia de calidad). MCMC, en cambio, ejecuta la relocalización hasta el 80 % de T1 (V504b) porque MCMC no produce floaters dañinos. Si T2 es demasiado pequeño (< 1 000), surgen muy pocos Gaussianos; demasiado grande bajo Classic (> 50 % de T1) lleva a sobrecrecimiento y a outliers de saturación RGB (véase Outdoor Overtraining Findings).

EN POCAS PALABRAS

Hasta cuándo la app puede crear nuevos Gaussianos. Después, solo se refina lo que ya existe. Con entrenamiento clásico a 35 000 iteraciones, 5 000 es aquí el valor correcto — cualquier valor superior hace la escena más blanda. Con MCMC, es el 80 % del total de iteraciones (es decir, 160 000 para un run de 200 000). Si cambias el ajuste preestablecido Quality, mejor deja este campo en paz.

Learning Rates (T3–T10)

T3 positionLearningRate

DETALLES

Predeterminado: 0,00016 **Rango:** 1e-7 – 1e-3 (recomendado) **Definido en:**

TÉCNICO

Learning rate de Adam para la posición XYZ de cada Gaussiano al inicio del entrenamiento (iteración 0). Sigue una curva cosine annealing y decae a lo largo del entrenamiento hasta T4 positionLearningRateFinal. El predeterminado 0,00016 viene del paper original de 3DGS (Kerbl et al. 2023) y en RadianceKit no se escala ni siquiera con mayor resolución de imagen — la posición se mueve en coordenadas mundo, no en espacio de píxeles. Un aumento claro ($> 0,0005$) hace que los Gaussianos salten grandes distancias y el loss se vuelve inestable; valores muy por debajo ($< 0,00005$) hacen que nubes de puntos mal inicializadas no encuentren nunca su sitio. V414 probó duplicar el valor inicial \rightarrow 16,8 % peor loss L1; los tuneos V544a confirmaron el default del paper como óptimo. Nota: bajo `.fullMCMC` lo dejamos deliberadamente en el default — MCMC necesita learning rates constantes para su lógica de relocalización, así que afinar aquí no aporta nada.

EN POCAS PALABRAS

Lo rápido que pueden moverse los puntos splat por el espacio. El valor predeterminado está muy bien afinado y básicamente no necesita cambios. Solo si los splats «tambalean» en la imagen o falta toda una esquina porque ahí nada se mueve sería el learning rate algo a ajustar — pero entonces típicamente hay algo mal aguas arriba (poses de cámara, nube de puntos inicial).

T4 positionLearningRateFinal **DETALLES**

Predeterminado: 0,0000016 (initializer + paper), 0,000016 (`.full` , `.fullMCMC` — 10× mayor) **Rango:** 0 — **Definido en:**

 **TÉCNICO**

Valor final de la curva cosine annealing de LR de posición. Se alcanza ya sea en `T1 maxIterations` o, si está fijado, en `T49 positionLRScheduleEndIteration`. El preset `.full` de RadianceKit usa 0,000016 — es decir, 10× más alto que el default del paper 0,0000016. Los experimentos V420 mostraron que 0,5× del valor final (0,000008) hace el loss un 6,4 % peor; V414 mostró que 2× el valor inicial lo empeora un 16,8 %. El valor final alto no es un trade-off sino una elección deliberada: con un decay demasiado agresivo, los Gaussianos pierden durante la fase de refinamiento la capacidad de reaccionar a candidatos de densificación recién llegados. Mediante la extensión V431/V433, la fase del schedule se puede acortar ($T49 < T1$), de modo que `T4` se alcance antes del final del entrenamiento y el resto del entrenamiento corra con el mini-LR constante — configuración típica: `T49 = 20 000` , `T1 = 35 000` , refinamiento por tanto en 0,000016 durante 15 000 iteraciones.

 **EN POCAS PALABRAS**

Lo despacio que se vuelve la learning rate de posición al final del entrenamiento. Lo hemos puesto deliberadamente menos agresivo que el paper original — los splats aún pueden moverse un poco hasta el final, lo que los hace más nítidos. Si mueves la perilla: más alto = splats más inquietos al final, más bajo = los splats ya no se pueden adaptar cuando aparecen nuevos.

T5 shDCLearningRate **DETALLES**

Predeterminado: 0,0025 (initializer + paper), 0,005 (`.full` y todos los ajustes preestablecidos MCMC – 2x) **Rango:** 0,0001 – 0,05 **Definido en:**

 **TÉCNICO**

Learning rate de Adam para el componente DC (grado 0, es decir, albedo constante) del color spherical harmonic. El SH-DC corresponde al tono base independiente de la dirección de un Gaussiano, esencialmente el «color base». Los experimentos V176 y V188 encontraron 2x más alto que el default del paper como óptimo — convergencia de color más rápida, especialmente porque con entrenamiento corto (5 000 iteraciones), el SH-DC en otro caso no converge. A diferencia de las LRs geométricas, el SH-DC no tiene decay; la learning rate se mantiene constante sobre todas las iteraciones (o solo sigue el decay opcional de fase extendida de `T51`). V416 probó cuadruplicar a 0,01 → loss 6,4 % peor con Adam beta2=0,99.

 **EN POCAS PALABRAS**

Lo rápido que se adapta el color base de cada splat. Casi nunca cambias este valor tú mismo — los ajustes preestablecidos tienen el valor correcto. Más alto sería más rápido pero puede llevar a colores inestables.

T6 shRestLearningRate **DETALLES**

Predeterminado: 0,000125 (initializer + paper), 0,00025 (`.full` y MCMC — 2x) **Rango:** 0,000001 – 0,005 **Definido en:**

 **TÉCNICO**

Learning rate de Adam para los coeficientes SH de orden superior (grados 1, 2, 3 — los componentes de color dependientes de la dirección de visión que producen brillos, reflejos y sombreado suave). 20x más pequeña que **T5** por convención del paper, porque esos coeficientes crecen cuadráticamente en número (3 para grado 1, 5 para grado 2, 7 para grado 3 → 15 floats totales por Gaussiano) y sin una learning rate menor sobresaturarían la imagen. Se desbloquean en dos pasos — hasta la primera marca en **T21** `shDegreeUpgradeIterations` solo está activo el grado 0 (así que solo **T5**), después el 1, luego el 2, finalmente el 3. Los valores bajos aquí son particularmente importantes en escenas con mucha iluminación difusa; en superficies muy especulares (pintura de coche, agua) afinar no ayuda — la propia representación SH está limitada.

 **EN POCAS PALABRAS**

Lo rápido que aprenden los efectos de color dependientes de la vista (reflejos, brillos). Por defecto muy pequeño, en otro caso todo empieza a brillar. Mejor deja el valor — si quieres mejores brillos, te sirve más MCMC y más tiempo de entrenamiento que esta LR.

T7 `opacityLearningRate` **DETALLES**

Predeterminado: 0,05 (initializer + paper), 0,1 (`.full`, MCMC — 2×) **Rango:** 0,001 – 1,0 **Definido en:**

 **TÉCNICO**

Learning rate de Adam para la opacidad logit de cada Gaussiano. La app almacena la opacidad como un valor float no acotado y la transforma con sigmoide a $[0, 1]$; la LR actúa en espacio logit. El default 0,05 del paper se restauró tras los tests V50 (mejor L1 single-run 0,1664), V71 revirtió el 0,025 de V67. La duplicación V188 a 0,1 hace el pruning más eficiente — los Gaussianos muertos caen más rápido por debajo del T14 `pruneOpacityThreshold`. V418 mostró: 0,05 con Adam `beta2=0,99` es un 7,1 % peor que 0,1 — la interacción con la configuración de Adam no es trivial. Valores bajos ($< 0,01$) hacen que los Gaussianos «muertos» persistan eternamente y desperdicien memoria; valores demasiado altos ($> 0,5$) pueden llevar a explosión de opacidad, por lo que el valor logit en el optimizador se hace clamp a $[-15, 3]$ (véase la nota «Opacity Explosion Prevention» en CLAUDE.md).

 **EN POCAS PALABRAS**

Lo rápido que los splats se vuelven transparentes u opacos. Importante para la limpieza — los splats que no contribuyen tienen que desaparecer rápido para que no aparezca neblina. El valor predeterminado encaja, solo lo cambian los profesionales.

T8 `opacityLearningRateFinal` **DETALLES**

Predeterminado: 0,0 (= «sin decay») **Rango:** 0 o 0,001 – **Definido en:**

 **TÉCNICO**

Valor final opcional de decay cosine para la LR de opacidad (V427). Cuando es 0,0, el decay está deshabilitado y la LR de opacidad se mantiene constante en **T7** durante todo el entrenamiento. V427 probó un decay 0,1 → 0,01 — resultado fue loss 11,5 % peor; revertido, de ahí el predeterminado «off». La hipótesis tras el campo: en la fase de refinamiento, una LR de opacidad constante podría llevar a oscilación, de modo que splats que ya hubieran alcanzado el nivel correcto de transparencia serían empujados de un lado a otro por fluctuaciones aleatorias de gradiente. Empíricamente no es así — la lógica de clamp logit ya lo atrapa. El campo se mantiene disponible para experimentos futuros; runs MCMC muy largos (> 500K iteraciones) también podrían beneficiarse.

 **EN POCAS PALABRAS**

Si la learning rate de opacidad debe hacerse más pequeña hacia el final. Predeterminado: no. Lo probamos, fue peor, lo dejamos off. Quédate en 0.

T9 `scaleLearningRate` **DETALLES**

Predeterminado: 0,005 (initializer + paper), 0,01 (`.full`, MCMC — 2×) **Rango:** 0,0001 – 0,1 **Definido en:**

 **TÉCNICO**

Learning rate de Adam para los tres componentes de escala de cada Gaussiano en espacio logarítmico (RadianceKit almacena $\log(\text{scale})$ para que las escalas se mantengan positivas). El default del paper 0,005, duplicado en RadianceKit a 0,01 para mejor convergencia de escala con las configuraciones de learning rate optimizadas. Experimento V423: 0,005 con Adam $\beta_2=0,99 \rightarrow$ loss 18,7 % peor y visiblemente muy pocos Gaussianos (la density control no podía clonar porque las actualizaciones de escala eran demasiado perezosas). La escala controla la extensión de cada Gaussiano — un aprendizaje demasiado rápido lleva a Gaussianos «aguja» (splats extremadamente largos y delgados, véase T34 `scaleRatioPruneThreshold`), un aprendizaje demasiado lento mantiene los splats demasiado compactos y la density control tiene que dividir demasiado a menudo.

 **EN POCAS PALABRAS**

Lo rápido que se adapta la forma de los splats. El predeterminado es bueno. Si subes esto, obtienes splats «aguja» — gotas extremadamente largas y delgadas que hacen flotar la imagen.

T10 rotationLearningRate **DETALLES**

Predeterminado: 0,001 (initializer + paper), 0,002 (`.full`, MCMC — 2×) **Rango:** 0,0001 – 0,05 **Definido en:**

 **TÉCNICO**

Learning rate de Adam para los cuatro componentes cuaternión de cada Gaussiano. El cuaternión se re-normaliza tras cada actualización de Adam (norma $L2 = 1$) — en otro caso, la matriz de covarianza degeneraría. RadianceKit duplica el default del paper en los ajustes preestablecidos Quality porque la rotación tiene magnitudes de gradiente absolutas más pequeñas que escala/posición (en la esfera unitaria cada paso se mantiene corto) y sin 2× la rotación estaría claramente sub-convergada en la ventana de 35 000 iteraciones. V188 lo documenta. En escenas NeRF-Blender (Lego, Chair) la rotación importa particularmente — los bordes de objetos solo se alinean correctamente tras 5 000–10 000 iteraciones.

 **EN POCAS PALABRAS**

Lo rápido que los splats aprenden a rotar — es decir, a orientarse correctamente sobre la superficie de un objeto. El predeterminado encaja. Dicho de otro modo: si los splats parecen lomas tumbadas torcidas en vez de abrazar la superficie, el tiempo de entrenamiento es corto antes que esta learning rate baja.

Densificación — Classic (T11–T16)

T11 densifyGradThreshold

DETALLES

Predeterminado: 0,000002 (initializer, calibrado para 0,5x resolución), 0,0000011 (`.full` , calibrado para 1,0x), 0,000004 (`.quickTest` , calibrado para 0,25x), $2e-7$ (`.fullClassicPaper`) **Rango:** $1e-8$ – $1e-3$ (dependiente de resolución) **Definido en:**

TÉCNICO

Umbral para la norma L2 del gradiente proyectado en espacio de pantalla `dMean2D` , por encima del cual un Gaussiano se marca para clonado o división. El valor absoluto depende directamente de la resolución de entrenamiento — `dMean2D` escala aproximadamente como $1/\text{resolución}^2$ (más píxeles = gradientes por píxel más pequeños). De ahí que cada paso de T22 `trainingRenderScale` necesite un umbral calibrado: $0,25x \rightarrow 4e-6$, $0,5x \rightarrow 2e-6$, $1,0x \rightarrow 5e-8$... $1,1e-6$ (`.full`). El default 0,0002 del paper está normalizado NDC y no es directamente comparable con la pipeline en espacio mundo de RadiancEKit. Con el flag T52 `adaptiveDensifyThreshold` introducido en V440, el valor puede derivarse en runtime del p98 de la distribución actual de gradientes — pero V440 lo probó en escenas reales y produjo 63 K Gaussianos (pérdida catastrófica por pruning); el flag se queda off. Q5 (T77–T79) proporciona una lógica adaptativa alternativa vía mediana rodante. **Este campo no es inocuo** — partir a la mitad crea 2–4x más Gaussianos (presión de memoria, riesgo OOM); duplicar puede sub-densificar la escena.

EN POCAS PALABRAS

Lo sensible que es la app al decidir si un splat está sub-representado y debería duplicarse. Valor más bajo = más sensible = más splats. Más alto = menos splats. Es uno de los valores más peligrosos: demasiado bajo y tu Mac se llena con millones de splats y puede caerse. Deja el campo en paz, o cámbialo solo en pasos del 10 %.

T12 densifyFromIteration **DETALLES**

Predeterminado: 500 **Rango:** 100 – 5 000 **Definido en:**

 **TÉCNICO**

Primera iteración en la que la densificación se vuelve activa. Antes solo ocurre aprendizaje «puro» sobre la nube de puntos SfM inicial, sin que se creen nuevos Gaussianos. El default 500 viene del paper 3DGS y da tiempo a la inicialización para estabilizarse — si la densificación arranca ya en la iteración 0, los puntos SfM mal posicionados se clonan muchas veces antes incluso de encontrar su lugar correcto. V349 probó 1000 → loss ligeramente peor; el default es óptimo.

 **EN POCAS PALABRAS**

Cuándo empieza la app a clonar splats por primera vez. Antes solo aprende los puntos ya existentes. 500 es el predeterminado — da a la app suficiente tiempo para orientarse antes de multiplicar.

T13 densifyInterval **DETALLES**

Predeterminado: 100 (initializer, MCMC), 200 (.full) **Rango:** 50 – 1 000 **Definido en:**

 **TÉCNICO**

Cuántas iteraciones hay entre dos pasos de densificación. Default del paper 100 — cada 100 iteraciones se evalúa la lista de candidatos a densificar, se clona/divide, y al mismo tiempo se elimina la lista de candidatos a poda ($\text{sigmoid}(\text{opacity}) < \text{T14 pruneOpacityThreshold}$). Las pruebas V112 encontraron 200 como óptimo para .full — esto descarga la GPU porque corren menos pasadas de reorganización, y da a cada Gaussiano más tiempo para asentarse tras una acción de clonado. V417 probó 100 con Adam $\text{beta2}=0,99 \rightarrow 5,8\%$ peor (957 K Gaussianos, sobre-densificación). Bajo MCMC el mismo campo se interpreta como intervalo de relocalización; véase T67 `mcmcRelocationInterval` para la lógica específica de MCMC.

 **EN POCAS PALABRAS**

Cada cuánto busca la app nuevos splats. 100 = a menudo, 200 = medio. Más alto significa: cada splat tiene más tiempo para asentarse antes de que se le vuelva a multiplicar. Eso es bueno. Reducir a 50 puede mantener la GPU ocupada de forma permanente sin que sea notablemente mejor.

T14 `pruneOpacityThreshold` **DETALLES**

Predeterminado: 0,005 (initializer, paper, MCMC), 0,001 (`.full`) **Rango:** 0,0001 – 0,1 **Definido en:**

 **TÉCNICO**

Umbral de opacidad sigmoide por debajo del cual un Gaussiano se elimina en el siguiente paso de densificación. Funciona junto con `T7` `opacityLearningRate` y la lógica de clamp logit en el optimizador. V393 bajó el default de 0,005 a 0,001 en `.full` — resultado: los splats que solo importan bajo ángulos de visión exóticos se mantienen más tiempo y contribuyen a detalle SH. V394 probó 0,0001 → ligeramente peor (muy poco podado, memoria desperdiciada). Importante: la density control DEBE podar siempre, aun cuando la capacidad de buffer ya esté llena por otras medidas (véase «Density Control Must Always Prune» en CLAUDE.md) — en otro caso, los Gaussianos muertos se acumulan y el número se congela.

 **EN POCAS PALABRAS**

Cuándo un splat cuenta como «lo bastante transparente» para ser eliminado. 0,005 es el default del paper, en Quality tenemos 0,001 — es decir, damos a los splats más oportunidad. Eso hace más fáciles de representar la luz suave y las sombras tenues. Subir (por encima de 0,01) hace caer rápido el número de splats — puede tener sentido con presión de memoria pero cuesta detalle.

T15 `opacityResetInterval` **DETALLES**

Predeterminado: 3 000 (initializer + paper), 100 000 (`.full` = efectivamente deshabilitado), 200 000 (`.fullMCMC` = deshabilitado) **Rango:** 1 000 – 100 000+ **Definido en:**

 **TÉCNICO**

Cuántas iteraciones entre resets de la opacidad de todos los Gaussianos a un valor bajo ($\sim 0,01$) — una medida del paper 3DGS para reevaluar splats «congelados». V194 mostró que con el setup de warmup + entrenamiento estocástico + LRs 2× de RadianceKit, el reset de opacidad cuesta un 5,5 % de calidad y que la lógica de clamp logit ya cubre la función reset. De ahí que en `.full` esté prácticamente deshabilitado (100 000 > 35 000, así que nunca se dispara). V421 probó reset cada 3 000 con Adam beta2=0,99 → 4,9 % peor; revertido. Bajo `.fullClassicPaper` (test paper-true Q1.5-A) se vuelve a poner deliberadamente en 3 000 — esa fue una de las palancas con las que se intentó alcanzar los presupuestos de Gaussianos de magnitud-paper.

 **EN POCAS PALABRAS**

Cuántas iteraciones entre que la app resetea la visibilidad de todos los splats a «casi invisible» — una especie de botón de reset para la opacidad. En nosotros deshabilitado (valor tan alto que nunca dispara) porque otros mecanismos lo hacen innecesario. Solo habilitar para experimentos fieles al paper.

T16 maxScreenSize **DETALLES**

Predeterminado: 0,0 (= deshabilitado) **Rango:** 0 (off) o > 0 **Definido en:**

 **TÉCNICO**

Tamaño máximo en espacio de pantalla (en píxeles proyectados) que un Gaussiano puede alcanzar antes de dividirse forzosamente. El valor está en 0 (V48 probado y revertido) — la density control de RadianceKit usa en su lugar el umbral de escala en espacio mundo de la lógica `dMean2D`. Se queda en el catálogo de campos porque futuros experimentos con Mip-Splatting (T74–T76) o estrategias de splatting específicas de escena podrían beneficiarse. Habilitar (valor > 0, p. ej., 20) forzaría a los splats que han crecido mucho en pantalla a subdividirse — relevante con superficies de pared grandes y suaves donde un único splat gigante ofrece muy poco detalle.

 **EN POCAS PALABRAS**

Límite de cuán grande puede llegar a ser un splat en pantalla. En nosotros, off. Activarlo haría que los splats planos enormes (p. ej., en una pared) se rompieran forzosamente en varios más pequeños. Déjalo off salvo que estés experimentando explícitamente con ello.

Loss (T17–T20)

T17 ssimWeight

DETALLES

Predeterminado: 0,2 (initializer + paper + `.full`), 0,05 (todos los ajustes preestablecidos MCMC)

Rango: 0,0 – 1,0 **Definido en:**

TÉCNICO

Peso del término D-SSIM en la función de loss combinada $loss = (1 - \lambda) * L1 + \lambda * D\text{-SSIM}$, donde $\lambda = T17$. El default 0,2 del paper 3DGS es óptimo para la densificación clásica — V383 probó 0,3 → 28,9 % peor, V373b confirmó 0,2 como sweet spot. Para MCMC se estableció independientemente en V521b/V534: 0,05 es óptimo porque MCMC, por su exploración estocástica, necesita un componente de señal L1 más fuerte — pesos SSIM más altos diluirían las decisiones de relocalización. SSIM es significativamente más caro de calcular que L1 (ventanas locales 11×11 sobre toda la imagen); RadianceKit usa una implementación acelerada por MPS que se mantiene por debajo de 1 ms por imagen 1080p. Los sweeps Q7 BayesOpt encontraron óptimos específicos de escena entre 0,05 (`.outdoorPreset` : 0,082) y 0,171 (`.indoorPreset`).

EN POCAS PALABRAS

Lo importante que considera la app «las estructuras son similares» frente a «cada píxel coincide». 0,2 es el predeterminado y da una buena imagen. Más bajo = más preciso a nivel de píxel pero puede dar transiciones más suaves. Más alto: estructuralmente más similar pero los detalles se suavizan. Deja que decidan los ajustes preestablecidos.

T18 **ssimWeightRefinement** **DETALLES**

Predeterminado: 0,0 (= «sin cambio, mantener ssimWeight») **Rango:** 0 o 0 – 1,0 **Definido en:**

 **TÉCNICO**

Valor SSIM opcional para la fase de refinamiento tras T2 `densifyUntilIteration`. V428 probó 0,2 → 0,3 en refinamiento → 16 % peor loss (tanto L1 como SSIM degradados); revertido, de ahí el default 0,0. La hipótesis tras el campo era que tras la densificación — cuando ya no se crean nuevos Gaussianos — una mayor parte SSIM maximizaría la nitidez estructural. Empíricamente falso: subir el peso SSIM significa indirectamente bajar el peso L1, y L1 es la señal mucho más significativa en la fase final de refinamiento. El campo se mantiene disponible para experimentos futuros con perceptual loss (T60) o edge loss (T19), donde una composición de loss específica de refinamiento podría tener sentido.

 **EN POCAS PALABRAS**

Ajuste especial para la segunda fase de entrenamiento (refinamiento tras la duplicación de splats). En 0,0: la misma ponderación SSIM que antes. Trastear no aporta empíricamente, así que off.

T19 **edgeLossWeight** **DETALLES**

Predeterminado: 0,0 (= deshabilitado) **Rango:** 0 o 0,001 – 1,0 **Definido en:**

 **TÉCNICO**

V437 loss experimental: peso de un loss L1 en dominio gradiente Sobel que compara directamente los bordes de imagen (Sobel ground-truth vs Sobel del render) por encima de L1+SSIM. Hipótesis: la información de borde es una piedra angular perceptual de la calidad de imagen y un término explícito debería animar a los Gaussianos a acertar mejor los bordes. Resultados de prueba: peso 0,1 → 11 % peor loss, 0,01 → neutro para calidad pero 10 % más lento. La pasada Sobel cuesta un forward MPS extra sobre ground truth y render. De ahí que esté deshabilitado permanentemente. Caso de uso futuro: escenas con bordes artificiales duros (arquitectura, mobiliario, renderizados) podrían beneficiarse — los ajustes Scene Class de Q7 sin embargo no lo eligieron, escalaron en su lugar el peso SSIM.

 **EN POCAS PALABRAS**

Extra experimental que trata los bordes como especialmente importantes. Empíricamente no aporta nada. Sigue off.

T20 skyMaskingEnabled **DETALLES**

Predeterminado: false (initializer y todos los ajustes preestablecidos) **Rango:** booleano **Definido en:**

 **TÉCNICO**

Habilita el enmascaramiento de cielo. En cada imagen se enmascara la región de cielo mediante Apple Vision Framework (VNGenerateForegroundInstanceMaskRequest), y el loss en esa región se pone a cero. Razón: las escenas exteriores sufren a menudo de píxeles de cielo azul/gris/blanco que llevan a la app a colocar Gaussianos exactamente ahí — lo que se percibe como «floaters». Sin máscara de cielo, el loss en esa región nunca sería cero, porque el cielo en la imagen varía ligeramente y la app sigue intentando reconstruirlo con splats. La máscara Vision se calcula una vez por cámara antes del entrenamiento y se mantiene en RAM. Típicamente se activa junto con `T45 skyDomeEnabled` (lógica UI en la vista de Ajustes). Déjalo deshabilitado para escenas interiores o renderizados sintéticos — la máscara detectaría erróneamente techos o paredes como «cielo».

 **EN POCAS PALABRAS**

Activa un modo especial para capturas exteriores: el cielo se ignora durante el entrenamiento, así que la app no intenta reconstruirlo con splats. Recomendado para cada escena exterior. Déjalo off para interiores o para renderizados 3D de Blender.

Progresión de grado SH (T21)

T21 shDegreeUpgradeliterations

DETALLES

Predeterminado: [1_000, 2_000, 3_000] (initializer), [2_000, 5_000, 8_000] (.full, MCMC), [1_000, 2_000] (.preview — grado 3 saltado) **Rango:** [Int], cada valor en [0, maxIterations], monótonamente creciente

Definido en:

TÉCNICO

Iteraciones en las que el grado SH activo se sube 0→1, 1→2, 2→3. Antes de la primera marca solo están activos los componentes DC (es decir, T5 shDCLearningRate), después de la primera marca DC + 3 coeficientes de grado 1, después de la segunda marca + 5 coeficientes de grado 2, después de la tercera marca los 15 coeficientes. El consumo de memoria por Gaussiano crece por pasos en consecuencia — 4 floats → 16 floats → 36 floats → 64 floats. Los ajustes preestablecidos Quality retrasan los upgrades respecto a los defaults del initializer (V228) porque la geometría debería estabilizarse primero, antes de que los detalles de color con su frecuencia más alta se añadan encima. V384 probó [1K, 2K, 3K] para .full → 9,3 % peor — confirma el retraso. .preview se topa en grado 2, porque el grado 3 no converge en 5 000 iteraciones y solo consume capacidad de optimizador. Q6 (T80–T81) ofrece una lógica de currículum alternativa que sobrescribe esta lista de forma dinámica.

EN POCAS PALABRAS

En qué puntos durante el entrenamiento la app aprende que los colores pueden verse distintos desde distintos ángulos de visión (brillos, reflejos). Solo tarde — así primero la forma es correcta, después el color. Los valores en los ajustes preestablecidos están puestos para que esto funcione bien. No cambies nada salvo que sepas exactamente por qué.

Rendimiento (T22–T25)

T22 trainingRenderScale

DETALLES

Predeterminado: 1,0 (initializer, `.full`, MCMC, Scene-Class), 0,5 (`.preview`), 0,25 (`.quickTest`)

Rango: 0,05 – 2,0 (típicamente 0,25, 0,5, 1,0) **Definido en:**

TÉCNICO

Resolución de renderizado durante el entrenamiento relativa a la resolución original de las imágenes de entrenamiento. En 0,5 cada imagen se reduce al 50 % de ancho × 50 % de alto (es decir, 25 % de los píxeles) y el renderizado Gaussiano ocurre a esa resolución más baja. Reduce cuadráticamente tanto memoria como cómputo. Importante: T11 `densifyGradThreshold` tiene que coincidir con la resolución elegida — las magnitudes de gradiente escalan con $1/\text{resolución}^2$, por lo que `.quickTest` (0,25×) tiene un umbral mucho más alto ($4e-6$) que `.full` (1,0×, $1,1e-6$). RadianceKit avisa con imágenes muy grandes y ajusta automáticamente — 3 MP de resolución objetivo. Con imágenes de entrada 4K extremas, 0,5 o incluso 0,25 tendría sentido; en otro caso, cualquier Mac correrá solo en compactación CPU.

EN POCAS PALABRAS

Lo grandes que son las imágenes durante el entrenamiento. 1,0 = original, 0,5 = la mitad. Mitad = cuatro veces más rápido pero faltan los detalles más finos. Los ajustes preestablecidos eligen el valor correcto; con imágenes de entrada extremadamente grandes (más de 12 megapíxeles) la app baja automáticamente.

T23 resolutionWarmupScale **DETALLES**

Predeterminado: 0,0 (= deshabilitado) **Rango:** 0 o 0,1 – **Definido en:**

 **TÉCNICO**

Optimización V133: entrena la fase de densificación (iter 0 a T_2) a una resolución más baja que la fase de refinamiento. V308 lo apagó de nuevo para `.full` porque con $T_{22} = 1,0$ y cosine annealing, el ahorro de tiempo era marginal y la calidad sufría ligeramente. Se queda en el catálogo de campos porque podría volver a ser útil con entradas 4K y runs largos — el currículum Q6 (T80) recogió una lógica similar, aunque ahí está atada al schedule de LR. Si está habilitado y T_{80} `curriculumResolutionRamp` también es `true`, Q6 gana y sobrescribe este valor.

 **EN POCAS PALABRAS**

Función especial: en la primera mitad del entrenamiento aprende de imágenes más pequeñas, en la segunda de grandes. Ahorra tiempo. Off porque la variante Q6 más nueva lo resuelve mejor.

T24 tileSize **DETALLES**

Predeterminado: 16 **Rango:** 8, 16, 32 **Definido en:**

 **TÉCNICO**

Tamaño de los tiles de rasterización en píxeles. El renderizado Gaussian Splatting es basado en tiles: la imagen se divide en tiles de 16×16 píxeles, cada tile recopila los Gaussianos relevantes para él, los ordena por profundidad y los mezcla. 16 es el estándar que usan prácticamente todas las implementaciones 3DGS y está codificado en duro en los kernels Metal de RadiancKit; cambiar este valor requeriría recompilación del shader y no es efectivo en el estado actual. Se queda como campo por si una versión futura del engine soporta tamaño de tile dinámico.

 **EN POCAS PALABRAS**

Parámetro de render interno. Predeterminado 16, no cambiar.

T25 `throttleDelayMs` **DETALLES**

Predeterminado: 0 (initializer, `.full`, MCMC, Scene-Class), 0 (`.preview`) **Rango:** 0 – 100 **Definido en:**

 **TÉCNICO**

Retardo artificial entre iteraciones de entrenamiento en milisegundos. 0 = a toda velocidad (predeterminado). Valores más altos hacen el Mac más «utilizable» durante el entrenamiento, al dar a GPU/CPU pausas regulares — la responsividad de otras apps mejora, pero el tiempo de entrenamiento crece linealmente con el retardo. Valores típicos: 1–2 ms («throttling ligero», +5 % de tiempo de entrenamiento, el Mac se siente más responsivo), 5 ms («medio», +15 %), 10+ ms («eco», potencialmente el doble de tiempo de entrenamiento). Se ofrece en el Inspector bajo «Performance» pero no en la vista predeterminada — véase el backlog `dev_ux-backlog.md` que sugiere quitarlo de la Vista Experto porque, malinterpretado, extiende dramáticamente el tiempo de entrenamiento.

 **EN POCAS PALABRAS**

Cuántos milisegundos de pausa hace la app entre pasos de entrenamiento. 0 = sin pausa, tan rápido como sea posible. Valores más altos hacen el Mac más utilizable durante el entrenamiento — pero el entrenamiento también dura más. En un M3 Ultra o Mac Studio puedes dejar esto en 0; en un MacBook Air 2 o 5 sería un buen valor.

Diagnóstico y preparación de la nube de puntos (T26–T30)

T26 depthDistortionWeight

DETALLES

Predeterminado: 0,0 (= deshabilitado) **Rango:** 0 o 0,0001 – 0,05 **Definido en:**

TÉCNICO

V366 experimental: peso de un loss de regularización de distorsión de profundidad. Penaliza Gaussianos que, a lo largo de un rayo de render, están apilados en profundidad pero conceptualmente pertenecen a la misma superficie — esto anima a distribuciones de profundidad concentradas y reduce los floaters. Pruebas: 0,01 → 4,5 % peor, 0,001 → 8,1 % peor. La ventaja teórica — mejorar la consistencia multi-vista — no se ve en el loss L1, porque la hipótesis asume implícitamente que la geometría de SfM es correcta y los Gaussianos solo necesitan «apilarse». En la práctica, la nube de puntos SfM suele ser el componente más débil, no el apilado. Se mantiene disponible para datasets multi-vista con poses particularmente limpias (sintéticos, Mip-NeRF 360 con ground truth).

EN POCAS PALABRAS

Función experimental para evitar múltiples splats alineados en el mismo sitio. No habilitado porque las pruebas no mostraron beneficio.

T27 `singleViewOverfit` **DETALLES**

Predeterminado: false **Rango:** booleano **Definido en:**

 **TÉCNICO**

Flag diagnóstico: cuando es true, cada iteración de entrenamiento debe usar el índice de cámara 0 en lugar de uno aleatorio del pool de cámaras. Razón: si el modelo no es capaz ni siquiera de hacer overfit a una sola vista (es decir, el loss sobre la vista 0 no va a cero ni siquiera tras 10 000 iteraciones), hay un bug fundamental en el forward/backward pass. Este switch se usó intensivamente durante el desarrollo de los shaders Metal y los kernels del rasterizador diferenciable — fase V42–V47. Hoy solo disponible como chequeo de sanidad si alguien ha modificado código del backend y quiere hacer un test de regresión. Vía CLI con `--single-view`.

 **EN POCAS PALABRAS**

Modo de prueba para desarrolladores. Pueden usarlo para comprobar si la app puede siquiera aprender de UNA sola imagen. Irrelevante para usuarios normales, déjalo siempre off.

T28 `maxCameras` **DETALLES**

Predeterminado: 0 (= «usar todas las cámaras») **Rango:** 0 o 1 – N **Definido en:**

 **TÉCNICO**

Límite diagnóstico de V43: entrenar solo con las primeras N cámaras, ignorar todas las demás. Razón original: probar la hipótesis de que demasiadas cámaras crean conflictos de gradiente (demasiadas señales de loss conflictivas para el mismo Gaussiano). Resultado de prueba: sin ventaja sistemática del limitado artificial — más frames aportan prácticamente siempre más calidad. Se queda como flag CLI (`--max-cameras N`) para experimentos dirigidos, p. ej., «¿funciona el entrenamiento en las primeras 100 imágenes de un vuelo de dron de 1 500?». No expuesto en la UI.

 **EN POCAS PALABRAS**

Campo diagnóstico para desarrolladores — usa solo las primeras N imágenes, ignora el resto. Usuario normal no necesita esto, valor en 0 = todas las imágenes. Más imágenes = mejor resultado (véase `feedback_more-frames-better.md`).

T29 maxInitialPoints **DETALLES**

Predeterminado: 0 (= «usar todos los puntos SfM») **Rango:** 0 o 1 000 – 200 000+ **Definido en:**

 **TÉCNICO**

Red de seguridad V54: limita el número de puntos iniciales SfM con los que comienza el entrenamiento. Las reconstrucciones COLMAP densas pueden producir > 60 000 puntos, lo que con escalas iniciales grandes lleva a un solapamiento de 200–300 Gaussianos por píxel — esto crea un «campo de niebla» en el que el entrenamiento no converge. El submuestreo a ~16 000 puntos (lógica de hard-cap en el motor de entrenamiento) trae la densidad inicial al nivel usado por el 3DGS de referencia, y reduce dramáticamente el solapamiento. Se fija automáticamente con SfMs muy densos; vía CLI con `--max-points N`.

 **EN POCAS PALABRAS**

Cuántos puntos de inicio de la reconstrucción de cámara se usan. Con reconstrucciones muy densas (más de 60 000) la app limita automáticamente a 16 000 — en otro caso hay demasiada niebla al principio. No tienes que fijar esto; la app se encarga.

T30 cameraClusterOutlierMultiplier **DETALLES**

Predeterminado: 10,0 (todos los ajustes preestablecidos — nunca sobrescrito) **Rango:** 1,0 – 100,0
Definido en:

 **TÉCNICO**

Multiplicador para el filtro de outliers del cluster de cámaras, introducido en la Fase 3.10 A.1. Antes del entrenamiento, el motor de entrenamiento calcula el centroide de todas las posiciones de cámara y la distancia máxima de cualquier cámara al centroide. Los puntos SfM cuya distancia al centroide supera $\text{multiplier} \times \text{maxCameraDistance}$ se descartan como outliers. El default 10× preserva el comportamiento previo a la Fase 3.10. Un bug sutil: SfM más apretado (cámaras más juntas) → más pequeña → umbral más pequeño → más puntos se descartan como outliers. SfM más holgado → umbral más grande → menos puntos descartados. Esta es una de las causas de la anti-correlación funnel-vs-training de la Fase 3.9: un mejor SfM puede llevar aguas abajo a peor entrenamiento porque se matan demasiados puntos iniciales. El campo está disponible como override CLI (`--camera-cluster-outlier-multiplier`) para los sweeps A.3; no expuesto en la UI. Los valores por debajo de 5 suelen ser demasiado restrictivos, por encima de 20 inefectivos.

 **EN POCAS PALABRAS**

Filtro especial que descarta puntos de la reconstrucción que están lejos de la nube de cámaras. 10 = la app es generosa, mantiene casi todo. Aumentar puede tener sentido cuando puntos lejanos (montañas a lo lejos) parecen manchas flotantes en la imagen. Bajar solo en emergencias — perderás detalle a la distancia.

Regularización (T31–T37)

T31 coarseToFineBlurRadius

DETALLES

Predeterminado: 0 (= deshabilitado) **Rango:** 0 o 1 – 10 **Definido en:**

TÉCNICO

V369 experimental: radio de box blur aplicado al inicio de la fase de densificación a la imagen ground truth, reducido linealmente a 0 al final de la densificación (T2). Hipótesis: el entrenamiento coarse-to-fine — primero aprender estructuras gruesas, después detalles — debería dar geometría más estable. Pruebas: $r=3 \rightarrow 9,6\%$ peor, $r=1 \rightarrow 5,1\%$ peor. Razón del fallo: la densificación decide en base a gradientes en dominio de imagen, y el blur reduce exactamente las señales importantes para «hay que clonar aquí». Se queda en el catálogo de campos para pruebas futuras con un esquema distinto de density control.

EN POCAS PALABRAS

Modo experimental «primero grueso, después detalle». No aportó nada, sigue off.

T32 scaleRegWeight

DETALLES

Predeterminado: 0,0 (= deshabilitado) **Rango:** 0 o 0,0001 – 0,05 **Definido en:**

TÉCNICO

V370 experimental: regularización L1 sobre escala en espacio mundo. Penaliza Gaussianos que crecen demasiado — evita «mega splats» que cubren superficies enteras de pared con un solo Gaussiano. Pruebas: 0,01 $\rightarrow 200\%$ peor loss (2 M Gaussianos, explosión total), 0,001 $\rightarrow 214\%$ peor. Razón: la regularización de escala entra en conflicto con la density control — escalas más pequeñas significa que se necesitan más Gaussianos, así que la density control divide más a menudo, lo que a su vez significa más trabajo de gradiente. Deshabilitado, pero documentado para experimentos Mip-Splatting (T74): en ese contexto un límite inferior de escala podría tener sentido.

EN POCAS PALABRAS

Regularización que fuerza a los splats a quedarse pequeños. Disparó explosiones de splats en pruebas (millones de splats). No habilitar.

T33 anisotropyRegWeight **DETALLES**

Predeterminado: 0,0 (= deshabilitado) **Rango:** 0 o 0,0001 – 0,05 **Definido en:**

 **TÉCNICO**

V445 experimental: penalización sobre el cociente $\max(\text{scale})/\min(\text{scale})$, pensada para evitar Gaussianos «aguja» extremadamente alargados que se perciben como floaters. Pruebas: 0,01 → 69 % peor, 0,001 → 15 % peor. Razón: la regularización fuerza a los splats a forma «redonda», lo que sobre una superficie plana (pared, mesa, suelo) es exactamente incorrecto — ahí un Gaussiano plano y ancho es más eficiente que uno esférico. Deshabilitado. V549f ofreció con T34 `scaleRatioPruneThreshold` un enfoque alternativo más dirigido, que también se revirtió.

 **EN POCAS PALABRAS**

Regularización que penaliza splats muy largos y delgados. Suena sensato pero en pruebas fue peor. Off.

T34 scaleRatioPruneThreshold **DETALLES**

Predeterminado: 0,0 (= deshabilitado) **Rango:** 0 o 5,0 – 100,0 (típicamente 10,0 – 30,0) **Definido en:**

 **TÉCNICO**

Pruning post-training experimental que elimina cada Gaussiano cuyo ratio $\max(\text{scale})/\min(\text{scale})$ supera el umbral lineal aquí fijado. Apunta a floaters «aguja/disco» extremadamente alargados que no se pueden eliminar solo con regularización. En pruebas, el pruning eliminó floaters como se esperaba, pero también splats planos útiles en paredes y suelos — la imagen se volvió más agujereada. De ahí off por defecto; el flag CLI (`--scale-ratio-prune N`) sigue para experimentos dirigidos. Valores recomendados si quieres probar: 30 (muy conservador, elimina solo outliers extremos), 10 (agresivo, cuesta detalle).

 **EN POCAS PALABRAS**

Intento de filtrar splats muy alargados tras el entrenamiento. Fue net-negativo en pruebas — floaters eliminados pero también detalle. Off.

T35 **opacityRegWeight** **DETALLES**

Predeterminado: 0,0 (= deshabilitado) **Rango:** 0 o 0,0001 – 0,05 **Definido en:**

 **TÉCNICO**

V446 experimental: penalización de cross-entropy binaria que tira de la opacidad hacia 0 o 1 (es decir, lejos de «semi-transparente»). Hipótesis: una distribución de opacidad más nítida mejoraría la claridad de imagen. Probado combinado con T33 → la regularización cuesta calidad, ambos deshabilitados. Deshabilitado. Atención: en la beta 1.4.3 apareció un bug que tenía exactamente este campo con un valor default cambiado (initializer = 0,01), lo que llevó a la extinción masiva del número de Gaussianos (460 K → 5 en una iteración). Desde 1.4.4 fijado en duro a 0,0 como default.

 **EN POCAS PALABRAS**

Regularización que hace los splats o totalmente transparentes o totalmente sólidos. No aporta nada, puede incluso volverse peligroso (extinción masiva por bug de 1.4.3). Déjalo en 0.

T36 **opacityDecayFactor** **DETALLES**

Predeterminado: 0,0 (initializer = deshabilitado), 0,9995 (`.full` , `.classicBalanced` — estándar HTGS) **Rango:** 0 (off) o 0,95 – 1,0 **Definido en:**

 **TÉCNICO**

Implementación V546 del esquema HTGS (Hierarchical Time-Gating, Eurographics 2025): cada T37 `opacityDecayInterval` iteraciones la opacidad sigmoide de cada Gaussiano se multiplica por este factor. $0,9995 \times 100$ aplicaciones dan ~95 % residual por fase de densificación — una presión a la baja ligera pero constante sobre todas las opacidades, que deja caer de forma fiable a los Gaussianos que aportan poco por debajo del T14 `pruneOpacityThreshold`. El resultado: 14 % mejor loss L1 en Horse Full (3-trial avg V546) vs V438 sin decay. Activo solo durante la fase de densificación (hasta T2), después el entrenamiento sigue sin decay, así que las opacidades establecidas durante refinamiento se mantienen estables. No se usa bajo MCMC (MCMC tiene sus propios mecanismos vía T67 `mcmcRelocationInterval` + T68 `mcmcDeadOpacityThreshold`).

 **EN POCAS PALABRAS**

Un «fade suave» de todos los splats a lo largo del tiempo de entrenamiento. Hace transparentes más rápido a los splats inactivos, así desaparecen durante la limpieza. Fue la palanca de calidad más importante de la actualización V546: 14 % mejor. Incorporado al ajuste preestablecido Quality. No se recomienda ni siquiera trastear porque está balanceado con precisión.

T37 **opacityDecayInterval** **DETALLES**

Predeterminado: 50 **Rango:** 10 – 500 **Definido en:**

 **TÉCNICO**

Intervalo de iteraciones al que se aplica T36 `opacityDecayFactor`. Default del paper HTGS 50, dejado así en `.full`. Intervalos largos (>200) cancelan parcialmente el efecto porque entre dos aplicaciones ocurren suficientes actualizaciones de gradiente como para que la opacidad vuelva a subir. Intervalos más cortos (<20) hacen el decay demasiado agresivo. Activo solo en la fase de densificación.

 **EN POCAS PALABRAS**

Cada cuánto se aplica el «fade». 50 = cada 50 iteraciones un pequeño paso de fade. Encaja.

Refinamiento (T38–T44)

T38 gradientAccumulationSteps

DETALLES

Predeterminado: 1 (= «una vista por paso Adam»)

Rango: 1 – 8 **Definido en:**

TÉCNICO

Función V424: número de vistas cuyos gradientes se acumulan antes de que corra una actualización de Adam. Con `> 1` la app corre por una vía backward project «unfused» separada que suma los gradientes a un buffer separado; la aplicación final escala por $1/N$ para mantener la magnitud constante. V424 probó 2-vistas → neutro para calidad pero 10 % más lento (porque unfused es más caro que fused). Revertido para `.full` pero usado deliberadamente para MCMC — `.fullMCMC` corre con, pero los tests V544a mostraron que con el gap de calidad respecto a Classic se encoge a 5 % (en lugar de 11 %). En el default initializer 1, en el preset actual 1, queda como flag CLI (`--accum-steps N`).

EN POCAS PALABRAS

Cuántas imágenes considera la app antes de ajustar los splats. 1 = cada imagen por separado. Más alto = mira varias imágenes a la vez y después aplica una media. No aporta nada en el caso estándar; bajo MCMC, 2 puede ayudar un poco.

T39 testViewIndices **DETALLES**

Predeterminado: `[]` (= vacío, todas las vistas se usan para entrenamiento) **Rango:** `Set<Int>`, subconjunto arbitrario de índices de cámara **Definido en:**

 **TÉCNICO**

Función V546: conjunto de índices de cámara que NO se usan para entrenamiento sino que se guardan como holdout para evaluación PSNR/SSIM/LPIPS. Se fija automáticamente cuando el flag CLI `--benchmark` está activo: entonces cada 8ª vista empezando por el índice 0 (estándar LLFF, idéntico a las convenciones Mip-NeRF 360 y del paper 3DGS). Sin benchmark vacío — el entrenamiento usa todas las vistas. **Atención:** fijar manualmente este campo sin entender los índices puede inutilizar el benchmark (p. ej., cuando todos los índices están por encima de N mientras hay solo N-50 vistas → sin holdouts → sin evaluación). Cuando exportas tu propio ajuste preestablecido, testViewIndices no se persiste porque es dependiente de la escena y en otro caso dejaría valores sin sentido entre datasets distintos.

 **EN POCAS PALABRAS**

Qué imágenes durante el entrenamiento se «dejan fuera» para usarlas después para medición de calidad. No fijas esto tú mismo; el flag `--benchmark` lo hace automáticamente (cada octava imagen es test). Si fijas tus propios índices: peligroso, puede corromper el benchmark.

T40 refinementPruneInterval **DETALLES**

Predeterminado: 0 (= deshabilitado) **Rango:** 0 o 100 – 5 000 **Definido en:**

 **TÉCNICO**

Función V425: cada N iteraciones durante la fase de refinamiento (tras T2) corre una pasada de prune adicional que elimina Gaussianos con sigmoid(opacity) < T41 refinementPruneOpacityThreshold. Razón: durante la densificación hay llamadas regulares de density control, después ya no — pero los Gaussianos cuya opacidad sigue cayendo se quedan en el buffer. V425 probado y revertido: el pruning adicional correlacionaba con V426 (Densificación de Dos Fases, que también acabó en un cascade failure de 0 Gaussianos). Deshabilitado. Flag CLI disponible para experimentos; si está habilitado, 1 000 o 2 000 son valores sensatos.

 **EN POCAS PALABRAS**

Limpieza adicional durante la fase de refinamiento. No aporta nada, sigue off.

T41 refinementPruneOpacityThreshold **DETALLES**

Predeterminado: 0,0 (= «usar T14 ») **Rango:** 0 o 0,001 – 0,1 **Definido en:**

 **TÉCNICO**

V425b: umbral de opacidad separado para el pruning de refinamiento. Tras la densificación, la mayoría de los Gaussianos han alcanzado una opacidad claramente más alta (> 0,001), por lo que el default T14 pruneOpacityThreshold sería demasiado laxo. Si T40 está activo, este campo determina su propio umbral. En 0,0 se usa T14 como antes. Solo relevante si T40 > 0.

 **EN POCAS PALABRAS**

Umbral para la limpieza adicional de refinamiento (véase T40). Ambos campos inactivos, así que irrelevante.

T42 midTrainingCompactificationIterations**DETALLES**

Predeterminado: `[]` (= deshabilitado) **Rango:** `[Int]`, valores en (`densifyUntilIteration`, `maxIterations`) **Definido en:**

TÉCNICO

Función V549: puntos de iteración explícitos durante la fase de refinamiento en los que corre una pasada de compactificación (elimina sigmoid(opacity) < 0,01 + Gaussianos de escala outlier, misma lógica que T56 `postTrainingCompactification`). Razón: las fases de refinamiento largas pueden mostrar acumulación de confeti/ floaters, cuyo SH luego sobreajusta a artefactos específicos de vista. Configuración típica si está habilitado: `[10000, 20000, 30000]` para 40K Classic. **PERO:** los tests A/B V549 sobre el dataset Family mostraron peor L1 en todas las configuraciones: `[10K,20K,30K]@0.01` → -48 % count pero +36 % L1; `[20K,30K]@0.005` → -44 % count pero +45 % L1; `[20K,30K]@0.001` → -17 % count pero +87 % L1. De ahí deshabilitado. Flag CLI `--mid-compact "10000,20000"` disponible, si prefieres el trade-off visual de floaters (menos confeti en la vista) sobre la regresión de loss.

EN POCAS PALABRAS

Acciones de limpieza a mitad del entrenamiento. En pruebas, la limpieza empeoró el resultado final (sí, menos floaters, pero también menos detalle). Off, se puede habilitar vía CLI, por si los floaters te molestan más que una imagen un poco más blanda.

T43 frustumCullEnabled **DETALLES**

Predeterminado: false **Rango:** booleano **Definido en:**

 **TÉCNICO**

Función V549b: tras el entrenamiento, todos los Gaussianos fuera de la unión de todos los frustums de cámara de entrenamiento se eliminan. Esos Gaussianos nunca estuvieron restringidos por la señal de loss y son siempre floaters. Particularmente efectivo para escenas en las que la novel view está detrás o al lado del recorrido de la cámara (p. ej., detrás de un vuelo de dron lineal) — los floaters allí nunca son visibles durante el entrenamiento pero sí más tarde al moverse en el visor 3D. A/B V549b en vuelos de dron mostró resultados positivos, de ahí disponible como opt-in. Default false porque para capturas de objeto con cobertura orbital completa la unión de frustums abarca toda la escena y la función no elimina nada — se ofrece en Settings bajo «Floater Reduction» y también se prueba implícitamente en el preset Q9 Outdoor vía T44 `frustumCullExpansion` (Q7 BayesOpt no lo habilitó, porque la cúpula de cielo outdoor resuelve mejor el mismo problema).

 **EN POCAS PALABRAS**

Filtro especial para vuelos de dron o capturas lineales: tras el entrenamiento, los splats que no fueron «vistos» por ninguna cámara se eliminan. Opcional, activable en Ajustes. Para capturas simples de objeto, innecesario.

T44 frustumCullExpansion **DETALLES**

Predeterminado: 1,1 **Rango:** 1,0 – 2,0 **Definido en:**

 **TÉCNICO**

Margen NDC para T43 `frustumCullEnabled`. 1,0 recortaría exactamente en el borde de imagen, lo que recortaría splats tambaleantes cerca del borde de manera demasiado agresiva. 1,1 = 10 % de padding más allá del encuadre exacto de la cámara — da algo de tolerancia para píxeles de borde que aún podrían ser visibles en una novel view ligeramente desplazada. Valores > 1,2 hacen el cull prácticamente inefectivo porque el frustum expandido abarca demasiado espacio.

 **EN POCAS PALABRAS**

Lo estrictamente que recorta el filtro mencionado arriba. 1,1 = un poco de margen de seguridad respecto al borde de imagen. Deja el valor.

Sky Dome (T45–T48)

T45 skyDomeEnabled

DETALLES

Predeterminado: false (initializer + todos los ajustes preestablecidos excepto P9 Outdoor) **Rango:** booleano **Definido en:**

TÉCNICO

Función V549e: antes de iniciar el entrenamiento se genera una nube de puntos esférica (esfera Fibonacci con T46 puntos de muestra), se coloca a un radio de T47 $\text{skyDomeRadiusMultiplier} \times \text{scene_extent}$ alrededor del centro de la escena y se inicializa con los colores de los píxeles enmascarados como cielo de todas las cámaras de entrenamiento (véase T20 `skyMaskingEnabled`). Estos Gaussianos de sky dome se insertan al inicio del buffer Gaussiano y durante el entrenamiento están «congelados» (gradientes position/scale/rotation = 0, solo SH y opacidad permanecen optimizables). Efecto: en lugar de áreas negras de «confeti» a la distancia, el usuario ve un cielo real en novel views. El MVP V549e funciona muy bien en escenas de dron y paisaje; en el preset P9 Outdoor activado por defecto. Déjalo off para escenas interiores — la esfera colgaría inútil fuera de la habitación.

EN POCAS PALABRAS

Activa una «cúpula de cielo» artificial alrededor de la escena. Hace las capturas exteriores mucho más bonitas: en lugar de manchas negras en el borde de imagen, la app muestra el cielo real. Obligatorio para vuelos de dron y paisajes, inútil para interiores.

T46 skyDomeSampleCount

DETALLES

Predeterminado: 5 000 **Rango:** 1 000 – 50 000 (típico 2 000 – 10 000) **Definido en:**

TÉCNICO

Número de puntos de muestra esfera Fibonacci sobre la esfera de la cúpula de cielo. Valores más altos → cúpula más densa (mejor a resoluciones grandes y mucho cielo visible), pero mayor consumo de memoria. 5 000 es el sweet spot para renderizados 4K; a resoluciones más bajas, 2 000–3 000 bastan. Los puntos se inicializan por distancia coseno a cada vector de vista de cámara de entrenamiento con los correspondientes píxeles enmascarados como cielo — los puntos de muestra cuyo cono de vista no es visto por ninguna cámara mantienen un valor inicial de opacidad bajo, pero permanecen sin cambios durante el entrenamiento (congelados).

EN POCAS PALABRAS

Lo denso que es el cielo artificial. 5 000 puntos suelen bastar. Más = transición más suave desde la distancia, pero cuesta algo de memoria.

T47 skyDomeRadiusMultiplier

DETALLES

Predeterminado: 30,0 (initializer + la mayoría de ajustes preestablecidos), 59,0 (P9 Outdoor, óptimo Q7 BayesOpt) **Rango:** 5,0 – 200,0 **Definido en:**

TÉCNICO

Radio de la esfera de la cúpula de cielo relativo a la extensión de la escena (= distancia media entre posiciones de cámara). 30 = la esfera tiene 30× el diámetro de la nube de cámaras. Demasiado pequeña (< 5) → la cúpula interfiere con la propia escena (p. ej., un splat de cúpula aterriza en primer plano); demasiado grande (> 100) → pérdida de precisión float32 en las posiciones de cúpula, lo que dispara fallos de render a la distancia. Q7 BayesOpt en Bicycle (Mip-NeRF 360) encontró 59,0 como óptimo específico de escena para outdoor — esto sugiere que el default 30,0 es demasiado pequeño para paisajes profundos y que los píxeles de cúpula renderizan visiblemente como una «pared» en regiones de borde de imagen.

EN POCAS PALABRAS

Lo lejos que debe estar la cúpula de cielo artificial. 30 = bastante lejos. Con grandes paisajes, 50–60 es mejor (el preset Outdoor lo hace automáticamente). Demasiado pequeño sería como tener manchas justo delante de la lente.

T48 frozenGaussianCount **DETALLES****Predeterminado:** 0 (= sin Gaussianos congelados)**Rango:** 0 o 1 – **T46** **Definido en:** **TÉCNICO**

Número de Gaussianos al inicio del buffer cuyos gradientes position/scale/rotation se ponen a cero en el optimizador — permanecen espacialmente rígidos durante todo el entrenamiento. La density control no puede clonarlos, dividirlos ni podarlos. Usado para la inyección de cúpula de cielo (véase **T45**): cuando la cúpula está on, este campo se fija automáticamente a **T46** skyDomeSampleCount. La fijación manual es posible (p. ej., para congelar una nube de puntos pre-colocada desde un escaneo LiDAR), pero no es directamente accesible en la UI. Importante: los primeros N Gaussianos en el buffer son siempre los congelados — el orden en el buffer decide, no un índice explícito.

 **EN POCAS PALABRAS**

Cuántos splats al principio están fijos y no pueden moverse. Se fija automáticamente al número de la cúpula de cielo cuando está activa. No necesitas tocarlo tú mismo.

Adam + schedule de LR (T49–T55)**T49** adamResetIteration **DETALLES****Predeterminado:** 0 (= deshabilitado) **Rango:** 0 o 100 – **Definido en:** **TÉCNICO**

Función V430: iteración en la que los acumuladores de momentum (m1, m2) del optimizador Adam se resetean a cero. La corrección de sesgo posterior corre con (iter - adamResetIteration) en lugar de iter. V430 probó reset en 5 000 (tras el fin de densificación) → loss 12,8 % peor. Razón: el momentum de Adam que se acumuló durante la densificación lleva información sobre las magnitudes de gradiente típicas y acelera la fase de refinamiento. Tirarlo cuesta las primeras ~500 iteraciones de refinamiento en convergencia. Deshabilitado. Se queda como flag CLI para experimentos de investigación.

 **EN POCAS PALABRAS**

Botón de reset para la «memoria» interna del optimizador Adam. Hizo daño en pruebas, si-gue off.

T50 positionLRScheduleEndIteration DETALLES

Predeterminado: 0 (initializer = «usar maxIterations»), 20 000 (`.full` — cosine termina en 20K aunque `maxIter=35K`), 30 000 (`.fullClassicPaper`) **Rango:** 0 o 1 000 – **Definido en:**

 TÉCNICO

Función V431: iteración en la que la curva cosine annealing para la LR de posición alcanza su mínimo. Si 0, es idéntico a `T1 maxIterations`. Si > 0 , el schedule corre hasta este valor y se mantiene constante en `T4`

`positionLearningRateFinal` después. Esto permite una «fase de refinamiento extendida» con learning rate mínima pero constante — refina posiciones despacio sin un nuevo decay. `.full` lo hace (el schedule termina en 20K, el entrenamiento corre hasta 35K), V434c/V434d confirmaron: 15K y 25K ambos por igual, 20K marginalmente óptimo. Usado junto con `T51` para modificar también las LRs no-posición en la fase extendida.

 EN POCAS PALABRAS

Cuándo la app deja de bajar más la learning rate de posición. Si es menor que la iteración máxima, el resto corre con una mini-tasa constante — esto refina muy despacio pero muy estable. Incorporado en el ajuste preestablecido Quality, no necesitas trastear.

T51 extendedPhaseLRDecay DETALLES

Predeterminado: 0,0 (= deshabilitado, LRs constantes) **Rango:** 0 o 0,01 – 1,0 **Definido en:**

 TÉCNICO

Función V433: multiplicador mínimo para las LRs no-posición (escala, rotación, opacidad, SH) en la «fase extendida» — es decir, tras alcanzar `T50` y la LR de posición ya en `T4`. Si 0,1, scale/rotation/opacity/SH se decaen ellas mismas en cosine de 1,0 (= su LR estándar) a 0,1× de su estándar. Si 0,0 (default), permanecen constantes. V457 probó decay completo (0,0, es decir, decay a cero) contra sin decay y encontró: avg 0,0400 (2 runs), el mismo loss que V438 sin decay. Comportamiento más limpio con decay pero no medible mejor. De ahí deshabilitado. Se queda en la CLI como `--nonpos-lr-scale F`.

 EN POCAS PALABRAS

En la fase final de refinamiento, hacer también más pequeñas las learning rates de color y forma. Hace el entrenamiento «más estable» pero empíricamente no mejor. Off.

T52 adaptiveDensifyThreshold DETALLES

Predeterminado: false **Rango:** booleano **Definido en:**

 TÉCNICO

V440 experimental: cuando es true, la app calcula en cada paso de densificación el p98 de la distribución actual de gradientes y lo usa como umbral dinámico (clamped a al menos 0,5× del valor configurado de T11 para que no derive demasiado). Hipótesis: la adaptación automática a la fase actual de la escena haría la density control más robusta — p. ej., poda más estricta al principio, más laxa después, o viceversa. V440 probado y revertido: caída catastrófica a 63 K Gaussianos (pruning masivo, porque el p98 en las primeras iteraciones es extremadamente alto y después casi nada supera el umbral). El umbral fijo está ya bien calibrado, el ajuste dinámico hace más daño que bien. Q5 (T77) ofrece una lógica adaptativa alternativa vía mediana rodante que evita el problema.

 EN POCAS PALABRAS

Versión adaptativa del umbral de densificación. En pruebas catastrófico (el número de splats se desplomó a 63K). Off. Q5 tiene una variante mejor.

T53 mergeAfterDensification DETALLES

Predeterminado: false (initializer), true (.full , .classicBalanced , .fullClassicPaper) **Rango:** booleano **Definido en:**

 TÉCNICO

Función V438: al final de la fase de densificación (iter T2) corre una pasada de merge única que combina Gaussianos cercanos con escala y color similares. Reduce el número de Gaussianos típicamente un 5–15 % sin pérdida visible de calidad. Razón: tras el intenso clonado, surgen clusters de Gaussianos casi idénticos que no aportan nada nuevo — mergearlos libera capacidad de optimizador para otras áreas. Por defecto en los ajustes preestablecidos Classic Quality. No usado bajo MCMC, porque MCMC mediante su lógica de relocalización no deja que esos clusters se formen en primer lugar.

 EN POCAS PALABRAS

Al final de la fase de duplicación de splats, combinar clones que son casi idénticos. Reduce el tamaño de datos sin efecto visible. Por defecto on en el preset Quality.

T54 densifyPhase2FromIteration **DETALLES**

Predeterminado: 0 (= deshabilitado) **Rango:** 0 o

T2 – T1 **Definido en:**

 **TÉCNICO**

V426 experimental: habilita una segunda fase de densificación que comienza tras la pausa de refinamiento en esta iteración y corre hasta T55. Hipótesis: tras una fase de refinamiento los acumuladores de gradiente tienen magnitudes más estables y pueden indicar con más precisión qué regiones aún necesitan Gaussianos adicionales. V426 probado y revertido: la densificación de dos fases cayó en cascade failure de 0 Gaussianos (combinada con el pruning de refinamiento V425 destruyó el buffer). Deshabilitado. Flag CLI disponible para experimentos.

 **EN POCAS PALABRAS**

Segunda ronda de multiplicación tras una pausa. En pruebas, aniquiló el inventario de splats. Off.

T55 densifyPhase2Untilliteration **DETALLES**

Predeterminado: 0 **Rango:** 0 o T54 – T1 **Definido en:**

 **TÉCNICO**

Fin de la densificación de dos fases V426. Solo relevante cuando T54 > 0. Ambos campos juntos deshabilitados.

 **EN POCAS PALABRAS**

Fin de la segunda ronda de multiplicación (véase T54). Ambos off.

Post-procesado + Apple AI (T56–T60)

T56 postTrainingCompactification

DETALLES

Predeterminado: true (en todos los ajustes pre-establecidos de producción), false (`.quickTest` , `.preview`) **Rango:** booleano **Definido en:**

TÉCNICO

Función V443: tras terminar el entrenamiento, los Gaussianos con `sigmoid(opacity) < 0,01` se eliminan en duro (prácticamente ya no contribuyen a la imagen). Reduce el número de Gaussianos típicamente un 58 % y el tamaño del archivo de exportación un 55 % sin pérdida visible de calidad. On por defecto en ajustes preestablecidos de producción — el resultado final debería entregarse lo más compacto posible. Off en `.quickTest` , porque un run diagnóstico no se exporta de todos modos. A diferencia de T42 `midTrainingCompactificationIterations` (V549), la compactificación solo ocurre al final — el refinamiento puede usar todos los Gaussianos hasta entonces.

EN POCAS PALABRAS

Limpieza tras el entrenamiento: los splats casi invisibles se eliminan. Hace el archivo de exportación aproximadamente la mitad de grande sin pérdida de calidad. Función obligatoria, déjalo off solo en runs diagnósticos.

T57 metalFXUpscaling

DETALLES

Predeterminado: false **Rango:** booleano **Definido en:**

TÉCNICO

Función V444: habilita el Apple MetalFX Spatial Upscaler en lugar de interpolación bilineal en la salida del visor 3D. Cuando la resolución de entrenamiento < tamaño de la vista (p. ej., entrenando a 0,5x, visualización de vista a resolución completa), MetalFX puede entregar una imagen claramente más nítida. Cambia en vivo en la vista, no se requiere reentrenar. Mutuamente exclusivo con T58 `mpsLanczosScaling` — MetalFX tiene prioridad. Recomendación: habilitar cuando la imagen en el visor parece «deslavada» comparada con el detalle esperado.

EN POCAS PALABRAS

Afilado de imagen basado en ML de Apple en el visor 3D. Ayuda cuando has entrenado a una resolución más baja y quieres mostrar el resultado a pantalla completa. Conmutador en vivo, pruébalo.

T58 mpsLanczosScaling **DETALLES**

Predeterminado: false **Rango:** booleano **Definido en:**

 **TÉCNICO**

Función V444: MPSImageLanczosScale para el escalado de vista en lugar de interpolación bilineal. Lanczos es un método de resampling basado en sinc clásico que entrega resultados significativamente más nítidos que bilineal con overhead mínimo. Conmutador en vivo. Sobrescrito por **T57** cuando ambos están on.

 **EN POCAS PALABRAS**

Método de afilado clásico para el visor 3D (Lanczos). MetalFX (T57) es basado en ML y normalmente mejor; Lanczos es una alternativa menos agresiva.

T59 livePreviewInterval **DETALLES**

Predeterminado: 50 (initializer y la mayoría de ajustes preestablecidos) **Rango:** 0 (off) o 10 – 5 000 **Definido en:**

 **TÉCNICO**

Cada cuánto se refresca el visor 3D durante el entrenamiento con los Gaussianos actuales. 50 = cada 50 iteraciones un nuevo render en el visor — suficiente para observar el progreso sin ralentizar el entrenamiento. 0 = el visor no se actualiza en absoluto (entrenamiento en segundo plano, velocidad máxima). Ajuste típico: bajo `.quickTest` bajarlo a 10 (quieres ver cada paso), en runs MCMC largos subirlo a 500–2000 (el overhead de actualización suma a ratos).

 **EN POCAS PALABRAS**

Cada cuánto se refresca la previsualización 3D durante el entrenamiento. 50 = cada 50 iteraciones. Más alto = menos a menudo = un poco más rápido, pero ves el progreso con menos frecuencia. 0 = sin previsualización (para la velocidad máxima).

T60 perceptualLossWeight DETALLES

Predeterminado: 0,0 (= deshabilitado) **Rango:** 0 o 0,001 – 0,5 **Definido en:**

 TÉCNICO

Función futura V444: peso de un término de perceptual loss vía MPSGraph (red pequeña tipo VGG). Capturaría similitud estructural y textural a un nivel semántico más alto que L1+SSIM — típico en pipelines de investigación donde «píxel-perfecto» importa menos que «se ve realista». Implementación pendiente (stub de código presente, pero el forward pass no implementado). Predeterminado 0,0. Se queda en el catálogo de campos para activación futura; flag CLI `--percep-weight F` reservado.

 EN POCAS PALABRAS

Función planificada que con asistencia de IA apunta a «aspecto natural» en lugar de «preciso a nivel de píxel». Aún no totalmente implementado.

Densificación MCMC (T61–T73)

T61 densificationStrategy DETALLES

Predeterminado: `.classic` (initializer + ajustes Classic), `.mcmc` (todos los ajustes preestablecidos MCMC + Scene-Class) **Rango:** `.classic` o `.mcmc` **Definido en:**

 TÉCNICO

Elige entre densificación clásica (clone/split/prune, Kerbl et al. 2023) y densificación MCMC (Stochastic Gradient Langevin Dynamics con relocalización, Kheradmand et al. NeurIPS 2024). Con `.classic` se evalúan T11–T16, con `.mcmc` los T62–T73. Cuidado al cambiar: los defaults Classic y MCMC están calibrados de forma completamente distinta — quien gire el selector en la Vista Experto sin cargar un ajuste preestablecido coincidente arriesga una extinción masiva tipo bug-1.4.3 (460 K → 5 en una iteración, porque el opacity reg 0,01 de MCMC mata las opacidades Classic). De ahí que los defaults del initializer MCMC estén deliberadamente «soft-washed» (todos los valores reg 0,0).

 EN POCAS PALABRAS

Qué algoritmo se usa para multiplicar los splats. Classic = método original (rápido, muchos splats). MCMC = método más nuevo (más lento, muchos menos splats, pero más compacto). Los ajustes preestablecidos eligen el correcto. Solo cambia tú si también cargas el ajuste preestablecido coincidente (P5–P7 o P8–P10).

T62 mcmcMaxGaussians

DETALLES

Predeterminado: 150 000 (initializer + `.fullMCMC + .mcmcBalanced`), 100 000 (`.mcmcPreview`), 1 500 000 (`.fullMCMCMip` — variante Mip-Splatting con presupuesto 10x), 1,19 M (`.renderPreset`), 1,25 M (`.outdoorPreset`), 670 K (`.indoorPreset`)
Rango: 0 (= «usar capacidad de buffer») o 10 000 – 5 000 000 **Definido en:**

TÉCNICO

Cota superior dura para el número de Gaussianos bajo estrategia MCMC. El número crece gradualmente por `T70 mcmcGrowthRate` (típicamente 5 %) por paso de relocalización hasta este tope. V473/V531 encontraron 150 K como sweet spot — por encima de 200 K diluye la calidad del splat (demasiados Gaussianos redundantes pequeños), por debajo de 100 K deja la escena sub-densificada. Con escenas muy grandes (p. ej., vuelo de dron de 1 545 fotos con 158 K init SfM), 150 K es demasiado bajo — de ahí la extensión 1.4.5 `T72 mcmcCapMultiplier + T73 mcmcAutoScaleByScene`. Q7 BayesOpt encontró óptimos específicos de escena entre 670 K (Indoor) y 1,25 M (Outdoor). Con valor 0, el engine usa la capacidad completa del buffer como tope.

EN POCAS PALABRAS

Número máximo de splats bajo MCMC. 150 000 es el predeterminado y basta para la mayoría de las escenas. Los ajustes Outdoor y Render (P8, P9) llegan a 1+ millón para escenas más ricas en detalle. Subir puede aportar detalle, cuesta memoria; bajar es más bien un freno de emergencia.

T63 mcmcNoiseScale **DETALLES**

Predeterminado: 0,00005 (5e-5 = default del paper) **Rango:** 1e-6 – 1e-3 **Definido en:**

 **TÉCNICO**

Multiplicador para el ruido Gaussiano que en cada iteración MCMC se añade a la posición de cada Gaussiano (lógica SGLD). Más alto = más exploración (los Gaussianos vagan más, encontrando puntos potencialmente mejores), más bajo = más explotación (los Gaussianos se quedan donde ya están bien). V467 y V536 confirmaron 5e-5 como óptimo — 1e-5/2e-5 demasiado poca exploración, 1e-4 demasiada (los splats se difunden). Cosine-decayed sobre el tiempo de entrenamiento hasta T69 mcmcNoiseDecayEnd — al final del rango de decay, el ruido es efectivamente 0 y los Gaussianos convergen.

 **EN POCAS PALABRAS**

Cuánta «vibración» aleatoria permite la app a los splats para que encuentren ellos mismos el mejor sitio. El valor predeterminado es óptimo en pruebas. Si lo subes, los splats se vuelven inquietos.

T64 mcmcOpacityRegWeight **DETALLES**

Predeterminado: 0,0 (= deshabilitado en defaults de RadianceKit, paper: 0,01) **Rango:** 0 o 0,001 – 0,05 **Definido en:**

 **TÉCNICO**

Penalización L1 específica de MCMC sobre la opacidad. Default del paper 0,01 (empuja los Gaussianos no usados hacia cero, los hace disponibles para relocalización). V464b mostró sin embargo: sin reg es medible mejor en RadianceKit (Sesión 28 confirmado). Razón: el criterio de pruning definido con T68 mcmcDeadOpacityThreshold es suficiente por sí solo — una penalización L1 adicional también fuerza a Gaussianos valiosos de baja opacidad a morir. De ahí default 0. **Atención:** en el build beta 1.4.3 el default del initializer estaba erróneamente en 0,01, lo que resultó en el bug de extinción masiva (véase explicación T61); desde 1.4.4 fijado a 0,0.

 **EN POCAS PALABRAS**

Regularización especial de MCMC. Off porque el otro mecanismo de MCMC (umbral en T68) ya cubre esto. Déjalo en 0.

T65 `mcmcScaleRegWeight` **DETALLES****Predeterminado:** 0,0 (= deshabilitado, paper: 0,01)**Rango:** 0 o 0,001 – 0,05 **Definido en:** **TÉCNICO**

Penalización L1 específica de MCMC sobre los autovalores de escala. Default del paper 0,01. V464b: sin reg mejor, mismo razonamiento que T64. Deshabilitado en todos los ajustes preestablecidos MCMC de RadianceKit. Misma precaución que T64: bug 1.4.3.

 **EN POCAS PALABRAS**

Como T64 pero para el tamaño de splat. Off.

T66 `mcmcRelocationInterval` **DETALLES**

Predeterminado: 100 (initializer + todos los ajustes preestablecidos MCMC, estándar del paper), 155 (P9 Outdoor — óptimo Q7 BayesOpt) **Rango:** 50 – 500 **Definido en:**

 **TÉCNICO**

Intervalo de iteraciones en el que MCMC reubica los Gaussianos muertos ($\text{sigmoid}(\text{opacity}) < T68$ `mcmcDeadOpacityThreshold`) a nuevas posiciones. V537 probó 50 (demasiado disruptivo, el loss fluctúa) y 200 (marginalmente peor, MCMC pierde responsividad). 100 es óptimo. Q7 BayesOpt en Bicycle encontró 155 como óptimo específico de escena para outdoor — los intervalos ligeramente más largos dan a Adam más tiempo para integrar los Gaussianos recién colocados antes de que el siguiente evento reloc los ponga bajo presión.

 **EN POCAS PALABRAS**

Cuántas iteraciones entre que MCMC mueve splats muertos a otro sitio. 100 es estándar. No necesitas trastear tú mismo — el ajuste preestablecido Outdoor ya tiene el valor óptimo.

T67 **mcmcWarmupIterations** **DETALLES**

Predeterminado: 500 **Rango:** 100 – 5 000 **Definido en:**

 **TÉCNICO**

Número de iteraciones iniciales durante las que no ocurre relocalización MCMC. Solo tras este warmup entra en acción la lógica de reloc. Razón: en las primeras iteraciones los valores de opacidad no se han estabilizado — si reloc empezara directamente, los Gaussianos serían colocados en sitios equivocados y tendrían que moverse de nuevo enseguida, lo que destruiría el momentum de Adam. Default del paper 500. RadianceKit adopta este valor porque V464b mostró que es robusto.

 **EN POCAS PALABRAS**

Cuántas iteraciones se permite a MCMC «asentarse» primero antes de empezar a mover splats. 500 es estándar y encaja.

T68 **mcmcDeadOpacityThreshold** **DETALLES**

Predeterminado: 0,005 (initializer, estándar del paper), 0,01 (`.fullMCMC` y todos los ajustes preestablecidos MCMC — óptimo V535) **Rango:** 0,001 – 0,05 **Definido en:**

 **TÉCNICO**

Umbral sigmoid(opacity) por debajo del cual un Gaussiano cuenta como «muerto» y es elegible para relocalización. V535 encontró 0,01 como óptimo (0,005 marginal, 0,02 peor). Más alto = reloc más agresivo (más Gaussianos movidos), más bajo = más cauteloso. 0,01 corresponde aproximadamente a «0,5 % de visibilidad visual». P10 Indoor usa 0,0142 como óptimo vía Q7 BayesOpt.

 **EN POCAS PALABRAS**

Por debajo de qué transparencia un splat cuenta como «muerto» para que MCMC lo empuje a otro sitio. 0,01 es óptimo en nuestras pruebas. No necesitas trastear tú mismo.

T69 `mcmcNoiseDecayEnd` **DETALLES**

Predeterminado: 0 (initializer = «sin decay»), 160 000 (`.fullMCMC` = 80 % de 200K), 96 000 (`.mcmcBalanced` = 80 % de 120K), 40 000 (`.mcmcPreview`) **Rango:** 0 o 1 000 – **Definido en:**

 **TÉCNICO**

Iteración en la que el ruido `T63 mcmcNoiseScale` se amortigua del todo a cero (cosine decay desde iter 0 hasta aquí). V497c/V502 encontraron 80 % de `maxIterations` como óptimo — da a MCMC suficiente tiempo de exploración pero deja el último 20 % a la convergencia sin ruido. 0 = ruido constante sobre todas las iteraciones (rara vez sensato, MCMC no puede converger entonces).

 **EN POCAS PALABRAS**

Cuándo se detiene el «tambaleo» aleatorio de los splats. En los ajustes MCMC al 80 % del total de iteraciones — primero exploración, después convergencia. Deja el valor.

T70 `mcmcGrowthRate` **DETALLES**

Predeterminado: 0,05 (estándar del paper = 5 %) **Rango:** 0,01 – 0,2 **Definido en:**

 **TÉCNICO**

Tasa de crecimiento del tamaño objetivo de la población MCMC por paso de relocalización. La lógica: en cada evento reloc el tamaño objetivo de población se multiplica por $(1 + \text{growthRate})$, hasta alcanzar `T62 mcmcMaxGaussians` (o la variante escalada vía `T72/T73`). V512/V522 encontraron 0,05 como óptimo — valores más altos llevan a crecimiento demasiado rápido (los Gaussianos se insertan antes de que el momentum de Adam pueda integrarlos), valores más bajos a escenas sub-densificadas al final.

 **EN POCAS PALABRAS**

Lo rápido que crece el número de splats bajo MCMC. 5 % por paso es óptimo. Deja el valor.

T71 **mcmcSigmoidK** **DETALLES**

Predeterminado: 100,0 **Rango:** 10,0 – 500,0 **Definido en:**

 **TÉCNICO**

Parámetro de nitidez sigmoide para la atenuación de ruido MCMC. En el paso SGLD, el ruido por Gaussiano se amortigua por — los Gaussianos de alta opacidad (cuyo logit es positivo) reciben exponencialmente menos ruido que los de baja opacidad. $K = 100$ es nítido, significa que la transición de «ruido pleno» a «sin ruido» ocurre muy rápido en torno a opacidad 0,5. V484–V487 encontraron $K = 100$ como óptimo — valores más pequeños (10–50) también dejan tambalearse a los Gaussianos de alta opacidad (destruye los convergidos), los más grandes (> 500) hacen la transición artificialmente dura y los Gaussianos muertos ya no se mueven en absoluto.

 **EN POCAS PALABRAS**

Parámetro especial que determina lo nítidamente que MCMC distingue entre «lo bastante transparente para mover» y «sólido, no tocar». El predeterminado es óptimo. No trastear.

T72 **mcmcCapMultiplier** **DETALLES**

Predeterminado: 3,0 (initializer + `.fullMCMC`), 2,0 (`.mcmcPreview`), 2,5 (`.mcmcBalanced`), 2,98 (P8 Render), 5,32 (P9 Outdoor), 1,76 (P10 Indoor) **Rango:** 0 (= deshabilitado) o 1,0 – 10,0 **Definido en:**

 **TÉCNICO**

Función 1.4.5: escalado de tope consciente de la escena. Cuando `T73 mcmcAutoScaleByScene` es true, el tope efectivo se calcula como (clamped a la capacidad del buffer). Contexto: con escenas grandes (p. ej., vuelo de dron de 1 545 fotos → 158 K SfM init), `T62 = 150 000` es demasiado bajo — la density control no podría crecer en absoluto. Con multiplicador 3,0, el tope se escala a 474 K en este ejemplo (158 K × 3,0). Q7 BayesOpt encontró óptimos específicos de escena: outdoor se beneficia de un multiplicador alto (5,32 → ~830 K de tope con 156 K bicycle init), indoor se contenta con 1,76 (las paredes saturan más rápido). Para la resolución completa del tope véase el método.

 **EN POCAS PALABRAS**

Multiplicador que adapta automáticamente el tope de splats al tamaño de la escena. Escena grande = más puntos de inicio = mayor tope. El default 3× encaja para la mayoría de las escenas; el ajuste Outdoor va a 5× (grandes rangos de profundidad), Indoor a 1,76× (las paredes acotan las cosas de todas formas).

T73 **mcmcAutoScaleByScene** **DETALLES**

Predeterminado: true (initializer + todos los ajustes preestablecidos MCMC) **Rango:** booleano **Definido en:**

 **TÉCNICO**

Función 1.4.5: interruptor maestro para la lógica de tope consciente de la escena (véase T72 +). Cuando es false, solo se usa T62 `mcmcMaxGaussians` como tope (vuelta al comportamiento 1.4.4). On por defecto porque los problemas de extinción masiva con escenas grandes desde 1.4.3 volverían en otro caso. Deshabilítalo manualmente solo cuando quieras fijar explícitamente un tope duro — p. ej., para entrenar una variante 150 K cuyo tamaño final sea planificable.

 **EN POCAS PALABRAS**

Habilita el ajuste automático del tope de splats al tamaño de la escena. On por defecto. Déjalo off solo cuando quieras exactamente un número específico de splats.

Mip-Splatting (Q1.5) (T74–T76)

Estado: Q1.5 fue rechazado el 2026-05-25 tras 14 iteraciones autónomas + comprobación nocturna de confianza con 1,5 M como «closed no-win» (max $\Delta@2\times = +0,27$ dB, la puerta original requería $\geq +1,5$ dB de media sobre $0,5\times/2\times$, FALLA en 0/11 escenas pareadas). Los campos permanecen **opt-in** para experimentos de investigación; todos los ajustes preestablecidos de producción los tienen off. Véase el veredicto: `docs/plans/2026-05-25-phase-q1.5-final-verdict.md`.

T74 useMipSplatting**DETALLES**

Predeterminado: false (todos los ajustes de producción), true (`.fullMCMCMip` — hermano de investigación) **Rango:** booleano **Definido en:**

TÉCNICO

Habilita Mip-Splatting (Yu et al. CVPR 2024): filtro de suavizado 3D + filtro 2D + compensación α que limita la frecuencia por Gaussiano al bound de Nyquist de la tasa de muestreo de cámara de entrenamiento más densa. Objetivo teórico: eliminar aliasing al renderizar a escalas distintas del entrenamiento (0,5x o 2x de la resolución de entrenamiento). Habilitado en los shaders de preprocess y backward projection, corrección funcional verificada en el test Q1.5-D. Pero: la puerta de aceptación original ($\Delta@1x \geq +0,3$ dB Y $\text{avg}(\Delta@0,5x, \Delta@2x) \geq +1,5$ dB) no se alcanzó en ninguna de las 11 escenas pareadas. Máximo observado: family 750K classic $\Delta@2x = +0,270$ dB. Las escenas outdoor (Truck, Flowers) incluso mostraron empeoramiento en 1x y 0,5x. Hipótesis: el suavizado 3D compite con la relocalización MCMC a Gs alto. El campo se mantiene para una reevaluación multi-escala futura con metodología correcta Mip-NeRF-360 (véase el backlog O3 en la ruta de benchmark).

EN POCAS PALABRAS

Filtro de aliasing de un paper de 2024. Teóricamente genial, en la práctica no aportó nada en nuestras pruebas y a veces incluso hizo daño. Sigue disponible para los curiosos, pero no lo recomendamos. Déjalo off.

T75 mipSmoothing3DScale **DETALLES**

Predeterminado: 0,2 (default del paper) **Rango:** 0,05 – 1,0 **Definido en:**

 **TÉCNICO**

Parámetro de escala de suavizado 3D (Yu et al. §3.3, default del paper 0,2). Más grande = más suavizado en espacio mundo por Gaussiano (= más anti-aliasing pero también más blur a la escala default), más pequeño = más nítido pero más propenso a aliasing. Solo se consulta cuando `T74 useMipSplatting = true`. No se ha optimizado más en las pruebas Q1.5 — la puerta A/B ya se perdió con el default del paper 0,2, sweeps adicionales serían inútiles.

 **EN POCAS PALABRAS**

. Si no has habilitado Mip, irrelevante.

T76 mipFilter2DVariance **DETALLES**

Predeterminado: 0,3 (= exactamente el comportamiento legacy V242) **Rango:** 0,1 – 1,0 **Definido en:**

 **TÉCNICO**

Varianza del filtro 2D Mip añadida a la diagonal Σ_{2D} (varianza directamente, no al cuadrado). 0,3 es exactamente el valor legacy V242 que estaba codificado en duro en el kernel antes de Mip-Splatting. Cuando `T74 useMipSplatting = false`, el kernel ignora este valor por completo y escribe el 0,3 codificado — para que la línea base no pueda regresar (garantía S3-1 ronda 1 de Codex). Cuando, se usa el valor aquí fijado. Se queda en el catálogo de campos para sweeps Mip.

 **EN POCAS PALABRAS**

Otro parámetro de Mip-Splatting. Con Mip off: irrelevante.

Densificación adaptativa (Q5) (T77–T79)

T77 adaptiveDensification

DETALLES

Predeterminado: false **Rango:** booleano **Definido en:**

TÉCNICO

Función Q5: tracker de mediana rodante como alternativa al T11 `densifyGradThreshold` fijo. Cuando es true, en cada paso de `densify`, el umbral actual se sobrescribe con `median(últimas N muestras avgGrad) × T79`

`adaptiveDensifyMultiplier`. $N =$

`T78 adaptiveWindow`. Más estricto que el V440 p98 (la trampa catastrófica de pruning a 63 K), `median + 2×` se sitúa en torno al p70–p80 de la distribución de gradientes en estado estable. Pruebas Q5: sola FALLA 0/3 escenas, pero junto con Q6 (véase T80/T81) PASA 1/3 escenas — el bundle Q5+Q6 pasó el 2026-05-25 como opt-in y se habilita vía CLI `--adaptive-densify`. Q6 es el «portador» de la ganancia de calidad, Q5 contribuye más a la estabilidad.

EN POCAS PALABRAS

Umbral de `densify` autoaprendido. En lugar de un ajuste fijo de sensibilidad, la app se adapta a la escena. Por sí sola no fue mejor en pruebas, pero junto con el currículum de Q6 sí. O ambos on o ambos off.

T78 adaptiveWindow

DETALLES

Predeterminado: 1 000 **Rango:** 100 – 10 000 **Definido en:**

TÉCNICO

Ventana de mediana rodante en eventos de densificación (NO iteraciones — cada paso T13 `densifyInterval` da una muestra). Default 1 000 — con eso, las últimas 100 000 iteraciones de entrenamiento contribuyen a la mediana, así que típicamente toda la historia de entrenamiento hasta aquí. Fase temprana (antes de T78 muestras): el tracker devuelve nil → fallback al umbral fijo T11. Solo relevante cuando.

EN POCAS PALABRAS

Cuántos viejos pasos de `densify` alimentan la mediana para T77. Default 1000 es bueno. Solo relevante cuando Q5 `adaptive` está on.

T79 adaptiveDensifyMultiplier **DETALLES****Predeterminado:** 2,0 **Rango:** 1,0 – 4,0 **Definido en:** **TÉCNICO**

Multiplicador sobre la mediana rodante para el umbral adaptativo. Default 2,0 corresponde aproximadamente al p70–p80 de la distribución típica de gradientes. Más bajo = crecimiento más agresivo (más clones), más alto = más estricto (menos clones). Pruebas Q5 en rango 1,5–3,0 — 2,0 mejor default. Solo relevante cuando.

 **EN POCAS PALABRAS**

Factor para T77/T78. Default 2,0 = más estricto que la mediana típica. No trastear.

Currículum (Q6) (T80–T81)**T80** curriculumResolutionRamp **DETALLES****Predeterminado:** false **Rango:** booleano **Definido en:** **TÉCNICO**

Función Q6: la resolución de entrenamiento empieza en 0,5x y cambia en T50 $\text{positionLRScheduleEndIteration} / 2$ (o $T1 \text{ maxIterations} / 2$, si T50 no está fijado) a T22 $\text{trainingRenderScale}$. Usa la infraestructura `resize/restoreImageBuffers` desarrollada en Q1.5.1. So-brescribe T23 `resolutionWarmupScale` cuando está habilitado. Q6 pasó como «portador de la ganancia de calidad» en el bundle Q5+Q6 (véase T77) — el aumento gradual de resolución da a la app tiempo para encontrar la geometría gruesa a menor resolución antes de pasar al trabajo de detalle fino. Vía CLI: `--curriculum-resolution`.

 **EN POCAS PALABRAS**

«Primero grueso, después fino» para la resolución de entrenamiento. Media resolución en la primera mitad, después resolución completa. Ayuda en ciertas escenas, no en otras — mejor habilitarlo junto con T81.

T81 curriculumSHProgression **DETALLES**

Predeterminado: false **Rango:** booleano **Definido en:**

 **TÉCNICO**

Función Q6: sobrescribe T21

shDegreeUpgradeIterations con [maxIter/4, maxIter/2, maxIter*3/4], distribuyendo los upgrades SH de forma uniforme a lo largo del tiempo de entrenamiento en lugar de cargarlos al principio. Hipótesis: una geometría estable se establece antes de la explosión de detalle de color, lo que coloca con más precisión los efectos de brillo dependientes de la dirección de visión. Q5+Q6 juntos PASAN 1/3 escenas, Q6 como portador de la ganancia (Q5 sola FALLA). Vía CLI: `--curriculum-sh`.

 **EN POCAS PALABRAS**

«Primero forma, después color» — los efectos de brillo se desbloquean solo tarde en el entrenamiento, así que los splats primero encuentran su posición y tamaño. Puede habilitarse junto con T80; por sí solo no aporta tanto.

Ajustes preestablecidos estáticos (TP1–TP9)

Aquí solo las diferencias estructurales respecto al default del initializer. La descripción completa de marketing de los once ajustes preestablecidos UI P1–P11 está en el Capítulo 7.

TP1 `.preview` **DETALLES**

Ajuste preestablecido diagnóstico/previsualización para sistemas ≥ 10 GB de RAM. Overrides vs. initializer: - 30 000 \rightarrow 5 000 - 15 000 \rightarrow 3 500 (70 % de maxIter) - $1,6e-6 \rightarrow 1,6e-5$ (10x más alto, decay menos agresivo) -,,,,, cada 2x (V176) - 3 000 \rightarrow 100 000 (efectivamente off, V172: el reset destruye entrenamientos cortos) - [1K, 2K, 3K] \rightarrow [1K, 2K] (V182: el grado 3 no converge en 2K iter) - 1,0 \rightarrow 0,5

 **EN POCAS PALABRAS**

Valoración inicial de una serie de imágenes recién importada — 2–3 min de espera; tras eso, el resultado basta para un binario «¿vale la pena un run Quality?».

TP2 `.full`

DETALLES

Producción Quality Classic. Overrides: - 30 000 → 35 000 (V550: pruebas 40K Truck overtraining +10,7 % Gs con -1,3 % L1) - 15 000 → 5 000 (V310 sweet spot, V338 7K peor) - Todas las LRs 2x (V188) - 1,6e-6 → 1,6e-5 (V45 10x) - 2e-6 → 1,1e-6 (V335) - 100 → 200 (V112) - 0,005 → 0,001 (V393) - 3 000 → 100 000 (V194 deshabilitado, V421 confirmado) - [1K, 2K, 3K] → [2K, 5K, 8K] (V228 retrasado) - 0,0 → 0,9995 (V546 HTGS, 14 % de mejora) - 50 (sin cambios, V546) - false → true (V438) - 0 → 20 000 (V431) - true (V443, ya default initializer para `.full`)

EN POCAS PALABRAS

Cualquier captura fotográfica estándar (objeto, sala pequeña, escultura) con < 500 imágenes. La mejora de loss del 14 % anunciada en V546 vs V438 se confirmó como promedio de 3 trials en Horse Full.

TP3 `.fullClassicPaper`

DETALLES

Hermano de prueba Q1.5-A de TP2, Classic fiel al paper. Overrides vs. TP2: - 35 000 → 30 000 (estándar del paper) - 5 000 → 15 000 (paper: 50 % de maxlter) - 1,6e-5 → 1,6e-6 (default del paper) - ,, vuelta a defaults del paper (0,05, 0,005, 0,001) - 1,1e-6 → 2e-7 (calibrado para ~1-2M Gs en Bicycle) - 200 → 100 (paper) - 0,001 → 0,005 (default del paper) - 100 000 → 3 000 (paper §5.2, arriesgado — puede disparar la regresión V194) - 0,9995 → 0,0 (el paper no tiene decay) - 20 000 → 30 000 (cosine corre hasta 100 % de maxlter)

EN POCAS PALABRAS

Experimentos de investigación Q1.5 que necesitan presupuestos de Gaussianos de magnitud-paper (1-2 M) para pruebas Mip-Splatting. Tras el veredicto «closed no-win» de Q1.5 el ajuste preestablecido sigue accesible para usuarios avanzados pero no se recomienda en producción.

TP4 `.fullMCMC`

DETALLES

Producción Quality MCMC. Overrides vs. initializer:
 - 30 000 → 200 000 (V534, MCMC necesita 5x más iter que Classic) - 15 000 → 160 000 (V504b 80 % de maxIter) - $1,6e-6$ → $1,6e-5$ - Schedule de LR como TP2 (todas 2x) - 0,2 → 0,05 (V521b/V534: MCMC necesita señal L1 más fuerte)
 - [1K, 2K, 3K] → [2K, 5K, 8K] - `.classic` → `.mcmc` - 150 000 (ya en initializer, confirmado en el preset) - $5e-5$ (V467/V536 óptimo) - 0,005 → 0,01 (V535 óptimo) - 0 → 160 000 (80 % de maxIter, V497c/V502) - 3,0 (ya en initializer) - true (ya en initializer) - 3 000 → 200 000 (efectivamente off, MCMC usa reloc en lugar de reset)

EN POCAS PALABRAS

Entrega web, capturas de objeto con aspiraciones de detalle, vuelos de dron (aunque entonces P9 Outdoor es aún mejor). 71 % menos Gaussianos que Classic a L1 comparable.

TP5 `.fullMCMCMip`

DETALLES

Hermano de prueba Q1.5-D de TP4, con Mip-Splatting + presupuesto MCMC de magnitud-paper. Overrides vs. TP4: - `mcmcMaxGaussians` 150 000 → 1 500 000 (10x, magnitud paper) - `useMipSplatting` false → true (Mip on)

EN POCAS PALABRAS

Todos los demás campos idénticos a TP4. Q1.5 D-PASS en Bicycle 2026-05-24 (rompe la racha de FAIL multi-escala de 12 iter). El veredicto final Q1.5 del 2026-05-25 sigue siendo closed-no-win — ganancia Mip-Splatting no reproducible a través de 11 escenas pareadas. El ajuste preestablecido se mantiene opt-in.

TP6 `.classicBalanced`

DETALLES

Mid-tier Classic. Overrides vs. TP2: - 35 000 → 20 000 (V149: 20K = 30K con 33 % menos tiempo) - 20 000 → 0 (cosine corre hasta maxIter = 20K, sin fase extendida)

EN POCAS PALABRAS

Casos estándar con menor tiempo de espera. V149 identificado como sweet spot.

TP7 `.mcmcPreview`

DETALLES

Diagnóstico MCMC. Overrides vs. TP4: - 200 000 → 60 000 (V494b) - 160 000 → 48 000 (80 %) - 150 000 → 100 000 (V473b) - 160 000 → 40 000 (V494b) - 3,0 → 2,0 (1.4.5: Preview = escalado más ligero)

EN POCAS PALABRAS

Ver rápido un resultado MCMC para juzgar si vale la pena TP4 o un ajuste Scene-Class.

TP8 `.mcmcBalanced`

DETALLES

Mid-tier MCMC. Overrides vs. TP4: - 200 000 → 120 000 (V518) - 160 000 → 96 000 (80 %) - 160 000 → 96 000 (80 %) - 3,0 → 2,5 (entre Preview 2,0 y Full 3,0)

EN POCAS PALABRAS

MCMC sin el run completo 200K. ~120 K iteraciones son un buen compromiso entre calidad y tiempo de espera.

TP9 `.quickTest`

DETALLES

Prueba de funcionamiento pura. Overrides vs. initializer: - 30 000 → 1 000 - 15 000 → 500 - 2e-6 → 4e-6 (calibrado para 0,25x resolución) - 100 → 50 - 3 000 → 100 000 (off, ya que demasiado corto) - 1,0 → 0,25

EN POCAS PALABRAS

Chequeo de sanidad «¿se inicia el entrenamiento con sentido en absoluto?». Duración < 30 s en M3 Ultra. Garantizado que se ve blando.

Método:

Firma: `public func resolveMcmcMaxGaussians(initialPointCount: Int, bufferCapacity: Int) -> Int` **Definido en:**

TÉCNICO Fuente única de verdad para la pregunta «¿hasta cuántos Gaussianos se puede dejar crecer a MCMC?» Calculado a partir de tres entradas: el `T62 mcmcMaxGaussians` configurado (con suelo de extinción masiva 150 000 si 0), el (número de puntos init de SfM) y el (tamaño de buffer Gaussiano preasignado). Lógica:

1. `base = T62 > 0 ? T62: 150_000` (el suelo de extinción masiva protege contra bugs de default de initializer como el incidente de extinción masiva 1.4.3)
2. Si `T73 mcmcAutoScaleByScene && initialPointCount > 0 && T72 mcmcCapMultiplier > 0`:
 - `scaled = max(base, ceil(initialPointCount × T72))` else
3. Si `bufferCapacity > 0`: `return min(scaled, bufferCapacity)`

4. Else `return scaled`

Ejemplo: Bicycle (Mip-NeRF 360, 194 frames de foto) → SfM init ~156 K puntos, `T62 = 150 000`, `T72 = 5,32`, capacidad de buffer 8 M. Tope resuelto = $\min(8M, \max(150K, \text{ceil}(156K \times 5,32))) = \min(8M, 830K) = 830 K$. Ese es el tope de crecimiento efectivo al que se atiene la lógica de relocalización MCMC.

EN PALABRAS CLARAS Calcula el número máximo real de splats bajo MCMC. Toma tu ajuste, mira con cuántos puntos empieza tu escena, y escala por el `Multiplier`, si está activada la adaptación automática. Así el tope se adapta a la escena en lugar de forzar el mismo valor para una escena diminuta y una enorme. No tienes que llamar tú al método — el entrenamiento lo usa internamente.

¿Qué campo para qué? (Cheat Sheet)

Objetivo	Campos a ajustar
Más detalle a la distancia	<code>T62 mcmcMaxGaussians</code> arriba, <code>T72 mcmcCapMultiplier</code> 5+
Más detalle en general (Classic)	<code>T1 maxIterations</code> arriba ($\leq 40K$), <code>T2 densifyUntilIteration</code> $\leq 14\%$ de T1
Reducir floaters en vuelos de dron	<code>T43 frustumCullEnabled</code> on, <code>T20 skyMaskingEnabled</code> on, <code>T45 skyDomeEnabled</code> on
Cielo agradable en escenas exteriores	<code>T45 skyDomeEnabled</code> on, <code>T47 skyDomeRadiusMultiplier</code> 30–60
Archivo de exportación más pequeño	Estrategia <code>.mcmc</code> (T61), <code>T56 postTrainingCompactification</code> on, <code>T62 mcmcMaxGaussians</code> $\leq 200K$
Entrenamiento más rápido	<code>T22 trainingRenderScale</code> 0,5, <code>T1 maxIterations</code> a la mitad — ¡pero no ambos!
Mejores brillos	<code>T21 shDegreeUpgradeIterations</code> con [2K, 5K, 8K] (sin front-load temprano), MCMC + 200K iter
Mantener el Mac responsivo	<code>T25 throttleDelayMs</code> 5–10 (cuesta ~15 % de tiempo de entrenamiento)
Previsualización en vivo más a menudo	<code>T59 livePreviewInterval</code> abajo a 10–20
Transiciones más suaves en sombras	<code>T17 ssimWeight</code> ligeramente arriba (0,15–0,25), pero no por encima de 0,3
Mantener compactos los interiores	Ajuste preestablecido P10 Indoor (, <code>T72 = 1,76</code>)

Campos peligrosos

Estos campos pueden, con una mala configuración, llevar a OOM, crash de la app, extinción masiva de Gaussianos o datos de benchmark inutilizables. Trátalos con cuidado:

- T11 `densifyGradThreshold` — partir a la mitad puede crear 2–4x más Gaussianos, lo que rápidamente revienta la memoria GPU. Ten en cuenta también: debe coincidir con T22 `trainingRenderScale` ($1,0x \rightarrow 1e-6$, $0,5x \rightarrow 2e-6$, $0,25x \rightarrow 4e-6$).
- T72 `mcmcCapMultiplier` — con escenas grandes con > 200 K puntos init SfM y un multiplicador > 5, surge un tope resuelto de millones de Gaussianos. En Macs de 36 GB de RAM, OOM es posible. El ajuste preestablecido Outdoor 5,32 funciona solo porque Mip-NeRF 360 Bicycle tiene 156 K puntos init \rightarrow tope 830 K.
- T39 `testViewIndices` — fijarlo manualmente puede inutilizar el benchmark (todos los índices > N \rightarrow sin holdouts). Deja que lo fije el flag `--benchmark`.
- T64 `mcmcOpacityRegWeight` y T65 `mcmcScaleRegWeight` — En la beta 1.4.3 puestos a 0,01, lo que llevó a la extinción masiva (460 K \rightarrow 5 Gaussianos en una iteración). Desde 1.4.4 fijados en 0,0, pero subirlos manualmente puede reproducir el problema.
- T15 `opacityResetInterval` — si no está en 100 000+ (efectivamente off) y el entrenamiento es más corto que 10 000 iteraciones, el reset destruye la convergencia. `.preview` por eso lo tiene en 100 000 a pesar de `maxIterations = 5 000`.
- T54/T55 `densifyPhase2*` — la densificación de dos fases acabó en pruebas en una cascada de 0 Gaussianos. Deja ambos en 0.
- T74 `useMipSplattting` — Q1.5 closed-no-win 2026-05-25, puede incluso empeorar el PSNR en algunas escenas outdoor. Default off, opt-in solo para investigación.

Si un campo está en esta lista y quieres cambiarlo, primero haz copia de seguridad de tu ajuste preestablecido actual (exporta como JSON) y considera si puedes medir el resultado de forma reproducible — en otro caso, no sabrás después si has provocado una mejora o un empeoramiento.

CAPÍTULO

Capítulo 7 — Ajustes preestablecidos de calidad integrados

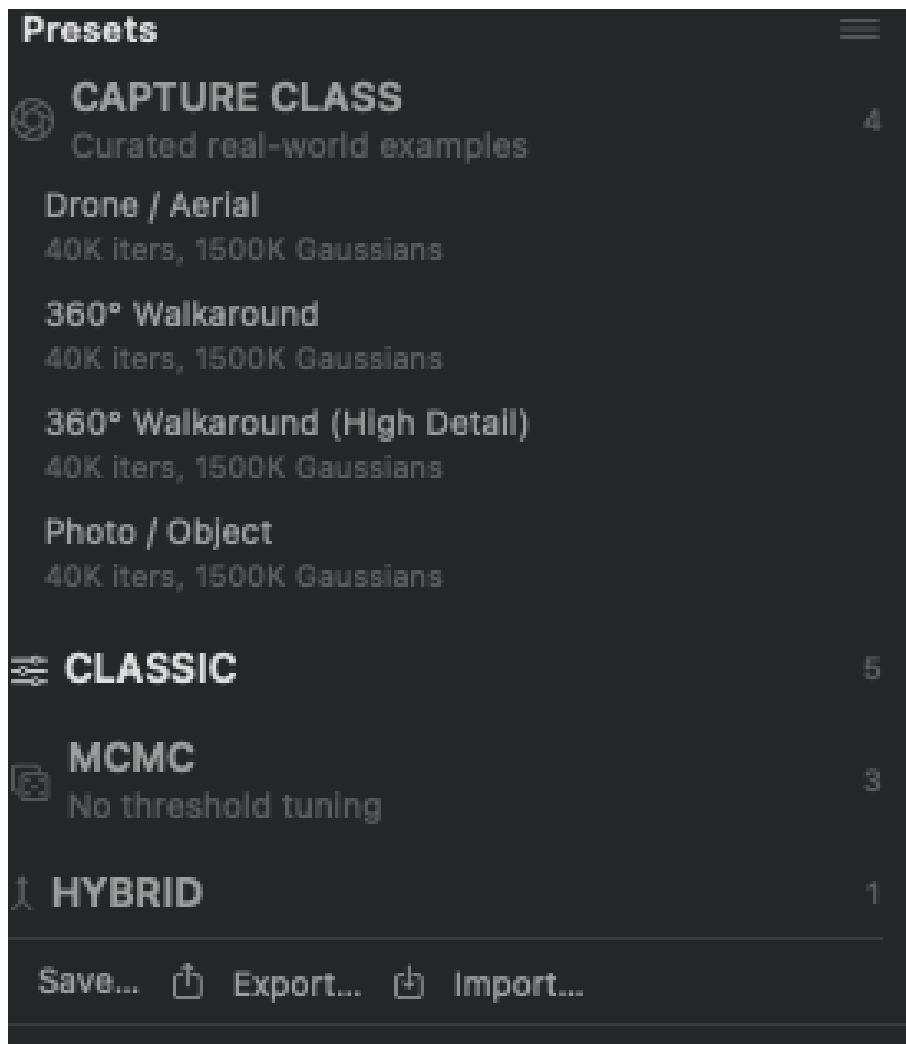


Figura 27: Sección de ajustes preestablecidos con los cuatro grupos expandidos — CAPTURE CLASS (4 ajustes: Drone/Aerial, 360° Walkaround, 360° Walkaround (High Detail), Photo/Object), CLASSIC (5 ajustes: Quick/Preview/Balanced/Quality/Ultra Detail), MCMC (3 ajustes, nota „No threshold tuning«), HYBRID (1 ajuste: Balanced (Hybrid))

LO QUE SE VE EN LA IMAGEN Sección de ajustes preestablecidos en el Inspector, los cuatro grupos expandidos. CAPTURE CLASS con los cuatro ajustes curados para el mundo real (Drone / Aerial, 360° Walkaround, 360° Walkaround (High Detail), Photo / Object) — es el grupo principal y el único visible en el modo principiante. CLASSIC con Quick (1K

iter), Preview (5K iter, selección activa con marca de verificación azul), Balanced (20K iter), Quality (35K iter) y Ultra Detail (35K iter). MCMC con el subtítulo „No threshold tuning« — MCMC no necesita umbral Densify-Until: Preview (60K iter, 100K Gaussianos), Balanced (120K, 150K), Quality (200K, 150K). HYBRID con el Balanced (Hybrid) (20K iter, 150K Gaussianos). Fila de acciones al pie: Save..., Export..., Import...

Un ajuste preestablecido es una configuración preparada para el entrenamiento.

RadianceKit incorpora trece ajustes preestablecidos en cuatro grupos: cuatro ajustes **Capture-Class** (P9–P12) — recetas curadas, validadas a ojo sobre material real de la comunidad para tipos reales de captura (dron, recorrido 360°, foto-objeto) y el eje principal desde la v1.6 —, cinco ajustes Classic (P1–P5: Quick/Preview/Balanced/Quality/ Ultra Detail), tres ajustes MCMC (P6–P8) y un ajuste Híbrido (P13) que combina las estrategias Classic y MCMC. Los antiguos ajustes «Scene-Class» (Render/3D, Outdoor, Indoor, sintonizados en la Fase Q7 de forma académica frente a escenas de Mip-NeRF-360 y NeRF-Blender) se retiraron como grupo visible en la v1.6 — la Capture-Class, validada a ojo sobre material real, es ahora el eje principal; las configuraciones sintonizadas en Q7 se conservan solo internamente. Eliges los ajustes preestablecidos en la barra lateral en **Presets** o en el modo principiante durante la importación. Los botones **+** abren diálogos para crear tus propios ajustes preestablecidos en paralelo — los trece integrados no se pueden eliminar, pero sí duplicar.

En la vista Experto los ajustes preestablecidos aparecen agrupados por tipo de captura y estrategia (Capture Class / Classic / MCMC / Híbrido). Un clic sobre una entrada escribe la configuración de entrenamiento almacenada en el estado actual. No es una instantánea — si después mueves deslizadores, el estado cambia, pero el propio ajuste preestablecido permanece intacto; un aviso de color indica entonces «modificado».

Cuál es el ajuste preestablecido adecuado en cada caso depende principalmente del tipo de escena y del hardware. Las tres tablas generales al final del capítulo lo resumen.

| P1 — Quick

DÓNDE

Inspector → Sección Presets → Grupo «Classic» → Entrada «Quick». Sufijo UUID ...001 .

TÉCNICO

Ajuste preestablecido de diagnóstico con 1 000 iteraciones, estrategia clásica (adaptativa) de densificación y un escalado de resolución de entrenamiento de 0,25× (la imagen de entrada se reduce al 25 % antes del entrenamiento). No pretende entregar una escena, sino determinar rápidamente si el setup (poses de cámara, nube de puntos, serie de imágenes) muestra siquiera movimiento significativo en los valores de loss. En un M3 Ultra suele tardar menos de 30 segundos con 50–200 imágenes. La pequeña resolución oculta la calidad real de la imagen, pero mantiene la huella de memoria y el esfuerzo de renderizado muy bajos. También se selecciona automáticamente por defecto en el primer arranque cuando el sistema tiene menos de 10 GB de RAM.

EN POCAS PALABRAS

Prueba rápida de funcionamiento. Metes las imágenes, esperas medio minuto y ves si aparece el contorno aproximado de la escena. Si la imagen en el visor parece una masa difusa — encaja, así debe ser. Si en cambio solo ves puntos oscuros o una forma totalmente distorsionada, probablemente las poses de cámara son incorrectas (véase Capítulo 9). Para un resultado compartible necesitas como mínimo P2 o P3.

| P2 — Preview (Classic)

DÓNDE

Inspector → Sección Presets → Grupo «Classic» → Entrada «Preview». Sufijo UUID ...002 .

TÉCNICO

5 000 iteraciones de densificación Classic, escalado de resolución 0,5×, tasas de aprendizaje duplicadas respecto al estándar. La densificación (clonado + división) está activa durante las primeras 2 500 iteraciones, después solo poda. Ajuste preestablecido por defecto para sistemas con ≥ 10 GB de RAM. En un M3 Ultra suelen ser de 90 segundos a 3 minutos para una escena de 200 imágenes. Entrega una impresión utilizable de geometría y pose de cámara, pero las texturas se ven visiblemente blandas — la resolución de renderizado 0,5× no se puede sortear directamente después reentrenando con P3 o P4, porque las tasas de aprendizaje están calibradas para coincidir con la media resolución.

EN POCAS PALABRAS

Es el estándar para «echar un vistazo rápido». Si acabas de importar imágenes nuevas y quieres ver si la escena se puede reconstruir en absoluto, este es el nivel adecuado. Unos 2–3 minutos de espera y luego puedes rotar en el visor 3D y juzgar si vale la pena invertir en más entrenamientos. Solo cuando el resultado de la previsualización ya tiene buen aspecto vale la pena pasar a Balanced o Quality.

I P3 — Balanced (Classic)

DÓNDE

Inspector → Sección Presets → Grupo «Classic» → Entrada «Balanced». Sufijo UUID `...005`.

TÉCNICO

20 000 iteraciones de densificación Classic a resolución completa. La densificación se ejecuta durante las primeras 15 000 iteraciones, a partir de la iter 3 000 con un intervalo de densificación de 100. Empíricamente el «punto óptimo» de las sesiones de entrenamiento documentadas: con densificación clásica en Horse Full y Truck, el loss L1 se estabiliza entre la iter 18 000 y la 22 000; un entrenamiento más largo no aporta mejoras significativas por debajo de Quality (P4). En un M3 Ultra suelen ser 30–60 segundos con 200 imágenes, 5–8 minutos con 1 000+ imágenes.

EN POCAS PALABRAS

El «buen compromiso». La mayoría de las escenas ya se ven bien con esto, sin que tengas que esperar una hora. Si quieres mostrar el resultado final en alguna parte (redes sociales, web, una demo para un cliente), suele ser suficiente. Solo cuando quieres hacer zoom en el modelo de splats o necesitas detalles de textura de superficie vale la pena saltar a P4 Quality o P7 MCMC.

I P4 — Quality (Classic)

DÓNDE

Inspector → Sección Presets → Grupo «Classic» → Entrada «Quality». Sufijo UUID `...003`.

TÉCNICO

35 000 iteraciones de densificación Classic con V546 «Opacity Decay» (HTGS, Eurographics 2025): después de cada ciclo de densificación, la opacidad de todos los Gaussianos existentes se multiplica por un factor $< 1,0$, lo que elimina de manera fiable durante la poda los Gaussianos que se han vuelto inactivos y, así, consigue un 14 % mejor de loss L1 con el mismo número de iteraciones que el run clásico de 35 000. El loss SSIM está habilitado (`ssimWeight=0.05`). En un M3 Ultra suelen ser 2–4 minutos con 200 imágenes. Alcanza un L1 final $\approx 0,023$ en NeRF-Blender (Lego, Chair, Drums) — la mejor variante Classic en los más de 560 experimentos documentados. Nota: necesita ~3–5 GB de memoria GPU; en sistemas de 8 GB, P3 es la opción segura.

EN POCAS PALABRAS

La mejor variante clásica. Entrega texturas nítidas y geometría fina, especialmente en capturas de objetos (una escultura, una silla, un jarrón). En grandes escenas exteriores o salas, sin embargo, apenas notas diferencia con Balanced — ahí vale más la pena pasarse a un ajuste preestablecido MCMC (P6–P8) o a uno Capture-Class (P9–P12) que dar el salto de P3 a P4. Quien quiera el máximo absoluto de la familia Classic, toma P5 Ultra Detail.

I P5 — Ultra Detail (Classic)

DÓNDE

Inspector → Sección Presets → Grupo «Classic» → Entrada «Ultra Detail». Sufijo UUID `...008`.

TÉCNICO

Alrededor de 35 000 iteraciones de densificación Classic — el ganador del run held-out de la matriz de calidad (2026-06-10). En las tres escenas Mip-NeRF-360 probadas, Ultra Detail supera al ajuste preestablecido MCMC «Quality» integrado (P8) con un tiempo de wall-clock comparable en torno a +0,94 dB PSNR. Es así el ajuste preestablecido Quality más fuerte del grupo Classic y la variante Classic más nítida que entrega RadianceKit. En un M3 Ultra suele estar en el mismo rango de tiempo que P4 Quality (2–5 minutos con 200 imágenes), pero necesita algo más de memoria GPU; en sistemas de 8 GB, P3 sigue siendo la opción segura.

EN POCAS PALABRAS

El nivel Classic más nítido y el ganador held-out de nuestras pruebas de calidad: en escenas reales, alrededor de un decibelio mejor que la variante MCMC «Quality» — con un tiempo de espera similar. Si quieres máxima fidelidad de detalle con la probada densificación clásica y tienes suficiente memoria GPU, esta es la primera elección. Si la memoria no alcanza o necesitas un archivo de exportación lo más pequeño posible, quédate con P4 Quality o con un ajuste preestablecido MCMC.

I P6 — Preview (MCMC)

DÓNDE

Inspector → Sección Presets → Grupo «MCMC» → Entrada «Preview». Sufijo UUID `...006`.

TÉCNICO

60 000 iteraciones de densificación MCMC (3DGS-MCMC, NeurIPS 2024) con un tope de 100 000 Gaussianos. MCMC reemplaza la lógica heurística de clone/split por una relocalización de cadena de Markov Monte Carlo: los Gaussianos muertos se reubican mediante profundidades de muestreo ponderadas por sigmoide, lo que arroja un número controlado y reproducible de Gaussianos. El tope limita estrictamente el máximo a 100K — ahorra memoria y tiempo de renderizado, pero cuesta detalle. En un M3 Ultra suelen ser 5–8 minutos con 200 imágenes. Adecuado como «prueba de funcionamiento MCMC» — te ayuda a valorar si pasar de Classic a MCMC tendría sentido antes de invertir más tiempo en P7 o P8.

EN POCAS PALABRAS

Como P2 Preview, pero con el método más reciente MCMC. A menudo entrega nubes de splats más compactas y distribuidas con mayor uniformidad que la variante Classic. Para una valoración inicial de una escena, los 5–8 minutos suelen ser suficientes. Si te gusta el resultado de previsualización, el siguiente paso es P7 (Balanced) o directamente P8 (Quality MCMC).

I P7 — Balanced (MCMC)

DÓNDE

Inspector → Sección Presets → Grupo «MCMC» → Entrada «Balanced». Sufijo UUID `...007`.

TÉCNICO

120 000 iteraciones MCMC con un tope de 150 000 Gaussianos. El nivel intermedio de MCMC — casi el número final de Gaussianos de P8 Quality, pero solo el 60 % de las iteraciones. Empíricamente, el loss L1 en las sesiones de entrenamiento documentadas se sitúa entre 0,026–0,028 sobre Horse Full, frente a P8 con 0,0246 — es decir, alrededor de un 7 % más alto, pero con la mitad de tiempo de espera. En un M3 Ultra suelen ser 8–15 minutos con 200 imágenes. Utiliza un procedimiento que escala el tope efectivo de Gaussianos a la densidad de puntos de la nube SfM de entrada (véase T75 en el Capítulo 6).

EN POCAS PALABRAS

MCMC con una profundidad de detalle decente, pero sin el largo run completo de P8. Para la mayoría de las escenas con esto basta, especialmente cuando quieres meter un run de MCMC en un hueco de tiempo de comida. Si la memoria está justa (p. ej., en procesadores M con solo 16 GB), quédate aquí — P8 necesita más memoria GPU.

I P8 — Quality (MCMC)

DÓNDE

Inspector → Sección Presets → Grupo «MCMC» → Entrada «Quality». Sufijo UUID `...004`.

TÉCNICO

200 000 iteraciones MCMC con un tope de 150 000 Gaussianos, loss SSIM 0,05, decay de ruido MCMC sobre el 80 % de las iteraciones. Mejor L1 en un solo run de los más de 560 experimentos: 0,0238 en Horse Full, promediado sobre 3 trials 0,0246 (frente a P4 0,0230 en la misma escena). MCMC entrega un 71 % menos de Gaussianos (150K frente a ~524K) — crucial cuando quieres entregar el resultado en la web, porque la nube más pequeña produce archivos de exportación notablemente más pequeños. Tiempo de entrenamiento en un M3 Ultra, normalmente 20–35 minutos con 200 imágenes; en sets de 1 000+ imágenes, más bien 1–2 horas. Mejor elección cuando se desea máxima calidad de imagen con un tamaño final mínimo.

EN POCAS PALABRAS

La mejor variante MCMC. Entrega nubes de splats muy limpias y compactas — ideal cuando luego quieres embeber el resultado como visor 3D web o enviarlo como archivo (el archivo es más pequeño que P4 Quality con una calidad óptica comparable). Pero necesitas paciencia — en capturas grandes, más de una hora de espera. Plántalo más bien como un run «de la noche».

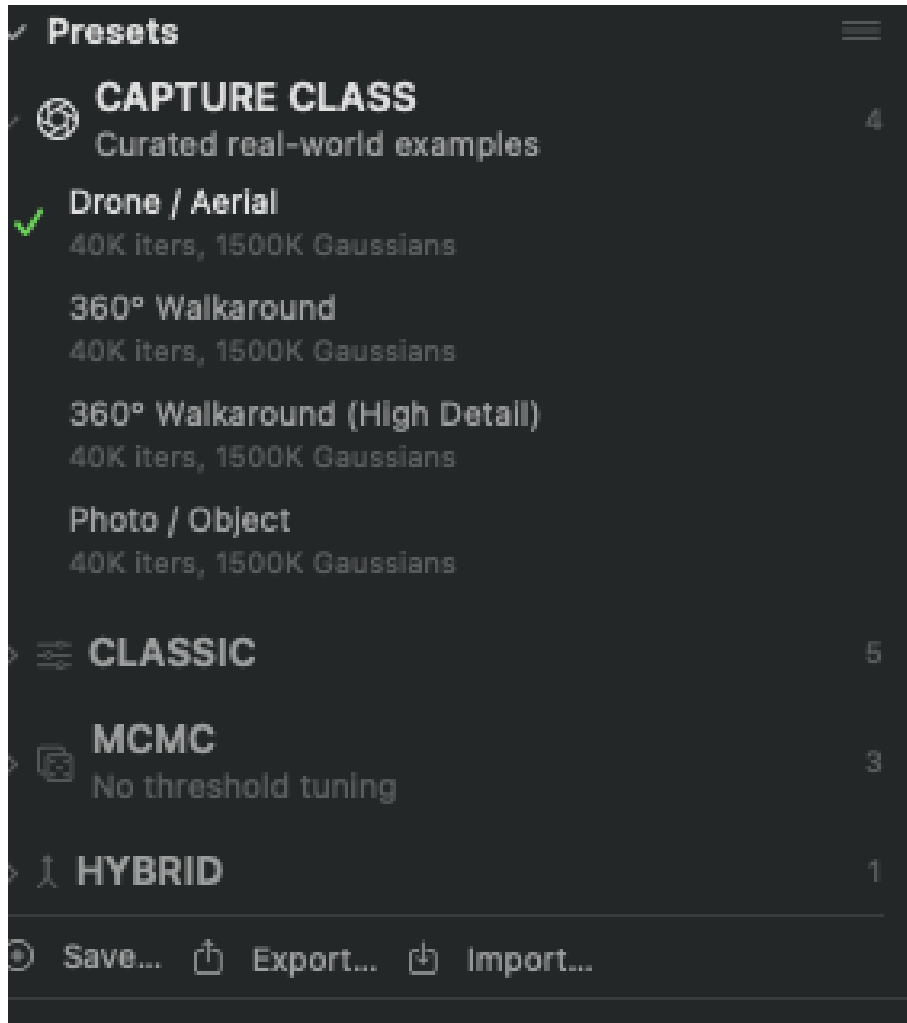


Figura 28: Grupo CAPTURE CLASS expandido con los cuatro ajustes curados para el mundo real — Drone / Aerial (MCMC, 40K iter, tope 1,5 M), 360° Walkaround (MCMC, 40K, tope 1,5 M), 360° Walkaround (High Detail) (Híbrido, 40K, tope 1,5 M, opt-in) y Photo / Object (Híbrido, 40K, tope 1,5 M). Este grupo está arriba del todo y es el único visible en el modo principiante.

LO QUE SE VE EN LA IMAGEN Inspector con el grupo CAPTURE CLASS expandido — el grupo principal de ajustes preestablecidos desde la v1.6, el único mostrado en el modo principiante. Cada entrada es una receta validada a ojo sobre material real de la comunidad para un tipo concreto de captura (dron, recorrido 360°, foto-objeto), no un valor optimizado frente a un test set académico. La selección por clic escribe la configuración de entrenamiento almacenada en el estado actual.

| P9 — Drone / Aerial

DÓNDE

Inspector → Sección Presets → Grupo «Capture Class» → Entrada «Drone / Aerial». Sufijo UUID ...010 .

TÉCNICO

Ajuste preestablecido Capture-Class para órbitas aéreas y de dron de edificios y paisajes. Densificador MCMC, 40 000 iteraciones, tope 1,5 millones de Gaussianos, loss SSIM 0,5 más término edge-aware 0,1. Lo decisivo es la penalización de anisotropía con peso 0,003 a un umbral de relación de 6 — el «mataespaguetis» contra los típicos artefactos en forma de aguja que produce el material de dron. Validado en un vuelo de dron DJI 4K real sobre el viaducto de Pensford (comprobado a ojo, no solo métricamente).

EN POCAS PALABRAS

Para tomas desde el aire — vuelos de dron alrededor de un edificio, sobre un paisaje, a lo largo de una fachada. La fuerte penalización de anisotropía limpia los artefactos en forma de aguja o espagueti que el material de dron suele producir. Si tu material está capturado desde el suelo, encaja mejor Photo / Object o un ajuste preestablecido Classic.

| P10 — 360° Walkaround

DÓNDE

Inspector → Sección Presets → Grupo «Capture Class» → Entrada «360° Walkaround». Sufijo UUID ...011 .

TÉCNICO

Ajuste preestablecido Capture-Class para vídeos de recorrido 360°. Densificador MCMC, 40 000 iteraciones, tope 1,5 millones de Gaussianos, loss SSIM 0,5 más término edge-aware 0,1, penalización de anisotropía suave (peso 0,001 a umbral de relación 15). La máscara de personas y de cielo están activas. El ajuste preestablecido espera un vídeo equi-rectangular 360° que internamente se re proyecta a recortes en perspectiva de unos 90° de ancho antes de iniciar el entrenamiento. Validado en recorridos 360° en 8K con palo selfie (escena Monument, comprobado a ojo).

EN POCAS PALABRAS

Para vídeos de recorrido 360° — caminas con una cámara 360° o un palo selfie por una sala o alrededor de un objeto. RadianceKit descompone él mismo el panorama esférico en ángulos de visión normales y enmascara transeúntes y cielo. Para máxima nitidez sobre el mismo material, prueba además la variante High Detail (P11).

I P11 — 360° Walkaround (High Detail)

DÓNDE

Inspector → Sección Presets → Grupo «Capture Class» → Entrada «360° Walkaround (High Detail)». Sufijo UUID ...013 (Opt-in).

TÉCNICO

Ajuste preestablecido Capture-Class opt-in para vídeos de recorrido 360° con máximo detalle. Densificador Híbrido (clonado/división clásico por gradiente absoluto + ruido MCMC + relocalización), 40 000 iteraciones, tope 1,5 millones de Gaussianos, penalización de anisotropía 0,0015 a umbral de relación 15, loss SSIM 0,2 y término edge-aware 0 — la receta de screen-split «r50» bloqueada. En material 360° supera al ajuste preestablecido MCMC estándar «360° Walkaround» (P10) en PSNR, LPIPS y confeti visible, y con cerca de un tercio del número de splats. Está deliberadamente opt-in *junto* al ajuste preestablecido 360 estándar, hasta que se valide en más escenas.

EN POCAS PALABRAS

La alternativa más nítida al ajuste preestablecido 360 estándar (P10): más detalle, menos confeti, archivo notablemente más pequeño. Está deliberadamente al lado en vez de reemplazarlo — confirmado hasta ahora en un puñado de escenas. Si tu recorrido 360° está capturado limpiamente, prueba este ajuste preestablecido primero y compara el resultado con P10.

I P12 — Photo / Object

DÓNDE

Inspector → Sección Presets → Grupo «Capture Class» → Entrada «Photo / Object». Sufijo UUID ...012 .

TÉCNICO

Ajuste preestablecido Capture-Class para órbitas de objeto a partir de fotos individuales nítidas (sin vídeo). Densificador Híbrido-t1 (con relocalización), 40 000 iteraciones, tope 1,5 millones de Gaussianos, loss SSIM 0,5 más término edge-aware 0,1, penalización de anisotropía suave (peso 0,001 a umbral de relación 15), Opacity Decay 0,9995 cada 50 iteraciones, **sin** enmascarado. Validado en 163 fotos de alta resolución de 41 MP de un esqueleto (comprobado a ojo). Pocas vistas (hasta unas 600) se mantienen así por debajo del umbral de colapso del Híbrido.

EN POCAS PALABRAS

Para capturas de objeto a partir de fotos individuales nítidas — rodeas una escultura, una maqueta, un producto con la cámara y haces fotos en vez de vídeo. Sin enmascarado, porque las fotos nítidas suelen tener un fondo limpio. Para fuentes de vídeo toma en su lugar un ajuste preestablecido 360° o Drone.

I P13 — Equilibrado (Híbrido)

DÓNDE

Inspector → Sección Presets → Grupo «Híbrido» → Entrada «Equilibrado (Híbrido)». Sufijo UUID ...009 .

TÉCNICO

20 000 iteraciones con la estrategia de densificación Híbrida: el clone/split clásico guiado por gradiente coloca capacidad donde la pérdida la necesita, el ruido SGLD de MCMC sigue explorando, y los Gaussianos muertos se relocalizan en lugar de perderse en la poda. El decaimiento de opacidad (V546) sustituye los reinicios de opacidad; una penalización de anisotropía (peso 0,001, umbral de relación 15) mantiene a raya los splats en forma de aguja. El tope de Gaussianos escala con la escena (base 150K, scene-aware $\times 3,0$). Validado en cinco escenas frente a MCMC puro con el mismo presupuesto: de media +0,45 dB PSNR con un 20–30 % menos de Gaussianos (stonehenge +1,23, family +0,82, garden +0,47 dB). En un M3 Ultra, típicamente 5–10 minutos con 200 imágenes.

EN POCAS PALABRAS

Una primera elección fuerte para un resultado final: detalle más nítido que los ajustes preestablecidos MCMC con un archivo igual de compacto, en una fracción del tiempo de entrenamiento de P8. Si solo tienes tiempo para un run de calidad y ninguna de las clases de captura encaja claramente, empieza aquí. Los ajustes preestablecidos Classic siguen siendo mejores para pruebas rápidas, y los ajustes preestablecidos Capture-Class (P9–P12) son la primera elección cuando tu escena encaja claramente en uno de esos tipos de captura.

¿Qué ajuste preestablecido en cada caso?

Escenario	Primera prueba	Run principal
Prueba de funcionamiento de imágenes nuevas, < 30 s	P1 Quick	—
Órbita de objeto a partir de fotos individuales nítidas	P2 Preview	P12 Photo / Object
Escaneo de un solo objeto (vídeo), < 500 fotos	P2 Preview	P4 Quality o P8 Quality MCMC
Vídeo de recorrido 360°	P6 Preview MCMC	P10 360° Walkaround (nítido: P11 High Detail)
Órbita aérea / de dron, paisaje	P6 Preview MCMC	P9 Drone / Aerial
Entrega web (pequeño, compacto)	P2	P8 Quality MCMC (archivo más pequeño con calidad completa)
Detalle nítido en poco tiempo, exportación compacta	P2 o P6	P13 Equilibrado (Híbrido)
Máxima fidelidad de detalle, estrategia Classic	P3 o P6	P5 Ultra Detail
Impresión, marketing, detalle completo	P3 o P6	P4 Quality (Classic) o P5 Ultra Detail

Comparativa rápida

Pre-set	Estrategia	Iter.	Máx-Gs	Escala render	Tiempo típico (200 imágenes, Q-Sweep M3 Ultra)	
P1 Quick	Classic	1 000	∞	0,25x	~30 s	—
P2 Preview	Classic	5 000	∞	0,5x	2–3 min	—
P3 Balanced	Classic	20 000	∞	1,0x	30–60 s	—
P4 Quality	Classic	35 000	∞	1,0x	2–4 min	V546 HTGS
P5 Ultra Detail	Classic	~35 000	∞	1,0x	2–5 min	Matriz $\Delta+0,94$ dB
P6 Preview MCMC	MCMC	60 000	100 K	1,0x	5–8 min	—
P7 Balanced MCMC	MCMC	120 000	150 K	1,0x	8–15 min	—
P8 Quality MCMC	MCMC	200 000	150 K	1,0x	20–35 min	V544a
P9 Drone / Aerial	MCMC	40 000	1,5 M	1,0x	10–25 min	Ojo / Vialucto
P10 360° Walkaround	MCMC	40 000	1,5 M	1,0x	10–25 min	Ojo / Monument
P11 360° Walkaround (High Detail)	Híbrido	40 000	1,5 M	1,0x	10–25 min	Ojo (opt-in)
P12 Photo / Object	Híbrido	40 000	1,5 M	1,0x	10–25 min	Ojo / Esqueleto
P13 Equilibrio (Híbrido)	Híbrido	20 000	150 K	1,0x	5–10 min	Matriz $\Delta+0,45$ dB

Ajustes preestablecidos personalizados

Con el botón **Save...** en la sección Presets (I1 en el Capítulo 2) guardas la configuración de entrenamiento actual como tu propio ajuste preestablecido. Los ajustes preestablecidos personalizados no son «Built-in» y se pueden renombrar, exportar (como JSON), compartir mediante arrastrar y soltar, duplicar y eliminar. Los trece ajustes preestablecidos integrados P1–P13 permanecen intactos al pulsar el botón de eliminar.

Regla práctica: Si cambias algo en un ajuste preestablecido que vas a querer más veces — Sky-Dome activo, mayor peso SSIM para una clase de escena concreta, distinto número de iteraciones —, guarda la variante como tu propio ajuste preestablecido. Así sabes de inmediato en el siguiente run que se trata de una configuración que se desvía del estándar.

CAPÍTULO

Capítulo 8 — Formatos de exportación

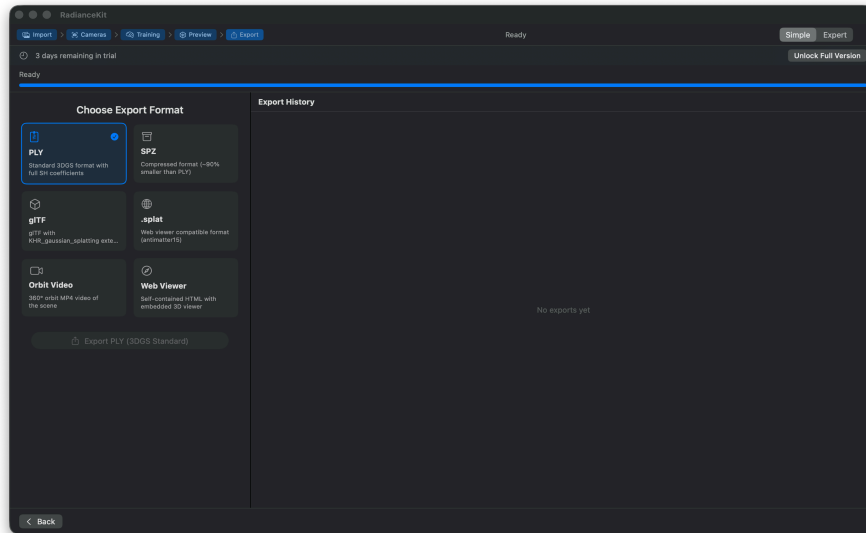


Figura 29: Selección de formato de exportación en el modo principiante — seis tarjetas de formato

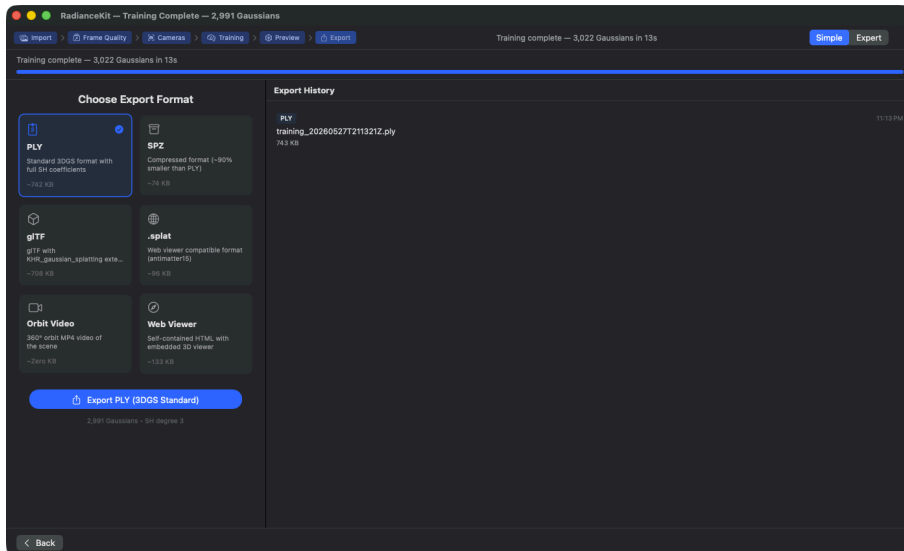


Figura 30: Cuadrícula de formatos de exportación tras un entrenamiento de 5K iter sobre un ramo de flores — las seis tarjetas con cálculo dinámico de tamaño (PLY 742 KB seleccionado, SPZ 74 KB, glTF 708 KB, .splat 96 KB, Orbit Video ~Zero KB, Web Viewer 133 KB), historial de exportación a la derecha con el PLY ya guardado

Lo que se ve en la imagen (2 991 Gaussianos, grado SH 3, ramo sintético de Blender libre de IP como conjunto de prueba): Las cifras de tamaño bajo cada tarjeta de formato se calculan en vivo a partir del número actual de Gaussianos y el overhead del formato — no están fijadas. Con 2 991 Gaussianos (grado SH 3) obtienes 742 KB de PLY, 74 KB de SPZ (factor $\sim 10\times$ más pequeño por la cuantización), 708 KB de glTF (con extensión `KHR_gaussian_splatting`, por eso casi equivalente al PLY), 96 KB de `.splat` (formato comprimido de 24 bytes por Gaussiano). Orbit Video muestra «~Zero KB» porque el tamaño solo se conoce tras codificar el MP4. Web Viewer (133 KB) empaqueta un archivo HTML autónomo con visor WebGL embebido y datos de `splat` comprimidos — más grande que el `.splat` puro por el overhead del visor. El historial de exportación a la derecha enumera la exportación PLY ya completada («`training_20260527T211321Z.ply`, 743 KB, 23:13») con la insignia de formato y la acción Reveal in Finder.

Un entrenamiento terminado entrega una nube de Gaussianos — una colección de unos pocos cientos de miles a millones de distribuciones Gaussianas 3D que en conjunto reconstruyen la escena. RadianceKit conoce diez formas de escribir esa nube a disco. Seis son formatos de datos 3D puros (PLY, Compressed PLY, SPZ, SOG, glTF, `.splat`), uno empaqueta la nube junto con un visor HTML listo (Web Viewer), uno renderiza un archivo MP4 desde una trayectoria de cámara orbital (Orbit Video) y dos no exportan contenido Gaussiano sino solo el resultado de SfM (poses de cámara y nube de puntos gruesa) para reutilizar en otras pipelines de entrenamiento (`transforms.json` + COLMAP Workspace).

Qué formato es el adecuado en cada caso depende del objetivo. Para archivar todos los datos sin pérdida de calidad usas PLY. Para visores web en tu propia página, `.splat` o el Web Viewer integrado suelen bastar. Si el archivo tiene que ser mínimo, vale la pena SPZ o SOG. Para reutilizar el resultado de SfM en Nerfstudio, Postshot o Brush, `transforms.json` y el COLMAP Workspace son las vías correctas.

Todas las funciones de exportación están en el menú «Export», así como en el modo principiante en la última etapa del asistente. La mayoría de los formatos son totalmente compatibles con el sandbox y funcionan en la versión de App Store. Solo SOG requiere un binario externo (`cwebp`) que no está necesariamente presente en el build de App Store — detalles en E4.

I E1 — PLY (.ply)

DÓNDE

Barra de menús → Export → 3D Formats → Export PLY... (⌘E). Modo principiante: paso del asistente Export → tarjeta de formato «PLY». **Tamaño:** típicamente 100 % (valor de referencia). **Compatible con:** SuperSplat, PolyCam, todos los visores 3DGS.

TÉCNICO

PLY es el formato de almacenamiento canónico para 3D Gaussian Splatting. RadianceKit escribe un archivo binario little-endian con el layout de propiedades 3DGS estandarizado: por Gaussiano, posición de tres componentes, tres normales siempre a cero, tres coeficientes SH DC (`f_dc_0..2`) para el color RGB base, seguidos de hasta 45 coeficientes SH adicionales (`f_rest_0..44`) en la disposición canal-mayor transpuesta definida por el paper de Kerbl 2023 (primero todos los coeficientes del canal R, luego todos los de G y luego todos los de B), seguido por la opacidad logit (valores brutos previos al sigmoide), tres escalas en espacio logarítmico y una rotación cuaternión wxyz. El grado SH máximo exportado se acota al mínimo entre el deseo del usuario y el grado realmente aprendido; el predeterminado es 3 (45 coeficientes rest). Antes de escribir, el tamaño del payload se calcula en entero de 64 bits para detectar overflow en nubes extremadamente grandes. El archivo se escribe de forma atómica, lo que en nubes grandes ocupa brevemente el doble de espacio en disco.

EN POCAS PALABRAS

Este es el «archivo original». El archivo más grande, la mayor compatibilidad, sin pérdidas. Si no sabes qué formato escoger, escoge PLY — se abre en casi cualquier herramienta de 3DGS. Para 1 millón de Gaussianos, eso son entre 200 y 800 MB según el grado SH. Si el archivo se hace demasiado grande, mira E2 (PLY comprimido) o E3 (SPZ).

I E2 — PLY comprimido (.ply)

DÓNDE

Barra de menús → Export → 3D Formats → Export Compressed PLY... Modo principiante: tarjeta de formato «Compressed PLY». **Tamaño:** aprox. 10–20 % vs. PLY (compresión de 5 a 10 veces). **Compatible con:** SuperSplat, motor PlayCanvas, visores basados en web.

TÉCNICO

La variante PlayCanvas del formato PLY con cuantización por chunks. Los Gaussianos se agrupan en chunks de 256. Por chunk se almacenan en la cabecera, por separado, los límites min/max para posición, escala y color; los Gaussianos individuales referencian sus valores relativos a esos límites y se comprimen a 32 bits cada uno: posición y escala con empaquetado 11-10-11 bits, rotación como cuaternión «Smallest-Three» 2-10-10-10 bits, color como RGBA 8-8-8-8. Los coeficientes SH superiores se cuantizan con solo 8 bits por componente (`shCoeffCount * 3 uchar` por Gaussiano). El formato en sí sigue siendo PLY con cabecera ASCII y por tanto en principio validable con herramientas de PLY, pero las propiedades de vértice están declaradas como campos `uint`. El grado SH es por defecto 0 (sin coeficientes rest) para maximizar la compresión — se pueden seleccionar grados SH superiores de forma explícita.

EN POCAS PALABRAS

La variante de PLY que ahorra espacio. La misma compatibilidad con engines que el PLY normal, pero 5 a 10 veces más pequeño. SuperSplat y PlayCanvas lo leen de forma nativa. Para despliegue web casi siempre es mejor que el PLY normal. La pérdida de calidad por la cuantización normalmente no se nota visualmente, mientras la escena no contenga detalles de muy alta frecuencia.

| E3 — SPZ (.spz)

DÓNDE

Barra de menús → Export → 3D Formats → Export SPZ... Modo principiante: tarjeta de formato «SPZ».

Tamaño: aprox. 10 % vs. PLY (90 % más pequeño).

Compatible con: Niantic Scaniverse, Niantic Spatial Fields, MetalSplatter.

TÉCNICO

El formato SPZ v2 de Niantic. Las posiciones se empaquetan como fixed-point de 24 bits (lo que da una resolución de aprox. 0,25 mm), las escalas como cuantización de 8 bits en espacio logarítmico, las rotaciones como Smallest-Three de 8 bits (en v2 solo se almacena xyz, w se deriva en el decoder a partir de la norma del cuaternión), las opacidades como valores sigmoideados de 8 bits. El SH DC se almacena con una fórmula de empaquetado específica de SPZ ($dc_raw * 0.15 * 255 + 0.5 * 255$), las bandas SH superiores con 5 bits (banda 1) o 4 bits (banda 2-3) por coeficiente. Todo el blob binario empaquetado se comprime después con gzip estándar (RFC 1952), lo que da un formato contenedor gzipped con bytes mágicos `1f 8b`. RadianceKit invoca para ello el `gzip` del sistema, porque la API `zlib` integrada de Apple genera un framing propio de Apple que no sería compatible con los lectores SPZ de Spatial Fields o MetalSplatter. El `gzip` del sistema también sigue siendo lanzable dentro del sandbox de macOS.

EN POCAS PALABRAS

El archivo estándar más pequeño. Si conoces Scaniverse de Niantic — ese es el formato que usa la app. Muy pequeño, muy amable con la carga para apps móviles. Utilizable directamente en el propio visor en la nube de Niantic (Spatial Fields). Cerca de un 90 % más pequeño que un PLY con los mismos datos, mientras que en la mayoría de las escenas es ópticamente apenas distinguible.

I E4 — SOG (.sog)

DÓNDE

Barra de menús → Export → 3D Formats → Export SOG.... Modo principiante: tarjeta de formato «SOG». **Tamaño:** aprox. 5–6 % vs. PLY (compresión de 15 a 20 veces — la opción más pequeña). **Compatible con:** motor PlayCanvas, editor SuperSplat.

TÉCNICO

«Spatially Ordered Gaussians» — un formato PlayCanvas que almacena la nube lista para GPU en varias imágenes WebP sin pérdida. Primero, todos los Gaussianos se ordenan espacialmente mediante código Morton 3D (Z-order de 30 bits, 10 bits por eje), lo que da más adelante a las imágenes localidad de caché en el renderer. Luego las posiciones se cuantizan a valores de 16 bits con transformación logarítmica simétrica (para un mejor rango dinámico) y se dividen en dos imágenes RGBA (`means_l.webp` para los 8 bits bajos, `means_u.webp` para los altos). Las rotaciones se codifican como Smallest-Three con 3×8 bits más modo de 2 bits en una imagen RGBA (el modo aterriza en alfa como `252 + largest`). Las escalas y el SH DC se cuantizan, cada uno, con un codebook de 256 entradas (distribuido por percentiles sobre todos los valores); los índices acaban en `scales.webp` y `sh0.webp` . Las cinco imágenes más un `meta.json` con codebooks y bounds se empaquetan en un archivo ZIP (encoder propio porque el sandbox bloquea el `zip` del sistema) y se guardan con la extensión `.sog` .

Aviso de sandbox: SOG es la única opción de formato que requiere un binario externo. La etapa del encoder WebP llama a `cwebp` desde `/usr/local/bin/cwebp` o `/opt/homebrew/bin/cwebp`. Si no se encuentra el binario `cwebp`, el código recurre a la codificación PNG cruda — pero: **el fallback PNG no funciona en SuperSplat**. En la versión de App Store, evalúa la disponibilidad según la variante del build; en la variante de desarrollador, `cwebp` debe instalarse mediante Homebrew (`brew install webp`).

EN POCAS PALABRAS

El formato 3DGS más pequeño en conjunto, notablemente más pequeño que SPZ. Pero: necesita la herramienta `cwebp` en tu Mac, porque RadianceKit por sí misma no puede generar todos los formatos de imagen. Instálala una vez con Homebrew y todo funciona. En la versión de App Store posiblemente no esté plenamente funcional — si en la exportación sale PNG en lugar de WebP, no podrás abrir el archivo directamente en SuperSplat. Quien trabaje sin Homebrew, que use SPZ (E3) en su lugar.

I E5 — glTF (.glb)

DÓNDE

Barra de menús → Export → 3D Formats → Export glTF... Modo principiante: tarjeta de formato «glTF». **Tamaño:** comparable al PLY. **Compatible con:** visores glTF con extensión KHR_gaussian_splatting (borrador de estándar Khronos).

TÉCNICO

Escribe un archivo binario `.glb` autocontenido (sin archivo bin separado adjunto) conforme a la especificación de la extensión KHR_gaussian_splatting. Las posiciones se almacenan como datos de vértice `POSITION` regulares de glTF (float3); todos los demás atributos (rotación como float4, escala como float3, opacidad como float, coeficientes SH como float3 × shCoeffCount) están en atributos de vértice adicionales y se referencian mediante la extensión. Importante: glTF usa un sistema de coordenadas right-handed Y-up, COLMAP/3DGS trabaja en Y-down/Z-forward. Por eso, el exportador aplica una rotación de 180 grados alrededor del eje X — las posiciones se reescriben como $(x, -y, -z)$, los cuaterniones se ajustan a $(w, x, -y, -z)$. Esto da una visualización geoméricamente correcta y con handedness (no espejada) en los visores glTF. Los chunks JSON y binarios se rellenan con padding a alineación de 4 bytes como exige el estándar GLB.

EN POCAS PALABRAS

El formato estándar oficial de Khronos para datos 3D, en la nueva extensión para Gaussian Splats. Ventaja: glTF está extendido en todos los grandes motores 3D (Babylon.js, Three.js, Unity, Unreal). Desventaja: la extensión todavía está en fase de draft en 2026, muchos visores aún no la pueden leer. Tiene sentido sobre todo si integras datos de splat en una pipeline glTF existente o escribes un visor que ya es capaz de glTF.

E6 — Splat (.splat)

DÓNDE

Barra de menú → Export → 3D Formats → Export .splat... Modo principiante: tarjeta de formato «.splat». **Tamaño:** exactamente 32 bytes por Gaussiano. **Compatible con:** gsplat.js, visores basados en web (referencia antimatter15), la mayoría de las demos 3DGS en navegador.

TÉCNICO

El formato `.splat` de antimatter15 — 32 bytes por Gaussiano, sin cabecera, sin indirección. Layout por entrada: 3 × float32 posición (coordenadas mundo), 3 × float32 escala (transformada con exp desde el espacio logarítmico del buffer interno), 4 × uint8 color RGBA (coeficiente SH DC escalado con `SH_C0 = 0.282...` y clamped a [0,255]), 4 × uint8 cuaternión (w,x,y,z, normalizado y codificado en el rango de bytes como `128 + 128*q`). Solo se almacena el SH DC — las bandas SH superiores se descartan. Eso hace al formato extremadamente compacto, pero cuesta los cambios de color dependientes de la vista que se dan con reflejos o brillos especulares. El orden de escritura es exactamente el orden de índice de la nube (sin ordenación espacial); los visores web como `gsplat.js` renderizan basándose en ese supuesto.

EN POCAS PALABRAS

El formato de elección si quieres mostrar el splat en tu propio visor web con `gsplat.js`. Muy compacto (32 bytes/Gaussiano), pero sin grado SH superior — así que sin reflejos brillantes ni cambios sutiles de color según el ángulo de visión. Para la mayoría de las aplicaciones web no es un problema, porque el color DC es totalmente suficiente y la falta de dependencia de la vista apenas se nota.

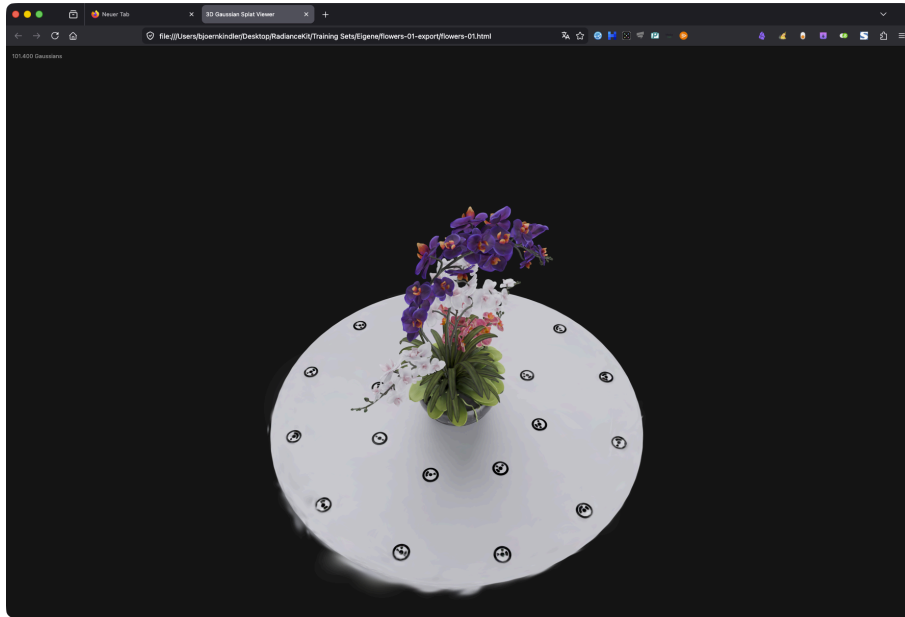


Figura 31: Web Viewer abierto en Firefox — el splat del ramo renderizado con esferas de marcadores de cámara alrededor, barra de pestañas del navegador visible arriba, sin necesidad de configurar CDN/ servidor. `flowers-01.html` autónomo abierto directamente desde el Finder con doble clic en el navegador por defecto — el programa WebGL2 embebido renderiza la nube Gaussiana de inmediato, sin red ni servidor. Los marcadores negros alrededor del ramo son las cámaras de entrenamiento, opcionalmente conmutables. Arrastrar con el ratón rota, scroll hace zoom.

| E7 — Visor web (.html)

DÓNDE

Barra de menús → Export → Media → Export Web Viewer... Modo principiante: tarjeta de formato «Web Viewer». **Tamaño:** datos de splat codificados en base64 ($\approx 4/3$ de overhead) + aprox. 5 KB de envoltorio HTML/JS. **Compatible con:** cualquier navegador moderno con WebGL2 (todos los escritorios, iOS 15+, Android 5+).

TÉCNICO

Empaqueta la nube Gaussiana junto con un renderer WebGL2 escrito totalmente inline en un único archivo `.html`. No hay dependencias de CDN, ni WASM, ni segundo archivo. La nube se codifica internamente primero como binario `.splat` (misma lógica de 32 bytes que E6), después se incrusta en base64 y luego se decodifica con `atob` en el navegador. El renderer integrado hace su propia ordenación WebGL2, controles de órbita con ratón y ordenación por CPU por frame; todo el código JS (shader, matemáticas, bucle) es visible en el HTML de salida. La convención de ejes en la frontera almacenamiento-renderer es exactamente la misma que en E5: posición `(x, -y, -z)`, cuaternión `(w, x, -y, -z)`. Opcionalmente se puede mostrar un overlay de branding (conmutador del free-tier). Como todo es inline, el archivo también funciona directamente desde el protocolo `file://` — no se necesita servidor web local para probarlo.

EN POCAS PALABRAS

Un único archivo HTML que puedes enviar a alguien por correo o embeber en una web. Doble clic en el Finder y el navegador muestra tu escena con rotación con el ratón. No hace falta subirlo a una nube, ni un segundo archivo, ni un servidor. Ideal para presentaciones a clientes, portafolio, adjuntos de correo. Desventaja: el archivo se hace alrededor de un tercio más grande que un `.splat` puro por la codificación en base64 — para escenas muy grandes, por tanto, merece la pena un hosting separado del archivo `.splat` junto con un visor estándar.

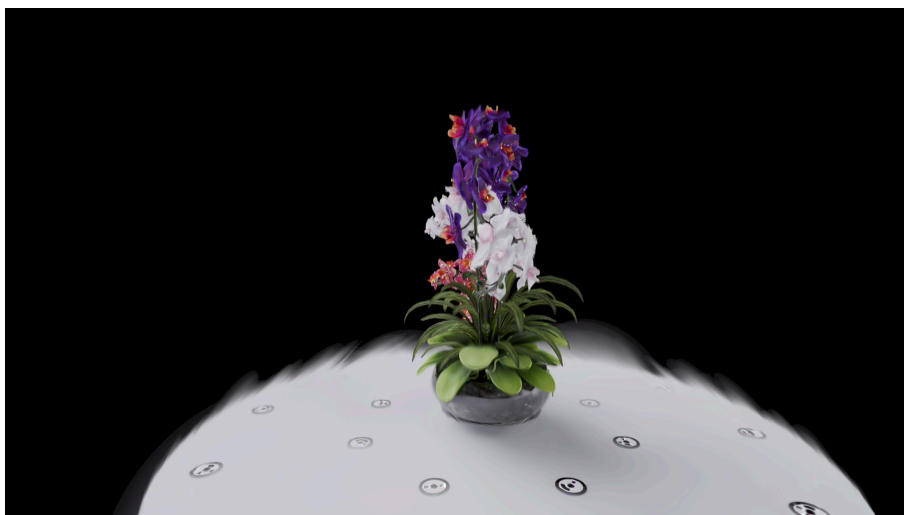


Figura 32: Único fotograma extraído de flowers-01.mp4 — el ramo en renderizado de perfil, plataforma blanca con marcadores de cámara visibles, fondo negro (fondo de vista predefinido, cambiable en Ajustes). La cámara orbita la escena en una trayectoria paramétrica (elevación + distancia fijas, yaw rota), duración típicamente 6–10 segundos a 30 o 60 fps. Resolución del fotograma escalable de 480p a 8K mediante VideoPreset.

| E8 — Vídeo orbital (.mp4/.mov)

DÓNDE

Barra de menús → Viewport → Record Turntable Video O bien Barra de menús → Export → Media → Export Orbit Video.... Modo principiante: tarjeta de formato «Orbit Video» con deslizador de duración 3–30 s. **Tamaño:** depende de duración, resolución, bitrate. **Compatible con:** todas las plataformas (H.264 y HEVC son estándar de Apple).

TÉCNICO

Renderiza la nube Gaussiana a lo largo de una trayectoria de cámara orbital paramétrica y codifica cada fotograma mediante AVAssetWriter en un archivo MP4 o MOV. La configuración de la órbita controla la velocidad de rotación (vueltas), distancia, elevación, FOV, duración y factor de ease-in/out. La exportación de vídeo orbital pasa por el propio ForwardPass de RadianceKit con evaluación SH completa — idéntica píxel a píxel a la vista del visor en la app (WYSIWYG). Por fotograma, la matriz de ajuste mundial (calculada por el renderer para rotar las coordenadas internas al mundo orbital Y-up) se multiplica con la cámara, tras lo cual se aplica un volteo de convención de cámara (camFlip: órbita Y-up → COLMAP Y-down). El render target offscreen se extrae mediante IOSurface a un CVPixelBuffer para el encoder. El encoder soporta H.264 y HEVC, bitrate y resolución configurables de 480p a 8K. Antes del primer fotograma, el renderer espera 200 ms para que se complete la ordenación inicial de splats. Esta exportación está limitada por GPU — a 8K y millones de Gaussianos, el tiempo de renderizado por fotograma es de varios segundos, así que son posibles tiempos totales de renderizado de 10–30 minutos para un vídeo de 6 s.

EN POCAS PALABRAS

Un archivo MP4 terminado con una rotación alrededor de tu escena. Perfecto para redes sociales, marketing, presentaciones. Puedes ajustar duración (3–30 segundos), sentido y velocidad de rotación. El archivo se puede embeber directamente en YouTube, Instagram, PowerPoint y donde sea. A veces es lento porque la app tiene que renderizar por completo cada fotograma — para un vídeo 8K puedes planificar de cinco a treinta minutos, según el número de Gaussianos.

I E9 — SfM Transforms (transforms.json)

DÓNDE

Barra de menús → Export → Photogrammetry → Export SfM (transforms.json).... **Tamaño:** típicamente 1–10 KB (solo poses + intrínsecos, sin imágenes, sin Gaussianos). **Compatible con:** nerfstudio, Brush, gsplat, OpenSplat, Meshroom, todos los entrenadores 3DGS feed-forward modernos.

TÉCNICO

Escribe el formato `transforms.json` de nerfstudio con una lista de poses de cámara más intrínsecos compartidos. Por cámara, la matriz de vista (interna de RadianceKit: World-to-Camera en convención COLMAP) se invierte, tras lo cual los vectores base Y y Z local de cámara se espejan para pasar a la convención nerfstudio (estilo OpenGL, la cámara mira a lo largo de $-Z$, $+Y$ es arriba). La matriz 4×4 final aterriza como array anidado row-major de doubles en el campo `transform_matrix` de cada frame. Los intrínsecos se almacenan en el nivel superior (focal x/y, punto principal x/y, ancho/alto de imagen, `camera_model = "OPENCV"`, más los coeficientes de distorsión `k1, k2, p1, p2`) — excepto cuando el exportador detecta varios conjuntos de intrínsecos distintos, caso en que se escriben por frame. Las rutas de imagen se escriben como `images/<filename>` relativas al archivo JSON; el usuario debe crear una carpeta hermana `images/` con las fotos de entrenamiento.

EN POCAS PALABRAS

Este archivo JSON describe, para cada foto, dónde estaba la cámara y a dónde miraba. El archivo por sí solo es pequeño e inútil — se usa junto con las imágenes originales en una carpeta. Nerfstudio, Brush y algunos otros entrenadores leen exactamente este formato, y con él puedes pasar tus resultados de SfM de RadianceKit a otra herramienta sin que allí haya que recalcular la reconstrucción de la cámara. Ahorra horas en escenas grandes.

I E10 — COLMAP Workspace (sparse/0/)



DÓNDE

Barra de menús → Export → Photogrammetry → Export SfM (COLMAP Workspace).... **Tamaño:** tres archivos binarios juntos típicamente 4–8 MB — `points3D.bin` domina (una línea por punto 3D de la nube sparse), `images.bin` y `cameras.bin` están cada uno notablemente por debajo de 100 KB.

Compatible con: el propio COLMAP, Nerfstudio, Postshot, Meshroom, todas las herramientas que esperan un directorio `sparse/` de COLMAP.



TÉCNICO

Escribe el layout estándar `sparse/0/` de COLMAP con tres archivos binarios: `cameras.bin`, `images.bin`, `points3D.bin`. La referencia de formato es la documentación oficial de COLMAP. `cameras.bin` contiene la lista de intrínsecos deduplicados (las cámaras con intrínsecos idénticos + tamaño de imagen se fusionan en una sola entrada); el modelo de cámara usado es `OPENCV` (modelo 4), con `fx/fy/cx/cy` más los cuatro coeficientes de distorsión `k1/k2/p1/p2`. `images.bin` lista por imagen la pose como cuaternión `wxyz` más traslación, seguidos del ID de cámara y el nombre de archivo; no se almacenan correspondencias 2D-3D. `points3D.bin` contiene la nube de puntos SfM con posición, color (RGB 0-255) y valores predeterminados para reproyección y longitud de track. Todo se escribe en little-endian. La reimportación en RadianceKit funciona a través del menú File → «Import COLMAP/Metashape Workspace...» (véase Q3 en el capítulo de backends SfM).



EN POCAS PALABRAS

El formato oficial de COLMAP. Si quieres continuar tu entrenamiento en Postshot, Nerfstudio u otro software compatible con COLMAP, esta es la vía. Tres archivos pequeños más tus imágenes originales, y el programa de destino lo acepta como si el propio COLMAP hubiera sido el programa origen. Más programas lo entienden que el formato `transforms.json` (E9), al mismo tiempo algo menos cómodo porque es binario en lugar de basado en texto.

¿Qué formato en cada caso?

Objetivo	Formato
Visor web en tu propia página	E7 Web Viewer (.html)
Visor web con <code>gsplat.js</code>	E6 Splat (.splat)
Reutilización de pipeline en Post-shot / Nerfstudio	E9 transforms.json + E10 COLMAP Workspace
Edición en SuperSplat	E1 PLY o E2 Compressed PLY
Niantic Scaniverse / Spatial Fields	E3 SPZ
Máxima compresión	E4 SOG (cwebp requerido)
Vídeo de marketing/redes	E8 Orbit Video

Comparativa rápida

Forma- to	Extensión	Sandbox	Tamaño (1M Gauss)	Mejor uso
E1 PLY	<code>.ply</code>	sí	~250 MB	Archivo, máxima compatibilidad
E2 Com- pressed PLY	<code>.ply</code>	sí	~40 MB	Web + SuperSplat
E3 SPZ	<code>.spz</code>	sí (spawn de gzip)	~40 MB	Niantic + móvil
E4 SOG	<code>.sog</code>	condi- cional (cwebp)	~20 MB	Máxima compresión
E5 glTF	<code>.glb</code>	sí	~250 MB	Pipeline Khronos
E6 Splat	<code>.splat</code>	sí	~32 MB	Visor web gsplat.js
E7 Web Viewer	<code>.html</code>	sí	~45 MB	Archivo au- tónomo pa- ra navega- dor
E8 Vídeo orbital	<code>.mp4 / .mov</code>	sí	variable	Redes/mar- keting
E9 SfM Trans- forms	<code>.json</code>	sí	~5 KB	Traspaso de poses
E10 COLMAP Works- pace	Directorio	sí	~4–8 MB	Traspaso de poses bina- rio

La columna de tamaño son valores de referencia aproximados para 1 millón de Gaussianos con grado SH 3. Los valores reales varían según la compresibilidad de la escena; el grado SH 0 reduce PLY/glTF por un factor de 4.

CAPÍTULO

Capítulo 9 — Backends de SfM

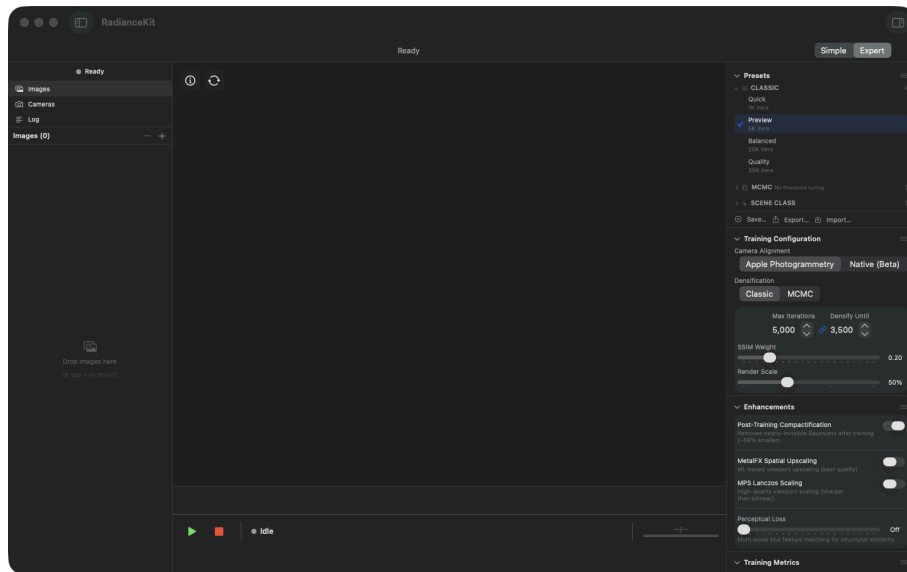


Figura 33: Modo Experto con selector de Camera Alignment en el Inspector (Apple Photogrammetry / Native (Beta))

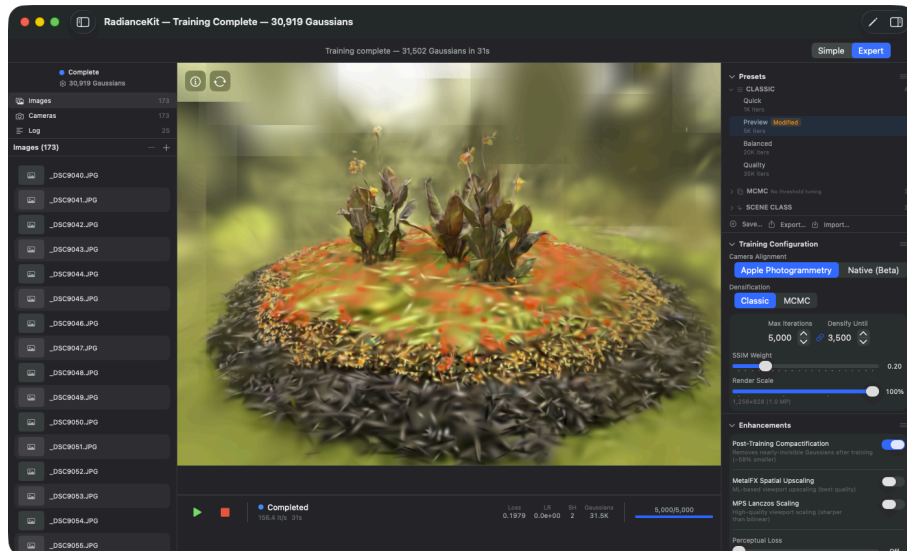


Figura 34: Inspector con Native (Beta) activo — segunda opción del selector de Camera Alignment seleccionada, el resto de parámetros de configuración del entrenamiento sin cambios

LO QUE SE VE EN LA IMAGEN El selector de Camera Alignment en el Inspector es un control segmentado con dos opciones — Apple Photogrammetry (predeterminado para builds

de App Store, totalmente compatible con el sandbox) y Native (Beta) (el backend propio de la pipeline FAST+BRIEF+GLOMAP de RadianceKit, desarrollado durante la Fase 3.8/3.9, estado 2026-05). Native (Beta) está validado únicamente en escenas orbitales y es más rápido que Apple Photogrammetry con $\geq 1\ 000$ fotogramas, pero todavía no alcanza la puerta de calidad de la Fase 3 §5 ($\text{finalLoss} \leq 0.0115$) — de ahí la etiqueta Beta. Los resultados de SfM externos de Metashape, COLMAP u otro software de fotogrametría se pueden importar adicionalmente mediante el menú File (Q3 formato de texto COLMAP, Q6 Workspace Import) — el selector no cambia, pero las poses importadas reemplazan al resultado de SfM.

SfM significa **Structure from Motion**. A partir de un conjunto de fotos solapadas, el software reconstruye para cada imagen la posición y la dirección de visión de la cámara en un sistema de coordenadas 3D común. En el proceso se genera una nube de puntos 3D gruesa, que inicializa el entrenamiento con Gaussian Splatting. El resultado de SfM es la entrada para el entrenamiento propiamente dicho y determina de forma decisiva la calidad final de la imagen.

RadianceKit ofrece cinco vías de SfM: dos backends integrados en la app (Q1 Apple Photogrammetry, Q4/Q5 Native), dos vías de importación desde herramientas externas (Q3 formato de texto COLMAP, Q6 Workspace Import binario) y Q2 COLMAP binario, que solo está disponible en builds de desarrollador fuera del App Store.Cuál es la adecuada depende del tipo de escena (órbita alrededor de un objeto, espacio interior, vuelo de dron) y de si un software externo ya proporciona una reconstrucción.

I Q1 — Apple Photogrammetry

DÓNDE

Expert View → Inspector → Configuración del entrenamiento → Selector Camera Alignment, entrada «Apple Photogrammetry».

TÉCNICO

Envuelve el framework de fotogrametría integrado de Apple, desarrollado originalmente para Object Capture. Apple extrae internamente las características con una pipeline propietaria (los pasos no están documentados públicamente), las verifica mediante matching multi-vista y resuelve el bundle adjustment en el Apple Silicon Neural Engine + GPU. El backend es totalmente compatible con el App Store (sin binario externo, Sandbox=true, on-device), pero solo entrega poses de cámara más una nube de puntos gruesa — sin métricas de diagnóstico como longitud de track o error de reproyección. Escala según la recomendación de Apple hasta unos pocos cientos de imágenes. Con más de ~500 fotografías en vuelos lineales de dron o grandes escenas exteriores se han observado de forma reproducible bloqueos o descarte silencioso de cámaras individuales.

EN POCAS PALABRAS

Esta es la forma más sencilla. Introduces las imágenes y la app calcula. Funciona muy bien para escaneos clásicos de objetos — cuando caminas alrededor de un mueble o una escultura y haces 50–200 fotos. Para vuelos de dron sobre paisajes o con muchísimas imágenes (más de 500), el método de Apple tiende sin embargo a volverse inestable. Para esas escenas prueba el backend Native (Q4/Q5) o calcula las cámaras en Metashape y cárgalas mediante Workspace Import (Q6).

POWER-USER

Q2 COLMAP binario — lanza el programa externo COLMAP como subprocesso y por eso **no está disponible** en la versión de App Store (sandbox). Solo funciona en builds de desarrollador fuera del App Store. Para la calidad que ofrece COLMAP, en la versión de App Store existe Workspace Import (Q3 o Q6): calcula el SfM en COLMAP o Metashape externamente y carga el resultado.

Q3 — Formato de texto COLMAP (Metashape / ETH3D)

DÓNDE

Menú «File → Import COLMAP / Metashape Workspace...» (Cmd+⇧+I) O arrastrar y soltar una carpeta con `sparse/0/cameras.txt`.

TÉCNICO

Lee la exportación de texto COLMAP estandarizada — tres archivos de texto `cameras.txt`, `images.txt`, `points3D.txt` en la subcarpeta `sparse/0/` — y la convierte en el modelo interno de resultado de SfM. Misma definición de formato que la exportación binaria de COLMAP, solo que como ASCII en lugar de binario. Lo exportan Agisoft Metashape, RealityCapture, PolyCam y el benchmark ETH3D exactamente en este layout. El parser comparte la detección de modelo de cámara con el parser binario (todos los modelos comunes: SIMPLE_PINHOLE, PINHOLE, OPENCV, OPENCV_FISHEYE, FULL_OPENCV). Robusto frente a líneas de comentario y líneas vacías. Escala en las pruebas hasta ~1 400 cámaras (ETH3D Tunnel) sin problemas.

EN POCAS PALABRAS

Si ya has trabajado con Metashape, RealityCapture u otro software comercial de foto-3D y has exportado el resultado — puedes cargar esa exportación directamente en RadianceKit, sin que la app tenga que recalcular. Esto ahorra horas de espera. Carga simplemente toda la carpeta mediante el menú File o arrástrala a la ventana.

I Q4 — Native SfM (incremental)

DÓNDE

Expert View → Inspector → Configuración del entrenamiento → Selector Camera Alignment, entrada «Native (Beta)». Incremental es el modo predeterminado de este backend — no hay un selector de mapper separado en el Inspector. Por CLI, el modo se puede establecer explícitamente con `--native-sfm` o `--sfm-mapper incremental`.

TÉCNICO

Implementación propia acelerada por GPU de toda la pipeline de SfM: características FAST+BRIEF O SuperPoint+LightGlue vía CoreML (con `--coreml-features`), seguidas de matching Hamming-KNN, matriz fundamental RANSAC, construcción de tracks, selección del par inicial, two-view bootstrap (F→E más DLT), mapper incremental greedy con registro PnP y triangulación multi-vista, y bundle adjustment final vía Levenberg-Marquardt reducido por Schur con loss Huber y jacobianos analíticos a través de Cholesky solve. Totalmente compatible con App Store: sin binario externo, `Sandbox=true`. Con el detector de colapso R2 entregado en la Fase 3.10: si la app registra menos del 60 % de los fotogramas de entrada o la tasa de puntos por cámara cae por debajo de 13, conmuta automáticamente al mapper global (Q5). Empíricamente limpio en escenas orbitales o de plato giratorio; en movimientos más generales (vuelo de dron, espacios interiores con geometría compleja) la tasa de éxito es menor — pero el detector atrapa esos casos. Escala con fiabilidad hasta ~200 cámaras, más con un tiempo de ejecución claramente mayor.

EN POCAS PALABRAS

Las virtudes de Apple (compatible con App Store, rápido para órbitas) con valores de diagnóstico adicionales. Funciona especialmente bien cuando caminas alrededor de un motivo como en un Object Capture. Con tomas más complicadas (vuelo de dron o salón), RadianceKit detecta automáticamente que no va a salir bien y cambia al método global. Marcado como «Beta» porque aún está en pruebas — la recomendación estándar sigue siendo Apple Photogrammetry para escaneos sencillos de objetos y Workspace Import (Q3 o Q6) para sets exteriores exigentes.

I Q5 — Native SfM (global)

DÓNDE

Se llama automáticamente cuando el mapper incremental (Q4) dispara el detector de colapso (menos del 60 % de los fotogramas de entrada registrados o tasa de puntos por cámara por debajo de 13). Forzable manualmente solo vía CLI `--sfm-mapper global`. En el Inspector, el método global no es accesible por un selector aparte — la app decide por sí misma cuándo conmutar.

TÉCNICO

Variante global de la pipeline nativa. Primero extracción de características + matching como en Q4, después estimación de pose relativa para todos los pares verificados, seguido de rotation averaging (sincroniza todas las rotaciones de cámara en el sistema de coordenadas mundial) y translation averaging (basado en LSQR sobre una formulación sparse libre de matrices, para evitar overflow de enteros con grandes cantidades de cámaras). Escala en principio a ~5 000 cámaras, en la práctica con calidad degradada por encima de unos pocos cientos — la medición de la puerta de aceptación de la Fase 3.8 §5 sobre K-1351 dio finalLoss 0.07 en lugar de los 0.0115 objetivo. Se trata como «fallback tier»: entra en juego cuando el mapper incremental degenera, pero no se reverifica en cuanto a calidad.

EN POCAS PALABRAS

La vía plan-B para el motor nativo. Se llama automáticamente cuando la vía incremental, más rápida, falla. Entrega un resultado utilizable, pero con escenas muy grandes o difíciles normalmente no es tan precisa como lo que obtienes de Metashape o de una instalación externa de COLMAP. Si Native se convierte en tu flujo estándar, en esos casos vale la pena el rodeo por Workspace Import (Q3 o Q6).

Q6 — Importación de workspace de texto Metashape / COLMAP (Fase Q7)

DÓNDE

Menú File → «Import COLMAP / Metashape Workspace...» (Cmd+⇧+I). Arrastrar y soltar una carpeta con `sparse/0/cameras.{bin,txt}` e `images/`.

TÉCNICO

Detecta automáticamente si una carpeta seleccionada mediante arrastrar y soltar o panel de apertura coincide con uno de los tres layouts de workspace de COLMAP (`sparse/0/`, `sparse/` o raíz) y si la reconstrucción es binaria (`cameras.bin`) o texto (`cameras.txt`). La vía binaria utiliza el parser binario de COLMAP, la vía de texto el loader de ETH3D — ambas producen el mismo modelo de resultado de SfM y el resto de la pipeline (importar imágenes, iniciar entrenamiento MCMC) es agnóstica a la fuente. Las imágenes se abren mediante el sistema de bookmarks del sandbox de la app, con seguridad por ámbito, de modo que la importación también funciona en la versión de App Store. Pensado específicamente para el caso «exportación de Metashape sin recalculación de la reconstrucción». La detección mencionada en la entrada del menú File avisa en el registro de la app si la carpeta elegida no es un workspace reconocible.

EN POCAS PALABRAS

Específicamente la función para usuarios de Metashape. Si tienes licencia de Metashape o Reality-Capture e hiciste allí la reconstrucción de cámara, puedes simplemente arrastrar la carpeta de exportación aquí y empezar a entrenar de inmediato. Ahorra varias horas de tiempo de cálculo en escenas grandes, porque RadianceKit no hace entonces el SfM por su cuenta.

¿Qué backend en cada caso?

Escenario	Backend recomendado
Escaneo de objeto, 50–200 fotos	Q1 Apple Photogrammetry
Exteriores grandes / dron / >500 imágenes	Q6 Workspace Import (calcular en Metashape o COLMAP y luego cargar)
Exportación de Metashape/RealityCapture disponible	Q6 Import (sin SfM necesario)
Conjunto académico de texto COLMAP / ETH3D	Q3 importación de texto COLMAP
Estrictamente compatible con App Store + escena orbital	Q4 Native incremental
Q4 falla	Q5 Native global (automático)
Datos de benchmark ETH3D	Q3 (autotest precomputado)

Comparativa rápida

Bac- kend	App Store	Sand- box	Binario externo	Mejor uso	Máx. ~cá- maras
Q1 Apple PG	✓	✓	—	Objeto or- bital	~300
Q2 COL- MAP bi- nario	✗ (solo build de de- sarrollador)	—	colmap/glomap	Exteriores grandes	~5 000
Q3 im- portación de texto COLMAP	✓	✓	—	Bench rigs	~1 500
Q4 Nati- ve incre- mental	✓	✓	—	Objeto or- bital	~200
Q5 Nati- ve global	✓	✓	—	Fallback de Q4	~1 351
Q6 Works- pace Im- port	✓	✓	—	Reutilizar Metasha- pe	por fuente

CAPÍTULO

Capítulo 10 — Modo principiante

El Modo principiante (en alemán Einsteiger-Modus, Cmd+1) es el flujo guiado para todo el que reconstruye una escena de Gaussian Splatting 3D por primera vez. En lugar de mostrar una barra lateral llena de campos del Inspector, la app te guía por cuatro pasos: primero importar imágenes o un vídeo y elegir un ajuste preestablecido de calidad, después corre el procesado (SfM + Entrenamiento), a continuación la escena terminada se puede inspeccionar en una previsualización 3D, y finalmente se exporta al formato deseado. Una barra de progreso estrecha en el borde superior de la ventana siempre te muestra en qué paso estás.

Comparado con el Modo Experto (Cmd+2), que muestra todos los paneles de control a la vez, el Modo principiante oculta las opciones no usadas, da avisos de validación con pocas o malas imágenes y, en cada paso, solo ofrece los botones que tienen sentido en el estado actual. Puedes cambiar entre Modo principiante y Modo Experto en cualquier momento (Cmd+1 / Cmd+2); todo el estado — imágenes importadas, ajuste preestablecido elegido, entrenamiento en curso, nube de puntos terminada — se conserva y está disponible al instante en el otro modo.

Z1 — Importar (elegir imágenes y ajuste preestablecido)

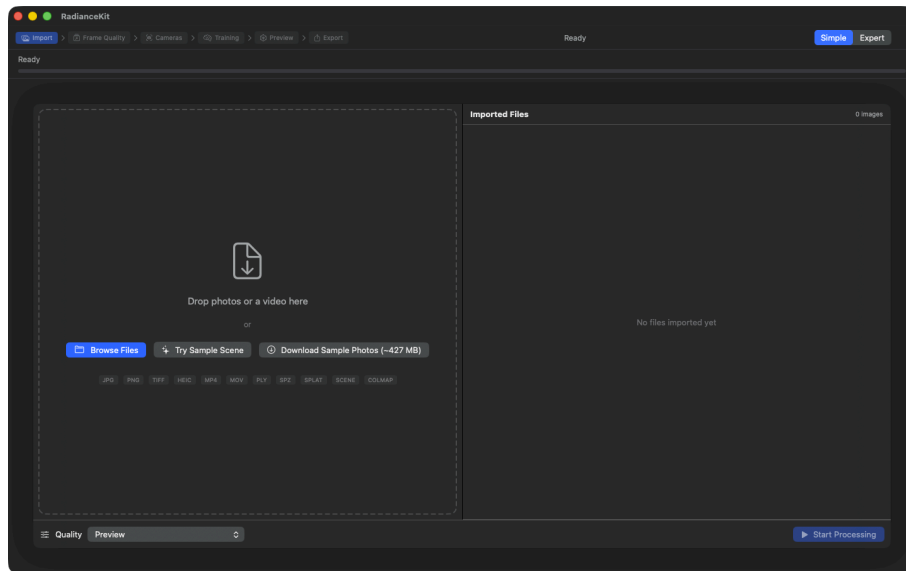


Figura 35: Modo principiante paso 1 — zona de drop vacía antes de la importación de imágenes, ruta de migas arriba (Import → Frame Quality → Cameras → Training → Preview → Export), insignias de formato JPG/PNG/TIFF/HEIC/MP4/MOV/PLY/SPZ/SPLAT/SCENE/COLMAP

LO QUE SE VE EN LA IMAGEN La ruta de migas (Import activo) muestra el flujo de cuatro pasos. Zona de drop a la izquierda con tres CTAs: «Browse Files» (NSOpenPanel), «Try Sample Scene» (demo empaquetada), «Download Sample Photos (~427 MB)» (subconjunto de flores Mip-NeRF360). Las insignias de formato debajo listan todos los tipos de archivo aceptados. A la derecha «Imported Files» con contador «0 images» y estado vacío «No files imported yet». Abajo, selector de Quality (predeterminado: Preview) y «Start Processing» (deshabilitado mientras no haya imágenes).

El primer paso consiste en darle material de imagen a la app. Mediante arrastrar y soltar al gran campo discontinuo del medio, mediante el botón «Browse Files» o haciendo clic en la escena de muestra empaquetada. A la derecha aparece una lista de todas las imágenes importadas con resolución y tamaño de archivo; debajo, en la barra de herramientas flotante, eliges el ajuste preestablecido de calidad y arrancas la pipeline con «Start Processing». Los avisos de validación (rojos con < 3 o < 10 imágenes, naranjas con 10–19) indican si la app espera una reconstrucción razonable o no.

C-01 ProgressIndicator (indicador de paso) **DÓNDE**

Arriba sobre el flujo, siempre visible.

 **TÉCNICO**

Muestra una barra de progreso horizontal a lo largo de toda la pipeline (Frame Quality → SfM → Training) con asignación de etapas: Frame Quality ocupa 0–5 % (Fase 3.11, muy corta), SfM ocupa 0–30 % de la barra, Training 30–100 %. Junto a ella, texto de estado y visualización en porcentaje nombrada por fase («SfM 41 %», «Training 12 500/20 000»), para que los usuarios no lean la aparente regresión «41 % SfM → 25 % Training» como un error — la barra muestra el progreso global de la pipeline, no el de la subetapa. El cálculo de ETA arranca en cuanto se mide suficiente velocidad de entrenamiento (típicamente tras las primeras 100 iteraciones). La misma visualización también se usa en el Modo Experto encima del Inspector.

 **EN POCAS PALABRAS**

La barra estrecha de arriba del todo es tu mapa a través del flujo. Te dice no solo qué está haciendo la app ahora mismo (alineando cámaras, entrenamiento corriendo, ...), sino también lo lejos que ha llegado en conjunto. La división es deliberada para que el cálculo de cámaras ocupe el primer tercio de la barra y el entrenamiento propiamente dicho los dos tercios traseros — de otro modo parecería que el progreso de repente ha vuelto a cero tras el SfM. Así puedes reclinar tranquilo, una mirada a la barra basta para ver la etapa aproximada. El texto al lado te dice si estás ahora en la etapa SfM (p. ej., «SfM 41 %») o en entrenamiento (p. ej., «Training 12 500/20 000»), para que los números no parezcan confusos. Si no se muestra el ETA, el entrenamiento es simplemente aún demasiado joven — la app solo estima cuando ha medido suficiente velocidad.

C-03 DropZoneView (área de arrastrar y soltar) **DÓNDE**

Lado izquierdo del paso Import, gran rectángulo discontinuo con símbolo. Se muestra en Modo principiante con la etiqueta «Drop photos or a video here».

 **TÉCNICO**

Área de drop que rebota brevemente el símbolo y tiñe el fondo en cuanto los elementos arrastrados pasan sobre el campo. Acepta JPG, PNG, TIFF, HEIC, MP4, MOV, PLY, SPZ, .splat, paquetes .radiancecene y directorios. Enrutado de drop por tipo: las imágenes se recopilan y se pasan ordenadas, los vídeos disparan la vía de frame-sampling, los archivos splat abren la previsualización directamente, los paquetes de escena se leen. Los directorios se enumeran y se importan todas las imágenes contenidas. Los bookmarks security-scoped para acceso compatible con sandbox se adquieren y liberan correctamente. Las extensiones no soportadas se muestran como banner de aviso durante 5 segundos.

 **EN POCAS PALABRAS**

El gran campo discontinuo es el control principal del primer paso. Solo arrastra ahí fotos o un vídeo, o toda una carpeta — la app coge todo lo que conoce y ignora el resto. Cuando el campo se vuelve azul y el símbolo rebota brevemente, la app ha reconocido el arrastre. Suelta y la importación arranca de inmediato: las imágenes se mueven a la lista de la derecha, los vídeos disparan automáticamente el paso de frame-sampling, y los archivos `.ply` / `.spz` / `.splat` ya entrenados abren la previsualización directamente. Si un formato no encaja en absoluto (p. ej., PDF o BMP), aparece un breve aviso en el borde superior — la app no se traga en silencio material desconocido.

C-05 Botón Browse Files **DÓNDE**

Dentro de la zona de drop, botón prominente.

 **TÉCNICO**

Botón que abre el diálogo de archivo de macOS con selección múltiple y los tipos de archivo JPG, PNG, TIFF, MP4, MOV, carpetas, así como el formato de escena propio de la app. Las URLs resultantes son security-scoped y se enrutan por las mismas vías de importación que arrastrar y soltar. Cuando el usuario selecciona una carpeta, se enumera recursivamente para imágenes.

 **EN POCAS PALABRAS**

Si arrastrar y soltar no te resulta cómodo, haz clic en este botón y navega en el diálogo de archivo de macOS a tus fotos. Puedes seleccionar varios archivos a la vez (Cmd-clic en las imágenes individuales) o seleccionar toda una carpeta — la app busca entonces recursivamente en la carpeta todos los tipos de imagen soportados. Esto es especialmente práctico cuando tus grabaciones están anidadas en subcarpetas (p. ej., «shoot-day1/», «shoot-day2/») — un clic en la carpeta principal basta. Funcionalmente, el botón hace exactamente lo mismo que arrastrar y soltar; elige la vía que te resulte más cómoda.

C-06 Botón Try Sample Scene DÓNDE

Dentro de la zona de drop, solo visible cuando el bundle de la app contiene la escena de muestra y aún no se han importado imágenes/splats.

 TÉCNICO

Solo aparece cuando (a) un `sample-scene.splat`, `.spz` o `.ply` está presente en el bundle de la app Y (b) no se han importado imágenes/vídeos y no hay nube de puntos. Al hacer clic, carga la nube de puntos terminada (prefiriendo el formato más pequeño — `.splat` ~3 MB, `.spz` ~1,4 MB, fallback `.ply`) y, tras 400 ms, fija valores de cámara codificados en duro a partir de los metadatos originales de la escena de flores para una perspectiva de entrada estéticamente agradable.

 EN POCAS PALABRAS

Si abres la app por primera vez y solo quieres ver qué sale al final — haz clic aquí. Eso abre una escena de flores ya entrenada que puedes rotar y exportar de inmediato, sin que la app tenga que calcular. La cámara está prefijada en una perspectiva de entrada estéticamente agradable, así que ves de inmediato algo bonito. Perfecto para probar los controles 3D y el paso de exportación sin riesgo, antes de abordar tus propias grabaciones. En cuanto importas tus propias imágenes, el botón desaparece automáticamente — solo se muestra mientras el proyecto está completamente vacío.

C-07 Botón Download Sample Photos DÓNDE

Dentro de la zona de drop, junto a «Try Sample Scene»; mismas condiciones de visibilidad.

 TÉCNICO

Dispara una descarga (repo github.com/bkindler/radiancekit-sample-photos) que carga aprox. 427 MB de 960 fotogramas a resolución completa y los mete en la app. Durante la descarga, el botón se deshabilita. El progreso aparece en la barra de progreso superior como «Downloading X %» en su propia etapa, porque esa etapa mantiene su propia escala 0–100 % y no se solapa con la posterior etapa SfM.

 EN POCAS PALABRAS


Como la escena de muestra, solo que con las fotos fuente en lugar del resultado terminado. De este modo puedes dejar correr una vez toda la pipeline tú mismo y ver lo que tardan SfM y entrenamiento realmente en tu Mac. La descarga es grande (aprox. medio DVD = 427 MB), pero solo ocurre una vez — después las fotos están en local y puedes reiniciar la pipeline tantas veces como quieras con distintos ajustes preestablecidos. Mientras corre la descarga, la barra de progreso superior muestra el estado actual de descarga en porcentaje, para que estimes cuándo va a arrancar. Truco: usa una Wi-Fi rápida o red por cable — los 427 MB tardarán si no un rato.

C-09 Selector Quality Presets

DÓNDE

Barra de herramientas flotante inferior del overlay de importación, a la izquierda del botón Start.

TÉCNICO

Control con etiqueta «Quality» agrupa los ajustes preestablecidos disponibles por categoría (Classic / MCMC / Custom). Los ajustes integrados se agrupan por categoría; las cabeceras de sección están codificadas en duro. Los ajustes custom solo visibles cuando existe alguno. Estado bloqueado: los ajustes que no están en la lista gratuita (Quick + Preview) reciben un sufijo «» junto al nombre cuando el usuario no ha comprado; al seleccionar, el selector vuelve a Preview y abre automáticamente el sheet de compra. Al seleccionar, el ajuste se aplica, lo que reemplaza toda la configuración de entrenamiento.

EN POCAS PALABRAS

Aquí eliges con qué precisión y cuánto debe calcular la app. «Quick» y «Preview» se pueden usar sin compra y entregan un resultado inicial en unos minutos — ideal para probar si tus imágenes son razonables en absoluto. «Balanced» y «Quality» necesitan la versión completa y entregan modelos notablemente más limpios, pero tardan horas en lugar de minutos. MCMC es una estrategia distinta que se las arregla con menos splats Gaussianos — bueno si quieres exportar el modelo de forma compacta después o ponerlo en la web. Reconoces los ajustes premium por el pequeño candado junto al nombre; si tocas uno sin licencia, el selector vuelve a Preview y el sheet de compra se abre automáticamente. Regla práctica: empieza siempre con Preview, mira el resultado y después decide si vale la pena un run más largo.

C-10 Botón Start Processing

DÓNDE

Barra de herramientas flotante inferior del overlay de importación, a la derecha del selector de ajustes.

TÉCNICO

Botón que se mantiene atenuado mientras no se hayan importado imágenes ni un vídeo. Al hacer clic, arranca la pipeline y cambia la máquina de etapas en el orden Frame Quality → SfM → Training. El propio botón no tiene más estado; un procesado en curso aparece en su lugar como pantalla de procesado separada.

EN POCAS PALABRAS

El botón «Vamos». Mientras está atenuado, faltan imágenes de entrada o un vídeo. Una vez que has arrastrado fotos, se vuelve activo y haces clic para iniciar SfM y entrenamiento en secuencia. A partir de ahí, la app se hace cargo de todo el flujo y aterriza automáticamente en la pantalla de procesado (Z2). No tienes que hacer clic en nada más — solo tras el final del entrenamiento la app cambia de nuevo a la previusualización (Z3). Si cambias de idea, también puedes cancelar en cualquier momento después con Cancel.

C-11 Deslizador de muestreo de vídeo **DÓNDE**

Lista de imágenes a la derecha, visible solo cuando se ha importado un vídeo (en lugar de imágenes).

 **TÉCNICO**

Deslizador 0,5 fps – 30 fps en pasos de 0,5. Al cambiar, la densidad de fotogramas se actualiza y, además, se calcula el número de fotogramas objetivo (al menos 10) a partir de densidad y duración del vídeo. El deslizador queda fuera de la lista de imágenes, porque los ítems de lista bloquearían eventos de ratón de los deslizadores. Bajo el deslizador se muestran los fotogramas objetivo calculados («247 frames») y la duración del vídeo («1m23s video»). El tooltip avisa: «Doubling the density doubles the number of frames and increases SfM time by ~100%.»

 **EN POCAS PALABRAS**

Si has importado un vídeo en lugar de fotos, este deslizador decide cuántas imágenes individuales debe extraer la app del vídeo. Más imágenes = mejor calidad, pero linealmente más tiempo de cálculo. Para un vídeo orbital de 30 segundos, 5 fps (150 imágenes) es un buen comienzo; para grabaciones de 1 minuto, 3 fps suele ser totalmente suficiente. Bajo el deslizador, la app muestra en vivo cuántos fotogramas resultan con el ajuste actual — así ves de inmediato si das en el rango razonable de unas 100–300 imágenes. Si el resultado queda pobre, mueve el deslizador a la derecha e inténtalo de nuevo; pero duplicar la frecuencia de fotogramas también dobla aproximadamente la duración del SfM.

C-12 Botón Clear All **DÓNDE**

Lista de imágenes a la derecha, abajo a la derecha; visible solo cuando se han importado imágenes.

 **TÉCNICO**

Botón rojo. El clic abre un diálogo de confirmación con título «Clear all imported files?» y mensaje «N images will be removed.». La confirmación borra todas las imágenes/vídeos importados, los directores de staging, la nube de puntos, el estado de entrenamiento, el resultado de SfM y todos los cachés; la etapa salta de vuelta a Import. En Cancel, todo se conserva. El diálogo se configura como vía predeterminada no destructiva (botón destructivo marcado en rojo).

 **EN POCAS PALABRAS**

Si quieres empezar de cero del todo, haz clic aquí. El aviso de confirmación aparece porque el borrado descarta todas las importaciones actuales, incluidas cualquier cámara ya calculada y resultados de entrenamiento — no puedes deshacerlo. Útil cuando quieres reemplazar por completo el material de imagen elegido o deshacerte de un proyecto viejo antes de iniciar uno nuevo. Nota: eliminar una sola imagen se hace desde la lista de la derecha (véase el siguiente ítem), no con este botón. Tus archivos en disco no se borran en el proceso — la app solo olvida sus referencias.

C-13 Lista de archivos ForEach (eliminación individual de imagen) **DÓNDE**

Lista de imágenes a la derecha, cada entrada.

 **TÉCNICO**

Lista sobre las imágenes importadas con swipe-to-delete. Por imagen una fila con icono, nombre de archivo, resolución («1920 × 1080») y tamaño de archivo (formateado KB/MB). La resolución viene de una caché de metadatos rellena de forma asíncrona desde las cabeceras de imagen, para que la UI no bloquee. La acción de borrado ofrece el borrado por swipe típico de macOS (swipe a la izquierda en una fila con trackpad), así como borrado por teclado para la fila seleccionada. Nota: la vía extendida de borrado de imagen con botón menos explícito, backspace y Cmd-Z para deshacer se añadió *solo en Modo Experto* en el Project Navigator — en Modo principiante se queda con el borrado por swipe.

 **EN POCAS PALABRAS**

La lista de la derecha muestra cada imagen importada con resolución y tamaño de archivo — práctico para ver de un vistazo si has mezclado material de alta y baja resolución. Para eliminar una sola imagen, deslízala con dos dedos hacia la izquierda en el trackpad — como en Mail de iOS — o selecciónala y pulsa Suprimir. La app no elimina el archivo en sí; solo lo saca del proyecto actual. Si necesitas un botón menos en condiciones o deshacer con Cmd-Z, pasa al Modo Experto (Cmd+2), allí existe en el Project Navigator. En Modo principiante se queda deliberadamente con el patrón simple de swipe.

C-15 Avisos de validación (3 niveles) **DÓNDE**

Bajo la lista de imágenes, sobre el botón Clear All.

 **TÉCNICO**

Tres umbrales sucesivos basados en el número de imágenes importadas (solo activos cuando hay imágenes y no vídeo): - < 3 imágenes: banner rojo (octógono rojo), texto «At least 3 images are required. Camera alignment cannot be computed from fewer images.» - 3–9 imágenes: banner rojo, texto «With fewer than 10 images, SfM often fails and the trained scene tends to overfit [...]. 15–20 images minimum recommended; 30+ for object captures.» - 10–19 imágenes: banner naranja (triángulo de aviso), texto «Workable, but quality usually improves with 20+ images and good coverage around the scene.»

A partir de 20 imágenes el banner desaparece. Los umbrales están codificados en duro y se basan en más de 560 experimentos de entrenamiento empíricos.

 **EN POCAS PALABRAS**

La app mira cuántas imágenes has importado y te da una valoración codificada por colores. Rojo significa: es muy probable que esto falle — o SfM no puede calcular cámaras o el entrenamiento sobreajusta sobre muy poco material. Naranja significa: podría funcionar, pero no esperes calidad sobresaliente, porque el algoritmo encuentra poco solapamiento entre las imágenes. Sin banner significa: buenas condiciones, tienes suficiente material. Si quieres realmente modelos limpios, apunta a al menos 30–50 tomas uniformemente distribuidas alrededor de tu motivo — también con gusto bastantes más para escenas exteriores o salas grandes. Puedes empezar a pesar del aviso, pero no te sorprendas si SfM aborta sin comentarios o el modelo se ve agujereado.

C-16 Detección de workspace COLMAP **DÓNDE**

Al soltar una carpeta — no un botón visible, sino lógica de detección.

 **TÉCNICO**

Al soltar un directorio se comprueba si contiene uno de los tres layouts canónicos de workspace: `sparse/0/cameras.bin`, `sparse/cameras.bin` o directamente `cameras.bin` en la raíz. Si es así, la enumeración estándar de imágenes se aborta y en su lugar se abre un alert modal preguntando al usuario si se debe usar la reconstrucción existente o si las imágenes deben pasarse de nuevo por Apple Photogrammetry. Misma vía también para workspaces en formato de texto (`cameras.txt`) y exportaciones ETH3D. Véase el backend Q6 del Capítulo 9 para detalles. Funciona en Modo principiante igual que en Modo Experto.

 **EN POCAS PALABRAS**

Si ya has trabajado con Metashape, RealityCapture o COLMAP y has hecho ahí el cálculo de cámaras, puedes simplemente arrastrar la carpeta de exportación aquí. RadianceKit detecta automáticamente por el contenido que es un workspace COLMAP (comprueba `sparse/0/`, `cameras.bin` y similares) y te pregunta si debe asumir el cálculo terminado o calcularlo por sí mismo. Asumirlo ahorra horas de espera en escenas grandes, porque SfM se omite por completo — el entrenamiento arranca de inmediato. Los workspaces en formato de texto (`cameras.txt`) y las exportaciones ETH3D también se detectan. Esta función está disponible en Modo principiante igual que en Modo Experto; más detalles están en el Capítulo 9 bajo el backend Q6.

¿Cuándo pasar a la siguiente etapa?

Puedes hacer clic en Start Processing en cuanto (a) se haya importado al menos una imagen o un vídeo y (b) el banner de validación esté en naranja o haya desaparecido. Con un banner rojo, la app sí te deja arrancar, pero con alta probabilidad podrás cancelar el procesado de inmediato. Recomendado: al menos 20 imágenes, nítidas, con solapamiento claro entre tomas consecutivas, todas aproximadamente desde la misma distancia al motivo. Elige un ajuste preestablecido antes de empezar que encaje con tu presupuesto de tiempo — con 30 imágenes y el ajuste Quick estás listo en unos minutos, con Quality dura más bien 1–2 horas.

Z2 — Procesado (SfM + Entrenamiento)

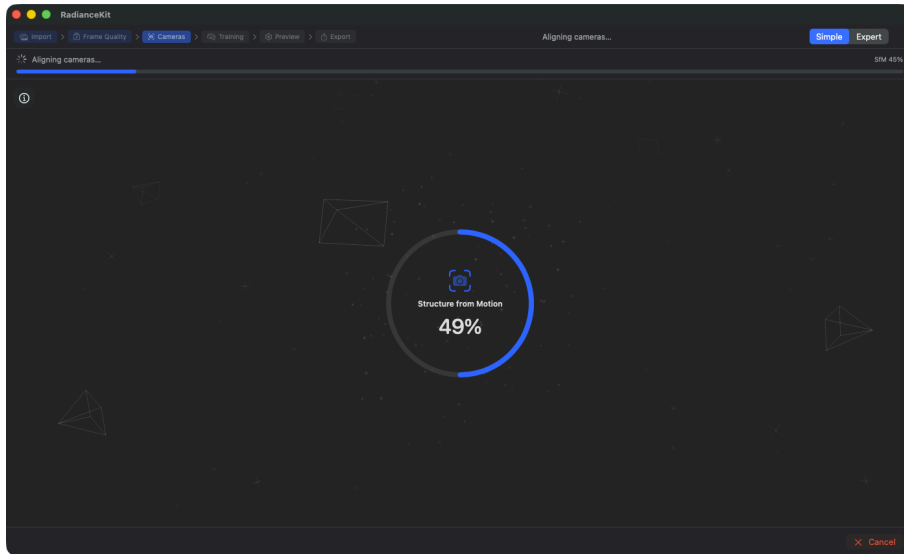


Figura 36: Z2 fase SfM — icono de etapa «Structure from Motion» con 41 % en el círculo grande, barra de estado superior en «SfM 25 %», botón Cancel abajo a la derecha

Fase SfM (se están alineando las cámaras): El gran círculo de progreso muestra el progreso de la subetapa (aquí 41 % de la sesión Apple Photogrammetry en curso). Texto de estado «Aligning cameras...» arriba a la izquierda. La ruta de migas marca «Cameras» como etapa activa. La barra de estado superior muestra el progreso global de la pipeline (25 %) — SfM ocupa la primera mitad de la barra. Cámaras de wireframe flotantes en el fondo sugieren que se están estimando poses.

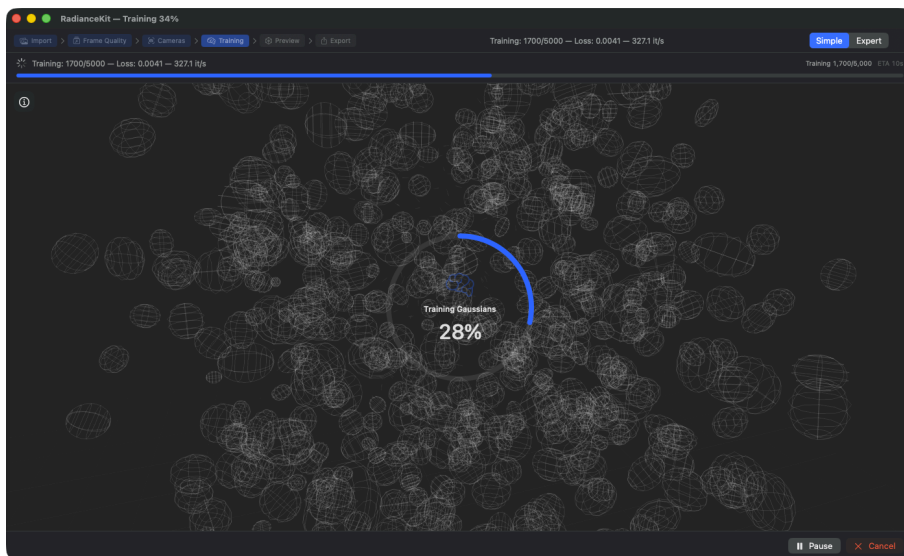


Figura 37: Z2 fase de entrenamiento — icono de etapa «Training Gaussians» al 6 %, métricas en vivo arriba (Training: 400/5000 — Loss: 0,1642 — 138,7 it/s), ETA 33 s, Pause/Cancel abajo

Fase de entrenamiento (se están optimizando los Gaussianos): El icono de subetapa cambia a «Training Gaussians», el porcentaje cuenta iteraciones del ajuste preestablecido elegido (aquí 400 / 5 000 para el ajuste Preview, es decir, 8 % de la etapa).

La línea de métricas en vivo muestra el valor de loss (0,1642), iteraciones por segundo (138,7 it/s) y ETA (33 s). El progreso global de la pipeline sube del 50 % al 100 % durante esta fase. El botón Pause (en lugar de solo cancel en la fase SfM) permite reanudar después; Cancel descarta el resultado del entrenamiento y vuelve a Z1.

En cuanto corre la pipeline, la app oculta el overlay de importación y muestra una pantalla de procesado a pantalla completa. En el medio corre un gran círculo de progreso (220 × 220 píxeles) con icono de etapa, texto de estado y número de porcentaje; en el fondo, una animación sutil de splats visualiza simbólicamente el cálculo en curso. Arriba a la izquierda se puede mostrar un panel de info que muestra métricas en vivo de entrenamiento y SfM. Abajo hay Pause/Resume, Cancel y, en caso de error, un botón Retry.

C-18 SplatTrainingView (animación de fondo)

DÓNDE

Fondo a pantalla completa tras el círculo de progreso, oculto al cancelar o en error.

TÉCNICO

Animación decorativa que, según el progreso de la pipeline (0...1), renderiza un número creciente de pequeñas partículas splat animadas. La fuente es un valor de progreso calculado que mapea las fases SfM a 0–0,2 y el entrenamiento a 0,2–1,0 (Frame Quality a 0–0,05). De este modo los splats visiblemente «se construyen» mientras corre el entrenamiento. Puramente decorativo — la visualización no muestra resultados intermedios reales del entrenamiento actual (eso sería la live preview en Modo Experto). En cancel o fallo se oculta y solo se mantiene visible el círculo de estado.

EN POCAS PALABRAS

En el fondo corre una pequeña animación de puntos danzantes, para que la pantalla no parezca tan vacía durante el cálculo. Este no es tu modelo 3D real — eso solo lo ves tras el entrenamiento en el paso Z3. La animación, sin embargo, tiene la misma tonalidad, así que puedes leer por la densidad aproximada lo lejos que ha llegado el entrenamiento. Al principio solo unos pocos puntos son visibles, hacia el final el fondo se rellena bastante más densamente — un bonito indicador visual además de la visualización en porcentaje del círculo. Si la animación te molesta (p. ej., porque quieres trabajar al lado en segundo plano), puedes pasar al Modo Experto, donde se omite.

C-19 Gran círculo de progreso **DÓNDE**

Centro de la pantalla de procesado, 220 × 220 píxeles.

 **TÉCNICO**

Dos anillos renderizados uno sobre otro: por fuera un anillo de pista atenuado, por dentro un anillo de progreso relleno con stroke de acento o rojo (rojo en error). Dentro del círculo, un icono de etapa (cerebro para entrenamiento, cámara para SfM, película para extracción de fotogramas de vídeo, destellos para Frame Quality), título de etapa y el número de porcentaje animado en vivo en fuente redondeada de 32 puntos. El icono pulsa suavemente mientras el procesado está activo. La visualización interpola en un timer de 30 Hz de forma suave hacia el progreso real actual — con un «creep» constante (0,0003/frame) más un share proporcional (4 % del gap) y un techo suave que se fija al 80 % del siguiente hito esperado (para SfM a partir de una tabla de hitos codificada). De este modo el progreso se siente fluido, incluso si las actualizaciones SfM reales solo llegan cada pocos segundos.

 **EN POCAS PALABRAS**

El gran círculo en el medio es tu visualización principal mientras la app calcula. Se rellena con suavidad, incluso cuando las actualizaciones de cálculo reales solo llegan cada pocos segundos — esto te da la sensación de que algo está pasando, en lugar de mirar un porcentaje congelado durante minutos. El símbolo del medio cambia según si se están extrayendo fotogramas (icono de película), se están alineando cámaras (icono de cámara) o se están entrenando Gausianos (icono de cerebro). El número de porcentaje se refiere al sub-paso actual — ves la pipeline global en la barra estrecha de arriba del todo. En error, el anillo se vuelve rojo en lugar de azul, y el icono ya no pulsa, así notas de inmediato que algo ha ido mal.

C-22 Botón Info (mostrar métricas) DÓNDE

Arriba a la izquierda en la pantalla de procesado, 32 × 32 píxeles.

 TÉCNICO

Botón simple con fondo material. Conmuta el panel de info on/off. El icono cambia entre info-circle outline e info-circle filled cuando está activo. Animación de fade-in suave. Tooltip «Show detailed processing metrics».

 EN POCAS PALABRAS

Por defecto, la pantalla está deliberadamente ordenada — solo el gran círculo de progreso, no ves más al principio. Si como usuario técnicamente interesado quieres saber con más precisión qué está pasando (qué iteración, lo alto que es el loss, cuántos Gaussianos), haz clic en el símbolo i arriba a la izquierda. Un pequeño panel se despliega abajo y muestra todos los valores en vivo. Un nuevo clic lo oculta de nuevo. El ajuste no es persistente — en cada nuevo run de entrenamiento el panel está inicialmente oculto de nuevo, lo que se elige deliberadamente para no asustar a los principiantes.

C-23 Panel de info (métricas en vivo) DÓNDE

Abajo a la izquierda en la pantalla de procesado, visible solo cuando `showProcessingInfo == true`.

 TÉCNICO

Panel de dos columnas con fondo material ultradelgado. Columna izquierda: líneas de info específicas de etapa — para SfM, texto de estado y porcentaje; para entrenamiento, iteración, loss combinado, loss L1, loss D-SSIM, número de Gaussianos (coloreado en naranja), velocidad (it/s), tiempo transcurrido, ETA calculado, grado SH y learning rate. Columna derecha: texto de estado, string de info de tiempo, gráfico de loss inline (véase C-28) y un discoverability nudge (véase C-32). Todos los valores se leen del estado de entrenamiento, que se actualiza en cada tick de entrenamiento.

 EN POCAS PALABRAS

El panel de info muestra todos los valores en vivo que en Modo Experto estarían permanentemente en la barra lateral del Inspector: iteración actual, valor de loss (menor: mejor), número de Gaussianos, velocidad, tiempo restante estimado, grado SH y learning rate. En el lado derecho corre además una curva de loss diminuta, que te dice de un vistazo si el entrenamiento va en la dirección correcta. Si el entrenamiento parece perezoso, una mirada aquí ayuda — un loss que ya no cae, o un ETA que ya no disminuye, indican problemas. Si el loss explota (de repente se hace enorme) o muestra NaN, el entrenamiento se ha vuelto inestable y un Cancel + Retry o cambiar a un ajuste preestablecido distinto es sensato.

C-25 Botón Pause/Resume **DÓNDE**

Barra de navegación inferior, visible solo durante la etapa de entrenamiento (NO durante SfM) y mientras corre el procesado.

 **TÉCNICO**

Botón con borde. Llama a Pause o Resume según el estado. La etiqueta cambia entre «Pause» (con icono de pausa) y «Resume» (icono de play). Durante el paso SfM el botón no se muestra, porque Apple Photogrammetry no tiene semántica de pausa. El estado de pausa preserva por completo la iteración, el estado de Gaussianos y el momentum del optimizador — Resume continúa donde se detuvo previamente.

 **EN POCAS PALABRAS**

Mientras corre el entrenamiento, puedes detenerlo en cualquier momento y reanudar después. Útil cuando quieres hacer otra cosa en el Mac mientras tanto que necesita mucha GPU — p. ej., edición de vídeo, prueba de juegos o una exportación de render desde otra app. Haz clic en Pause, haz tu cosa, haz clic en Resume, el entrenamiento continúa exactamente donde estaba. Contador de iteración, número de Gaussianos y momentum del optimizador se preservan por completo, el estado de pausa no te cuesta calidad. Durante la fase SfM, Pause no está disponible — Apple Photogrammetry no tiene función de parada, en una emergencia ahí tienes que trabajar con Cancel.

C-26 Botón Cancel **DÓNDE**

Barra de navegación inferior, visible mientras corre el procesado (SfM o entrenamiento).

 **TÉCNICO**

Botón rojo con borde. Abre un diálogo de confirmación con título «Stop and discard progress?», botones «Discard Progress» (destrutivo) y «Keep Running» (cancel). En la confirmación se fija la cancel flag, se termina la tarea de entrenamiento, se termina el subproceso SfM si procede, y se escribe una línea de resumen con estado de cancelación en el log JSONL. A diferencia de Pause, los buffers y el estado de entrenamiento se descartan.

 **EN POCAS PALABRAS**

El botón de cancelar. A diferencia de Pause, esto es final — si quieres reiniciar después, el procesado corre desde el principio, todas las iteraciones ya entrenadas se pierden. Útil cuando te equivocaste con el ajuste preestablecido, el entrenamiento corre muchísimo más despacio, o la app está produciendo obviamente resultados basura y no quieres esperar. Antes del cancel real, la app pregunta de nuevo mediante un diálogo de confirmación, para que no pierdas accidentalmente horas de tiempo de cálculo. Si solo quieres interrumpir brevemente, mejor toma Pause.

C-27 Botón Retry**DÓNDE**

Barra de navegación inferior, visible cuando la pipeline ha fallado (el estado SfM empieza con «SfM failed» o el entrenamiento está en estado de error).

**TÉCNICO**

Botón de acento. Reinicia toda la pipeline. Antes de arrancar se comprueba si aún están presentes las imágenes/vídeos importados. Los logs de error anteriores se mantienen en el directorio JSONL; un nuevo run escribe un nuevo archivo de log con la marca de tiempo actual.

**EN POCAS PALABRAS**

Si SfM o entrenamiento aborta con un mensaje de error, puedes intentarlo de nuevo aquí. A veces eso ayuda, porque muchos pasos (RANSAC, densificación) tienen componentes aleatorios y un segundo intento puede tener éxito donde el primero falló. Toda la pipeline corre entonces de nuevo desde el principio — SfM y entrenamiento, en un nuevo archivo de log JSONL. Si también el segundo intento falla, normalmente las imágenes de entrada son el problema (muy pocas, muy poco solapamiento, motion blur, mala luz); entonces vuelve atrás con Back y cambia tu material. Truco: mira en paralelo en los training logs (Help → Open Training Logs), allí dice con más detalle dónde exactamente se atascó.

C-28 Gráfico de loss inline**DÓNDE**

En el panel de info, columna derecha, visible solo durante el entrenamiento con historial no vacío.

TÉCNICO

Área de dibujo compacta (40 píxeles de alto), dibuja el historial de loss como línea de 1 píxel en color de acento. Los datos se filtran a valores finitos (protección NaN para entrenamientos inestables). Min/Max se calculan sobre todo el historial — el gráfico hace así auto-zoom al rango de valores. El último valor de loss queda arriba a la derecha sobre el gráfico. El propio historial se construye en el estado de la app en cada tick de entrenamiento (típicamente cada 100 iteraciones).

EN POCAS PALABRAS

Una curva de loss diminuta que te muestra de un vistazo si el entrenamiento está «convergiendo» (la línea cae a la derecha) o si está estancado o explotando (línea plana o en subida). Con un entrenamiento sano, la línea cae con fuerza al principio y luego se aplana — ese es el curso esperado, similar a una curva de bajada a la mitad. El gráfico hace auto-zoom al rango de valores actual, así que también las pequeñas mejoras al final del entrenamiento permanecen visibles. Si la línea de repente dispara hacia arriba o se congela, es una buena señal de que algo va mal — o el material es problemático o un ajuste preestablecido distinto sería más adecuado. El gráfico lo encuentras en el panel de info, que muestras arriba a la izquierda con el símbolo i.

C-32 Discoverability nudge (pista del Modo Experto) **DÓNDE**

En el panel de info, columna derecha abajo, visible solo durante el entrenamiento Y en Modo principiante.

 **TÉCNICO**

Línea pequeña con icono de ojo y texto de caption «Switch to Expert Mode (⌘2) for live splat preview», en tono discreto y fuente de 10 puntos. Sin elemento interactivo, solo una pista. No reacciona al clic — el usuario tiene que pulsar realmente Cmd+2 o hacer clic en el menú Mode → Expert Mode.

 **EN POCAS PALABRAS**

Una pista sutil de que en Modo Experto durante el entrenamiento la versión intermedia actual de tu modelo 3D es visible en vivo en la vista. En Modo principiante esto se oculta deliberadamente para mantener la UI tranquila — pero muchos usuarios ni siquiera saben que esta función existe, así que aquí lo señalamos con suavidad. Pulsa Cmd+2 y el entrenamiento continúa en segundo plano mientras puedes mirar cómo tu modelo se va formando ante tus ojos. Esta también es una buena herramienta para estimar ya tras unas pocas miles de iteraciones si el resultado va a ser bueno, o si más bien prefieres cancelar y empezar de cero. Cmd+1 te trae de vuelta a la vista Simple en cualquier momento.

¿Cuándo pasar a la siguiente etapa?

La app cambia automáticamente a Z3 (Preview) en cuanto el entrenamiento se completa con éxito — no tienes que hacer clic. La barra de navegación inferior cambia entonces de Pause/Cancel a un botón Back (atrás a Import) y un botón Export (adelante a Export). En casos de error (mensaje de error rojo, el icono de etapa es una X) aparece Retry en su lugar, y tienes que decidir si arrancar de nuevo o ir atrás a Import con Back para cambiar material de imagen.

Z3 — Previsualización (rotar modelo 3D)

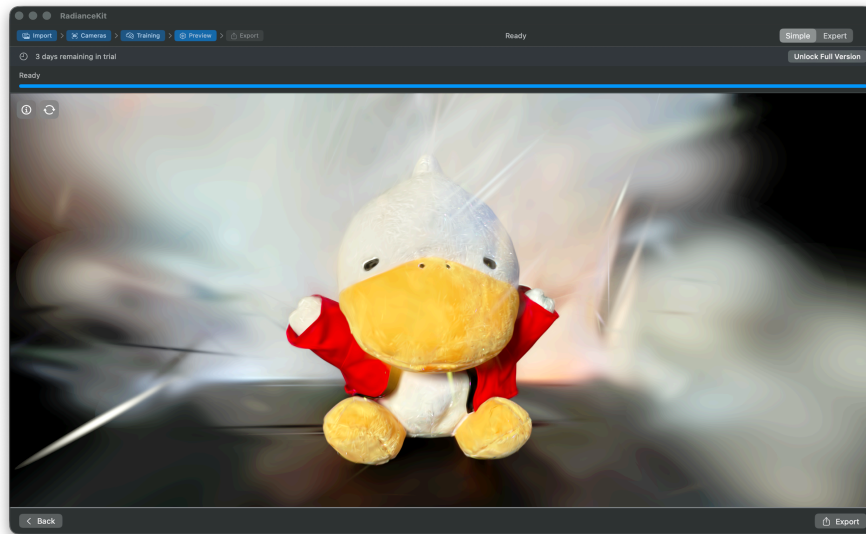


Figura 38: Paso de previsualización del Modo principiante con visor 3D

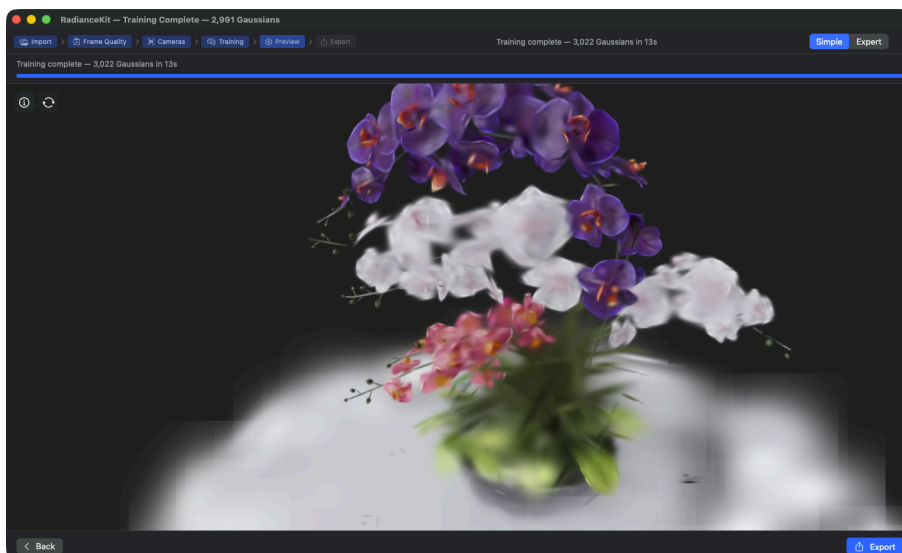


Figura 39: Z3 previsualización tras completarse el entrenamiento — ramo Blender reconstruido, la cabecera muestra «Training complete — 3.022 Gaussians in 13s», botones Back y Export abajo

LO QUE SE VE EN LA IMAGEN La ruta de migas marca «Preview» como etapa activa. La vista 3D a pantalla completa renderiza la escena de ramo ya entrenada (conjunto de prueba sintético en Blender, subconjunto de 60 frames de 960 cámaras hemisféricas). Barra de estado de cabecera: «Training complete — 3 022 Gaussians in 13 s» — da el número final de Gaussianos y el tiempo de entrenamiento. Arrastrar en la vista rota la cámara (yaw/pitch); la rueda de scroll hace zoom a lo largo de la dirección de visión. El botón «Back» (abajo a la izquierda) vuelve a Z2 para reanudar o re-run; el botón «Export» (abajo a la derecha, primario) navega adelante a Z4.

Tras completarse el entrenamiento, la app aterriza automáticamente en la previsualización. Aquí ves tu modelo Gaussian Splating terminado en una vista Metal a pantalla

completa y puedes rotarlo, hacer zoom y desplazarlo con ratón y trackpad. Sobre la vista hay un pequeño overlay con controles de cámara e info — auto-rotación, estadísticas de entrenamiento, botón de reset. Antes del siguiente paso (Export) es recomendable inspeccionar el modelo desde distintos ángulos para asegurarte de que la reconstrucción está limpia.

C-36 SplatViewportView (vista principal 3D)

DÓNDE

Fondo a pantalla completa del paso de previsualización.

TÉCNICO

Vista 3D basada en Metal que renderiza la nube de puntos terminada. El renderer es el propio rasterizador ForwardPass de RadianceKit — el mismo que ya renderiza los splats durante el entrenamiento — así que es verdadero WYSIWYG (lo que se entrena se muestra y se exporta exactamente igual). Pipeline de renderizado basado en tiles con transparencia independiente del orden. Si el renderer no puede inicializarse (p. ej., porque Metal no está disponible en el sistema), aparece en su lugar un fondo negro con texto «Metal not available». La vista ignora el safe area, para que el modelo llegue al borde de la ventana.

EN POCAS PALABRAS

La vista principal. Aquí ves tu modelo 3D terminado reconstruido de tus fotos, renderizado en la GPU en tiempo real. Haz clic y arrastra con el botón izquierdo del ratón para rotar. Rueda de scroll o gesto de trackpad con dos dedos para zoom. Botón derecho o Cmd+arrastrar para hacer pan. El modelo consiste en decenas de miles de elipsoides 3D semitransparentes («Gaussianos») que reconstruyen tu escena de forma fotorrealista — cada uno tiene una posición, orientación, forma y color que el entrenamiento ha aprendido. En el raro caso de que tu Mac no soporte Metal, ves en su lugar un fondo negro con un mensaje de aviso — RadianceKit necesita absolutamente una GPU capaz de Metal.

C-37 CameraControlsOverlay (overlay de controles) **DÓNDE**

Sobre la vista, flotante.

 **TÉCNICO**

Overlay UI compacto con botones para auto-rotación (turntable), reset de cámara, elección de fondo (gris/negro/blanco), guardar captura, toggle del panel de info. Vincula a los parámetros de cámara (distancia, azimuth, elevación, target, FOV) y controla el auto-turntable. Durante el entrenamiento (cuando el usuario en Modo Experto quiere ver la vista co-riendo en paralelo), el overlay muestra además una línea compacta de estado de entrenamiento.

 **EN POCAS PALABRAS**

La pequeña barra flotante sobre el modelo. Aquí inicias la auto-rotación (el modelo rota por sí mismo, bueno para capturas y demos cortas), reseteas la cámara con Reset a la posición inicial (por si te pierdes), cambias el fondo (gris para neutro, negro para máximo contraste, blanco para modelos brillantes) y haces capturas directamente que se guardan bajo /Pictures. Práctico cuando quieres mostrar un determinado detalle desde un ángulo muy específico sin tener que exportar adicionalmente todo el modelo. La auto-rotación también es una buena prueba de si el modelo se ve igual de bien por todos los lados o si hay un «lado deslucido» que surgió por tomas que faltan.

C-38 Botón Export (barra de navegación) **DÓNDE**

Barra de navegación inferior en Z3.

 **TÉCNICO**

Botón de acento con etiqueta «Export» e icono de share. El clic dispara el cambio a Z4. Antes, la vista padre comprueba si la versión completa está desbloqueada — si no, en lugar de la etapa de exportación se muestra la vista de lock (véase U-06).

 **EN POCAS PALABRAS**

Cuando estás contento con el resultado, haz clic en Export y aterrizas en el último paso, donde eliges el formato y guardas. Sin una versión completa comprada, aterrizas en su lugar en un bloqueo de pantalla con una pista de desbloqueo y un botón de compra — la app no quiere empujarte la versión completa, pero Export es una de las funciones premium. Una vez completada la compra, la app sigue directamente en estado desbloqueado y aterrizas en la etapa de exportación familiar. Si cambias de idea, vuelves a la previsualización con el botón Back y puedes seguir rotando el modelo.

¿Cuándo pasar a la siguiente etapa?

Antes de exportar, rota el modelo una vez por completo y comprueba: ¿Están presentes todas las áreas que cubriste en tus imágenes de entrada? ¿Hay «floaters» flotantes (nubes de splat Gaussianos flotando libremente en el aire)? ¿Se ve el fondo/cielo limpio o emborronado? Los problemas graves solo se pueden corregir reentrenando — ya sea con más imágenes, un ajuste preestablecido distinto, o en Modo Experto con ajustes de reducción de floaters.

Z4 — Exportar (elegir formato y guardar)

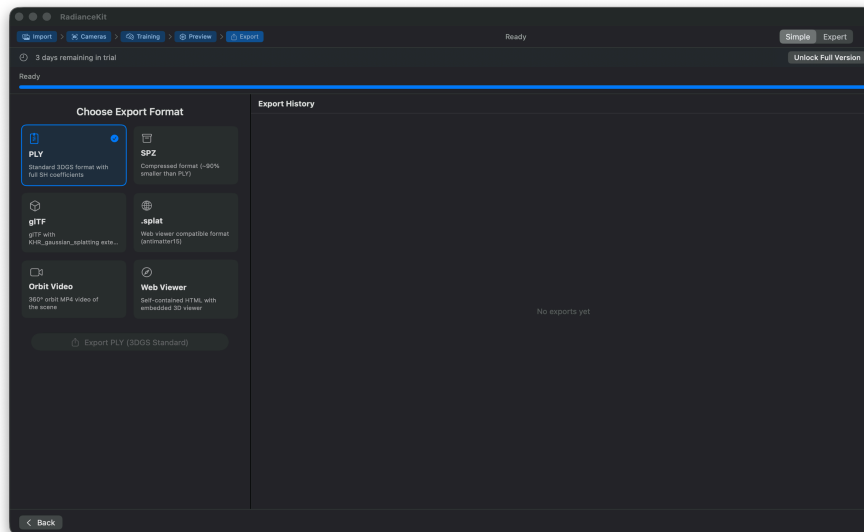


Figura 40: Paso de exportación del Modo principiante con tarjetas de formato

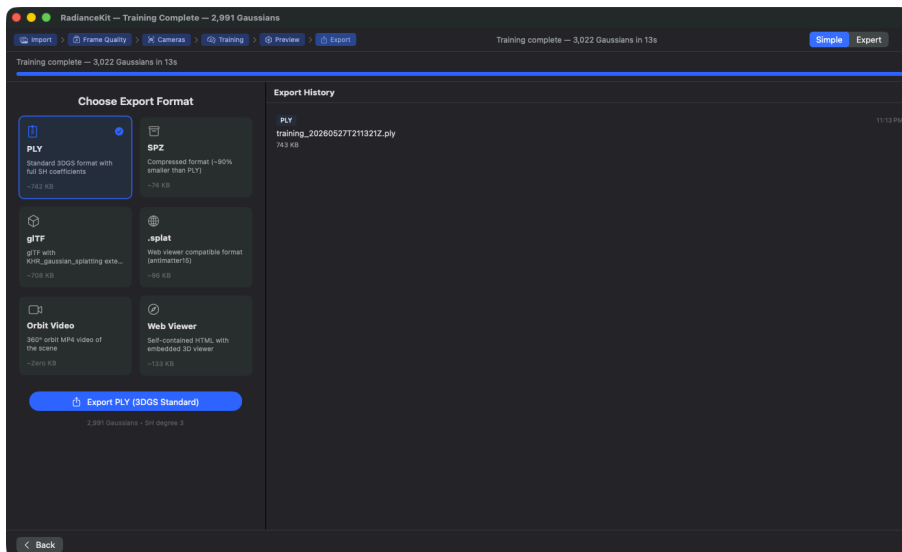


Figura 41: Z4 tarjetas de exportación — 6 formatos (PLY 742 KB seleccionado, SPZ 74 KB, glTF 708 KB, .splat 96 KB, Orbit Video, Web Viewer 133 KB), barra lateral de exportación a la derecha con PLY ya exportado

LO QUE SE VE EN LA IMAGEN La ruta de migas marca «Export» como etapa activa. Cuadrícula de tarjetas a la izquierda «Choose Export Format» con las seis opciones: PLY (estándar 3DGS, 742 KB, con coeficientes SH completos — preseleccionado aquí con check azul), SPZ (formato 3DGS comprimido, ~90 % más pequeño que PLY, 74 KB), glTF (con extensión `KHR_gaussian_splatting`, 708 KB), `.splat` (compatible con visor web vía `antimatter15`, 96 KB), Orbit Video (MP4 360° de la escena, cálculo de tamaño en vivo), Web Viewer (HTML autónomo con visor 3D embebido, 133 KB). Las cifras de tamaño se calculan en vivo a partir del número actual de Gaussianos y el overhead del formato. A la derecha, «Export History» lista las exportaciones ya completadas con insignia de formato, nombre de archivo y marca de tiempo — el clic revela en Finder. CTA primaria abajo a la izquierda: «Export PLY (3DGS Standard)» con subtítulo de Gaussianos «2.991 Gaussians · SH degree 3».

En el último paso eliges entre 6 formatos de exportación (PLY, SPZ, glTF, `.splat`, vídeo orbital, visor web) mediante una cuadrícula de tarjetas de 2 columnas, haces clic en Export y eliges el destino en el diálogo de macOS. A la derecha corre un historial de todas las exportaciones previas — al seleccionar tarjeta se muestra de inmediato bajo cada una el tamaño de archivo estimado, así p. ej. prefieres SPZ si quieres ir a la web (pequeño), y PLY si quieres importar a otro software (SuperSplat, Postshot, Blender vía plugin) (grande y completo).

C-39 Cuadrícula de formatos de 2 columnas

DÓNDE

Lado principal izquierdo del paso de exportación.

TÉCNICO

Cuadrícula de tarjetas con dos columnas flexibles y 12 puntos de espaciado. Itera sobre los formatos ofrecidos en Modo principiante — un subconjunto filtrado de la lista completa de formatos que contiene solo los 6 más importantes: PLY, SPZ, glTF, `.splat`, vídeo orbital, visor web. Compressed PLY y SOG se ofrecen SOLO en Modo Experto.

EN POCAS PALABRAS

Una cuadrícula de tarjetas con los 6 formatos que son relevantes en Modo principiante: PLY (formato estándar para otras herramientas 3D), SPZ (variante comprimida para web), glTF (estándar oficial Web3D), `.splat` (para el visor web `antimatter15`), vídeo orbital (MP4 terminado para presumir) y Web Viewer (archivo HTML autónomo con reproductor 3D embebido). Así cubres el 90 % de los casos de uso. Si necesitas uno de los formatos menos habituales (PLY comprimido o SOG para compresión extrema), pasa al Modo Experto, allí están disponibles los 8 formatos. La selección compacta aquí es deliberada, para que los principiantes no se vean abrumados por la variedad.

C-40 Botón de tarjeta de formato **DÓNDE**

Cada tarjeta en la cuadrícula.

 **TÉCNICO**

Botón simple con layout de tarjeta: icono (p. ej., document zipper para PLY, archive box para SPZ, icono de vídeo para vídeo orbital) arriba, nombre de formato como headline, caption de descripción (truncado a 2 líneas), debajo el tamaño de archivo estimado (calculado en vivo a partir de formato, número de Gaussianos y grado SH y formateado como KB/MB). Al hacer clic se selecciona el formato. La tarjeta seleccionada recibe un fondo de acento, borde de acento y un icono de check arriba a la derecha. El tooltip es la descripción del formato.

 **EN POCAS PALABRAS**

Una tarjeta por formato. Haz clic en una, se resalta con un color de acento y un check, y el botón de export debajo ajusta su texto («Export PLY», «Export SPZ», etc.). Cada tarjeta muestra un símbolo encajado, el nombre, una explicación breve de dos líneas y el tamaño de archivo estimado para tu resultado de entrenamiento actual. El tamaño te ayuda a elegir con sentido — si quieres enviar el resultado por correo, toma la variante más pequeña (normalmente SPZ o .splat); si quieres seguir trabajando en otro software 3D, toma el de mejor compatibilidad (típicamente PLY). Al pasar por encima de una tarjeta, el tooltip muestra una descripción más detallada, por si las diferencias entre los formatos no te quedan claras.

C-41 Deslizador de duración de vídeo **DÓNDE**

Bajo la cuadrícula de formatos, visible solo cuando se elige un formato de vídeo (vídeo orbital o vídeo social).

 **TÉCNICO**

Deslizador 3–30 segundos en pasos de 1 segundo, vincula a la duración del vídeo en el estado de la app. Ancho máximo 300 píxeles. Solo se muestra cuando se selecciona un formato de vídeo. Para formatos no de vídeo, el deslizador se elimina por completo de la vista — sin espacio muerto.

 **EN POCAS PALABRAS**

Si eliges un vídeo orbital como exportación, aquí puedes determinar la duración. 3 segundos = rotación muy rápida, 30 segundos = rotación lenta y tranquila alrededor de tu modelo. Para reels de redes sociales (Instagram, TikTok), normalmente 6–10 segundos es ideal — lo bastante largo para mostrar el modelo, lo bastante corto para que los espectadores no se vayan. Para presentaciones o vídeos de portfolio puedes con gusto tomar 15–20 segundos. El deslizador solo aparece cuando se selecciona un formato de vídeo; para formatos de archivo como PLY o SPZ no tendría sentido y se oculta.

C-42 Botón Export

DÓNDE

Bajo la cuadrícula de formatos (y bajo el deslizador de duración, si se eligió vídeo).

TÉCNICO

Gran botón de acento. Etiqueta: «Export {format-name}», icono de share. Al hacer clic se abre el diálogo de guardar de macOS con una extensión adecuada al formato y nombre predeterminado «scene.{ext}»; en la confirmación, la exportación se escribe en la URL elegida. Deshabilitado cuando no hay resultado de entrenamiento o ya está corriendo una exportación.

EN POCAS PALABRAS

Clic, elige el destino en el diálogo de macOS, listo — la app escribe el archivo en el formato elegido en el destino seleccionado. Nombre predeterminado es «scene.{extensión}» (p. ej., «scene.ply» o «scene.spz»), lo puedes cambiar en el diálogo como quieras antes de guardar. El botón está atenuado mientras no haya aún resultado de entrenamiento (no debería ocurrir aquí, porque en otro caso no estarías en el paso de exportación) u otra exportación ya está corriendo. En cuanto corre la exportación, aparece bajo él una visualización de progreso; la app sigue operable, así que puedes preparar ya la siguiente exportación.

C-43 Barra de progreso de exportación

DÓNDE

Bajo el botón Export, visible solo mientras corre una exportación.

TÉCNICO

Visualización de progreso con ancho máximo 300 píxeles, debajo el caption «Exporting... N %». El valor corre de 0 a 1 y se actualiza durante la escritura — con PLY en chunks de 10 000 Gaussianos, con SPZ una vez tras la cuantización, con vídeo orbital en intervalos de frame.

EN POCAS PALABRAS

Mientras corre la exportación, ves aquí el progreso como barra estrecha más visualización en porcentaje. PLY suele estar listo en segundos, porque el archivo simplemente se escribe binariamente. SPZ tarda un poco más, porque los datos se cuantizan y comprimen en el proceso. El vídeo orbital es la exportación más costosa en tiempo — aquí cada fotograma se rerenderiza; según resolución y duración, puede durar un minuto o más. Durante la exportación, la app sigue operable, así que puedes preparar ya el siguiente formato o seguir haciendo clic en la vista.

C-44 Visualización de error de exportación

DÓNDE

Bajo la barra de progreso, visible solo cuando ocurrió un error durante la última exportación.

TÉCNICO

Línea roja con icono de aviso y texto de error. Fondo rojo con opacidad del 8 %, esquinas redondeadas. Ancho máximo 400 píxeles. Causas comunes de error: SOG espera `cwebp` en el PATH del sistema (no compatible con App Store); error de escritura con espacio de disco lleno; error de sandbox con destinos de guardado fuera del área permitida.

EN POCAS PALABRAS

Si la exportación va mal, aquí aparece en rojo una breve descripción en texto claro del problema. Normalmente la causa es obvia — sin espacio en disco, sin permisos de escritura para la carpeta de destino, o un destino fuera de las áreas permitidas por el sandbox. Específicamente con el formato SOG ocurre que falta `cwebp` en el sistema; en este caso SOG no es utilizable y tienes que recurrir a SPZ. Si el mensaje de error no está claro, mira en el directorio de logs (Help → Open Training Logs), allí dice de forma más exhaustiva qué fue mal. En caso de duda, ayuda simplemente elegir un destino distinto — p. ej., el escritorio.

C-46 Lista de historial de exportación

DÓNDE

Lado derecho del paso de exportación.

TÉCNICO

Lista sobre el historial de exportación (persistente como JSON en los UserDefaults, mantenida tras cada exportación exitosa). Cada fila muestra insignia de formato (pequeña, color de acento), marca de tiempo (HH:mm), nombre de archivo (truncado a 1 línea) y tamaño formateado. El clic en una fila abre Finder con el archivo seleccionado. Estado vacío: «No exports yet».

EN POCAS PALABRAS

Una lista de tus exportaciones previas — formato, hora, nombre de archivo, tamaño, en orden cronológico. Haz clic en una fila y el archivo se resalta en Finder sin que tengas que navegar por carpetas tú mismo. Práctico cuando necesitas la última exportación de nuevo una hora después y ya no sabes dónde la guardaste — el historial lo recuerda. Si nunca has exportado nada, aquí hay una pista amistosa «No exports yet». La lista sobrevive a los reinicios de la app, porque se almacena en los UserDefaults.

C-48 Menú contextual del historial (clic derecho) **DÓNDE**

Clic derecho en una fila del historial.

 **TÉCNICO**

Menú contextual en cada entrada de la lista con dos acciones: «Reveal in Finder» (abre Finder con archivo seleccionado, como el clic simple) y «Copy Path» (coloca la ruta de archivo completa como texto en el portapapeles). Lo último es útil para arrastrar y soltar en otras apps o para pasar a la línea de comandos.

 **EN POCAS PALABRAS**

El clic derecho en una entrada del historial abre un pequeño menú con dos acciones. «Reveal in Finder» hace lo mismo que un clic normal — abre Finder con el archivo seleccionado, así que lo ves de inmediato. «Copy Path» coloca la ruta de archivo completa en el portapapeles, para que la puedas pegar p. ej. en comandos de terminal, en otras apps o en una nota. Particularmente práctico cuando quieres pasar la exportación a alguien o abrirla en otro programa que trabaja con entrada de ruta. Funcionalmente un detalle pequeño pero útil que apuesta por patrones de interacción típicos de Mac.

¿Cuándo está completo el flujo?

Tras una exportación exitosa, tienes tu modelo 3D como archivo en disco y el historial muestra una nueva entrada. No hay un botón «Done» — puedes anexas cualquier número de exportaciones en distintos formatos sin reentrenar. Si quieres volver a la previsualización (p. ej., para comprobar de nuevo una perspectiva de cámara), usa el botón Back en la barra de navegación inferior. Si quieres empezar una escena completamente nueva, ve vía Back a Z1 y usa allí Clear All, o File → New Project (Cmd+⇧+N).

Cambiar a Modo Experto

Pulsa Cmd+2 en cualquier momento o elige Mode → Expert Mode (M8). Todo el estado se conserva: imágenes importadas, ajuste preestablecido elegido, entrenamiento en curso o terminado, nube de puntos terminada, historial de exportación, incluso la etapa actual. En Modo Experto, en lugar de la etapa de cuatro pasos, se muestra la barra lateral completa del Inspector con todos los ~150 campos de control. En particular: el Project Navigator (véase Capítulo 2) ofrece las operaciones de imagen extendidas (botón menos, borrado por backspace, deshacer con Cmd-Z, previsualización Quick Look), la previsualización en vivo en la vista durante el entrenamiento, así como todos los parámetros de loss, MCMC, densificación y Mip-Splatting. Cmd+1 cambia de vuelta a Modo principiante — esto tampoco pierde estado.

Preguntas frecuentes

¿Por qué mi botón **Start Processing** se mantiene atenuado?

Aún no has importado imágenes o un vídeo. Arrastra al menos un archivo a la zona de drop o usa «Browse Files». En cuanto la lista de imágenes a la derecha contenga al menos una entrada, el botón se vuelve activo. (Con solo 1–2 imágenes sí arranca, pero SfM aborta directamente con un error — véase el banner de validación rojo.)

¿Por qué mi botón **Export** está bloqueado?

En Modo principiante hay dos niveles: (a) Si la pipeline de entrenamiento aún no ha terminado y no tienes ninguna, el botón está deshabilitado — tienes que terminar primero Z2. (b) Si aún no has comprado la versión completa (`PurchaseManager.hasAccess == false`), ves, en lugar de la etapa de exportación, una vista de lock con icono de candado y botón «Unlock Full Version», que abre el sheet de compra. Los ajustes Quick y Preview permiten entrenar gratis, pero la exportación es premium.

¿Por qué no puedo elegir un ajuste preestablecido?

Puedes elegirlo — pero si tocas un ajuste premium (Balanced, Quality, variantes MCMC) sin una versión completa comprada, el selector vuelve automáticamente a Preview y el sheet de compra se abre. Quick y Preview son los únicos ajustes utilizables libremente.

¿Por qué mi zona de drop está vacía y de un gris discontinuo, aunque estoy arrastrando imágenes?

Probablemente un desajuste de tipo UTI. La app acepta JPG, PNG, TIFF, HEIC, MP4, MOV más los formatos splat propios de la app. Otros formatos de imagen (BMP, GIF, WebP, formatos RAW) NO se reconocen. Si estás seguro de que tu tipo de imagen debería estar incluido, comprueba la extensión del nombre de archivo — la app va principalmente por extensión, no por contenido del archivo.

¿Por qué SfM tarda tanto, aunque solo tengo 30 imágenes?

Apple Photogrammetry no escala linealmente — con algunas constelaciones de imagen (espacios interiores con texturas complejas, motion blur, mala luz) tarda significativamente más de lo que sugeriría el número de imágenes. Si SfM sigue colgado tras más de 10 minutos con 30 imágenes, aborta e inténtalo de nuevo con mejor material, o pasa al Modo Experto y prueba COLMAP/Native SfM (Cmd+2 → Inspector → Camera Alignment).

¿Dónde encuentro mis training logs?

Help → Open Training Logs (Cmd+⇧+L). Eso abre `~/Documents/RadianceKit/Logs/`. Cada sesión de entrenamiento escribe su propio archivo JSONL con marca de tiempo en el nombre de archivo — la primera línea es la configuración, después sigue una línea de progreso cada 100 iteraciones, la última línea es el resumen con loss final y flag de éxito.



COLOFÓN

*Compuesto en SF Pro · Código en SF Mono ·
Typst 0.14 · 22. June 2026*

© 2026 Bjoern Kindler · Bischofshofener Str. 9, 82008 Unterhaching, Alemania

Hecho con ❤️ en Unterhaching