



RADIANCEKIT

# Manuale Utente

Ricostruzione 3D fotorealistica  
tramite Gaussian Splatting

---

Versione 1.5.0 · macOS 26.0+ · Maggio 2026

BJOERN KINDLER · KINDLER-DEV.DE

# Panoramica

---

Introduzione — Cosa devi sapere .....	3
Cos'è RadianceKit? .....	3
Cos'è il Gaussian Splatting? .....	3
Capitolo 1 — Barra dei menu .....	5
Menu File .....	6
Menu Mode .....	9
Menu Training .....	11
Menu Viewport .....	14
Menu Export .....	20
Menu Help .....	25
Nota: Cmd-Z nel menu Edit .....	29
Riepilogo delle scorciatoie da tastiera .....	30
Capitolo 2 — Inspector (Vista Esperto) .....	31
Sezione Look (L1–L5) .....	34
Sezione Preset (I1–I11) .....	37
Sezione Configurazione del Training (I12–I22) .....	43
Sezione Enhancements (I26–I29, I42–I44) .....	50
Sezione Metriche (I30–I38) .....	57
Sezione Grafico di perdita (I39–I41) .....	63
Quando ricorrere all'Inspector? .....	66
Capitolo 3 — Impostazioni .....	68
Tab General .....	69
Tab AI Helpers .....	74
Impostazioni Specchio dell'Inspector .....	77
Quando cosa? .....	78
Capitolo 4 — Finestre ausiliarie .....	79
User Guide (W1–W4) .....	80
Keyboard Shortcuts (W5–W6) .....	83
Manage Storage (W7–W12) .....	85
Pareto Dashboard (W13–W22) .....	89
Holdout Analysis (W23–W29) .....	96
BayesOpt Console (W30–W39) .....	101
Finestra principale: andamento della perdita e Gaussian Count (I39–I41, rimando) . . . .	108
Box delle regole pratiche .....	109
Capitolo 6 — Configurazione del training .....	111
Iterazione (T1–T2) .....	113
Learning Rate (T3–T10) .....	115

Densification — Classic (T11–T16) .....	123
Loss (T17–T20) .....	127
Progressione del grado SH (T21) .....	131
Prestazioni (T22–T25) .....	132
Diagnosi e preparazione della nuvola di punti (T26–T30) .....	135
Regolarizzazione (T31–T37) .....	138
Refinement (T38–T44) .....	141
Sky-Dome (T45–T48) .....	145
Adam + LR-Schedule (T49–T55) .....	147
Post-processing + Apple AI (T56–T60) .....	151
Densification MCMC (T61–T73) .....	153
Mip-Splatting (Q1.5) (T74–T76) .....	160
Densification adattiva (Q5) (T77–T79) .....	162
Curriculum (Q6) (T80–T81) .....	164
Preset statici (TP1–TP9) .....	164
Metodo: .....	167
Quale campo per cosa? (cheat sheet) .....	168
Campi pericolosi .....	169
Capitolo 7 — Preset di qualità integrati .....	170
Quando quale preset? .....	180
Confronto rapido .....	181
Preset propri .....	183
Capitolo 8 — Formati di esportazione .....	184
Quale formato quando? .....	198
Confronto rapido .....	199
Capitolo 9 — Backend SfM .....	200
Quale backend quando? .....	206
Confronto rapido .....	206
Capitolo 10 — Modalità principiante .....	207
Z1 — Importazione (scegliere immagini e preset) .....	207
Z2 — Elaborazione (SfM + training) .....	215
Z3 — Anteprima (ruotare il modello 3D) .....	221
Z4 — Esportazione (scegliere formato e salvare) .....	224
Passaggio a Modalità esperto .....	229
Domande frequenti .....	230

# Come leggere questo manuale

---

Ogni voce del manuale segue lo stesso schema. Sul lato sinistro trovi i percorsi operativi e i dettagli tecnici; a destra, in una colonna laterale calda, trovi sempre la spiegazione in parole semplici. Piccole icone all'inizio di ogni riga ti dicono a colpo d'occhio quale tipo di informazione segue.

## LE QUATTRO ICONE



**Dove si trova?** Il percorso concreto di clic attraverso l'app — barra dei menu, sezione dell'Inspector o passaggio della Modalità principiante. Anche le scorciatoie da tastiera associate sono qui. L'icona è un segnaposto e mostra dove si trova la funzione nell'interfaccia.



**Dettagli.** Valori predefiniti, intervalli di valori e percorsi del codice. Lo incontrerai soprattutto nelle impostazioni di training, che non sono voci di menu ma parametri numerici. L'icona mostra una piccola scheda tecnica.



**Tecnico.** Cosa fa internamente la funzione, quali parametri agiscono, a cosa reagisce e quali effetti collaterali ha. Per i lettori che vogliono capire cosa succede dietro le quinte. L'icona è un blocco di cursori e rappresenta simbolicamente le manopole sotto il cofano.



**In parole semplici.** Il messaggio chiave in parole chiare — senza gergo, senza codice. Leggi prima questa sezione se vuoi solo sapere rapidamente a cosa serve una funzione e quando usarla. L'icona è una nuvoletta di dialogo e sta per «in poche parole». Questa colonna è sempre su uno sfondo color sabbia caldo così che l'occhio la trovi subito.

## COLORI DEI CAPITOLI

Ogni capitolo ha un proprio colore di accento che riconoscerai dalla targhetta ID (ad esempio **M1**) alla sinistra di ogni titolo di voce e dalle piccole icone davanti ad essi. Sfogliando, puoi vedere a colpo d'occhio in quale capitolo ti trovi.

- |                 |                       |                       |                                 |                   |
|-----------------|-----------------------|-----------------------|---------------------------------|-------------------|
| <b>1</b> Menu   | <b>2</b> Inspector    | <b>3</b> Impostazioni | <b>4</b> Finestre ausiliarie    | <b>6</b> Training |
| <b>7</b> Preset | <b>8</b> Esportazioni | <b>9</b> SfM          | <b>10</b> Modalità principiante |                   |

#### CONSIGLI DI NAVIGAZIONE

**Avvio rapido.** Se ti interessa solo come usare l'app, vai direttamente al **Capitolo 10 — Modalità principiante**. È la variante guidata in quattro passaggi e non richiede alcuna conoscenza pregressa.

**Approfondimento.** I **Capitolo 2 — Inspector** e **Capitolo 7 — Preset** spiegano i controlli e i profili di qualità preconfigurati a tua disposizione in Modalità esperto.

**Consultazione.** L'indice e la ricerca a tutto testo del PDF ti aiutano a trovare una funzione specifica. Non devi leggere il manuale dalla prima all'ultima pagina.

# Introduzione — Cosa devi sapere

---

## Cos'è RadianceKit?

RadianceKit è un'app nativa per macOS che trasforma una serie di fotografie ordinarie o un video in una ricostruzione 3D esplorabile. L'input è, ad esempio, da 50 a 500 scatti che hai fatto intorno a un oggetto, in una stanza o attraverso un paesaggio. L'output è quella che viene chiamata scena Gaussian Splatting — un modello 3D che puoi visualizzare sul Mac in tempo reale da qualsiasi angolazione, che può essere esportato e incorporato in siti web, e che, nei suoi aspetti principali, sembra fotorealistico.

L'app funziona completamente in locale sul tuo Mac — nessuna immagine viene caricata nel cloud, non è richiesto alcun login e non c'è alcun abbonamento. Sfrutta intensamente la GPU del tuo Mac Apple Silicon (serie M): un training completo può durare da due minuti a diverse ore a seconda della scena e del preset. Mentre il calcolo è in corso, puoi continuare a lavorare normalmente sul Mac; RadianceKit continua in background e ti avvisa quando il risultato è pronto.

Ci sono due modalità operative: la *Modalità principiante* (Simple Mode) ti guida attraverso il flusso di lavoro Importazione → Scelta del preset → Training → Esportazione in quattro passaggi. La *Modalità esperto* (Expert Mode) apre un grande Inspector con tutti i controlli, una finestra di anteprima in tempo reale e grafici diagnostici. Puoi passare tra le modalità in qualsiasi momento; i dati nella scena vengono preservati.

## Cos'è il Gaussian Splatting?

Il Gaussian Splatting (spesso abbreviato come *3DGS* o semplicemente *Splatting*) è un metodo relativamente nuovo per la rappresentazione 3D fotorealistica, presentato nel 2023 in un paper di Graz e INRIA. L'idea: invece di modellare una scena come una classica mesh poligonale (triangoli) o una griglia di voxel, viene assemblata da milioni di piccoli, morbidi «fiocchi» 3D — ogni singolo fiocco è una distribuzione gaussiana 3D (da qui il nome) con la propria posizione, dimensione, forma, colore e trasparenza. Questi fiocchi vengono addestrati in modo che, presi insieme da ogni angolazione di visualizzazione delle tue foto di input, producano l'immagine corretta.

In pratica, ciò significa che il Gaussian Splatting può rappresentare riflessi, luci speculari, fogliame morbido, capelli o tende come la modellazione 3D classica non può — o può solo a costi enormi. In cambio, il risultato non è un modello 3D modificabile nel senso classico — non puoi semplicemente spostare un singolo muro o riposizionare un vaso. È più una *cattura congelata* dello spazio attraverso cui puoi muoverti liberamente. Per molte applicazioni — visualizzazione architettonica, presentazione di prodotti, tour virtuali, perizie forensi, patrimonio culturale — è esattamente il punto di forza giusto.

Sono necessari due passaggi per trasformare le immagini di input in una scena 3D. Innanzitutto, l'app calcola tramite un processo chiamato *Structure-from-Motion (SfM)* dove si trovava la tua fotocamera per ogni foto. Come sottoprodotto, ciò produce una nuvola di punti approssimativa della scena. Poi inizia il vero e proprio training di Gaussian Splatting: partendo da questa nuvola approssimativa, i milioni di fiocchi 3D vengono gradualmente distribuiti, ingranditi, raffinati e ri-sintonizzati nella posizione e nel colore finché non producono l'immagine corrispondente da tutti gli angoli di visualizzazione di input.

Non devi sapere nulla di tutto questo per usare RadiancKit. La Modalità principiante nasconde completamente questi passaggi. Ma se vuoi capire cosa significano i numeri diagnostici in Modalità esperto (iterazione, loss, gaussiane, SSIM ...) o perché alcune scene risultano più belle di altre, i capitoli successivi del manuale ti daranno le risposte.

## CAPITOLO

# Capitolo 1 — Barra dei menu

---

La barra dei menu di RadianceKit organizza tutte le funzioni che non si trovano direttamente nella finestra principale o nell'Inspector. Si tratta in primo luogo di azioni che agiscono sull'intera scena (aprire, salvare, nuovo progetto), che controllano il training (avviare, mettere in pausa, riprendere), che gestiscono il viewport (auto-rotazione, screenshot, colore di sfondo) e che attivano esportazioni in vari formati 3D e media. A ciò si aggiungono punti di salto verso tutte le finestre ausiliarie (User Guide, Pareto Dashboard, Holdout Analysis, BayesOpt Console).

Le scorciatoie da tastiera si trovano a destra di ogni voce di menu. Convenzioni: `⌘` significa il tasto Command (tasto Apple), `⇧` è Shift, `⌥` è Option (Alt) e `⌘` è Control. Esempio: `⇧⌘T` sta per Shift+Command+T. Tutte le scorciatoie qui documentate sono elencate in più tramite `Help` → `Keyboard Shortcuts` (`⌘/`) in una propria finestra di panoramica.

Le seguenti 42 voci sono documentate nell'ordine dell'inventario (M1–M42), raggruppate per il menu di primo livello corrispondente. Tutte le voci sono state verificate rispetto allo stato attuale del codice in (righe 175–477). Nessuna voce è stata rimossa o superata rispetto all'inventario; una nuova voce del menu Edit (Cmd-Z per «Remove Image») viene accolta tramite il framework di sistema `NSUndoManager` e non appare quindi nel codice `RadianceKitApp` (vedi nota alla fine del capitolo).

## Menu File

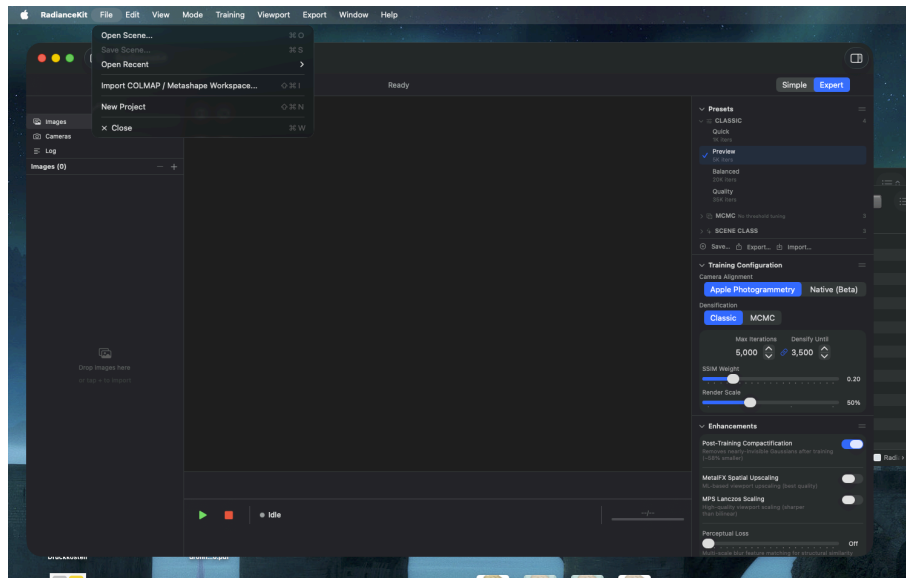


Figura 1: Menu File aperto — voci da M1 a M6

Il menu File sostituisce la voce standard «New Window» di Apple con azioni specifiche del progetto. Comprende caricare/salvare scene, una lista dinamica dei file recenti, l'importazione del workspace così come il reset duro a uno stato vuoto.

### M1 File > Open Scene...



Barra dei menu → File → Open Scene... (⌘O).



Apri una finestra di dialogo per i formati bundle `RadianceScene`, `.ply`, `.splat` e `.spz`. Selezione singola, può mostrare sia file sia directory (per il formato bundle). Dopo la selezione riuscita, il percorso viene aggiunto alla lista dei recenti e la scena viene caricata in modo asincrono — quella precedente viene sostituita e la pipeline di training iniziata con lo stato caricato. I file PLY/SPZ/Splat vengono letti tramite i rispettivi loader di formato; il bundle `.radiancescene` è una directory con manifest, snapshot della cloud e risultati SfM.

### IN PAROLE SEMPLICI

Così carichi nuovamente una scena già addestrata nell'app. Funziona con il formato proprio di RadianceKit e con i formati standard PLY, SPLAT e SPZ che altri programmi di splatting generano. Usalo quando ad esempio hai addestrato una scena durante la notte e il giorno dopo vuoi continuare o esportare. All'apertura lo stato precedente nella finestra principale viene sostituito — salva prima, quindi, se la scena attuale ti sta ancora a cuore. Il percorso finisce automaticamente in «Open Recent» (M3), così alla prossima volta ci arrivi più velocemente.

**M2** File > Save Scene...

Barra dei menu → File → Save Scene... (⌘S).

 **TECNICO**

Apri una finestra di dialogo di salvataggio con il tipo di contenuto bundle `RadianceScene` e nome file precompilato `scene.radiancescene`. Scrivi un pacchetto di directory con `manifest.json`, la nuvola gaussiana serializzata (snapshot PLY) e un dump del risultato SfM, in modo che dopo la riapertura funzioni anche il continue-training. La voce è disabilitata finché non esistono ancora gaussiane. Non salva nel percorso dei log di training, ma dove indica la finestra di salvataggio — tipicamente sotto `~/Documents/`.

 **IN PAROLE SEMPLICI**

Salva la tua scena attuale come file (più precisamente: come cartella pacchetto che assomiglia a un file). Solo dopo puoi riaprire questa scena in seguito con «Open Scene...» (M1). Nel pacchetto finiscono sia la nuvola gaussiana sia il risultato SfM, così puoi aggiungere in seguito anche il continue-training (M12–M14). Finché non hai concluso un training, la voce è in grigio. Il nome predefinito è `scene.radiancescene` — puoi però assegnare un nome proprio nella finestra di salvataggio.

**M3** File > Open Recent > [Nome scena]

Barra dei menu → File → Open Recent → (lista).

 **TECNICO**

Sottomenu dinamico generato da una lista dei percorsi aperti più di recente (salvati nelle impostazioni). Ogni voce della lista viene denominata con il nome del file e caricata al clic. Se la lista è vuota, appare invece l'etichetta disabilitata «No Recent Scenes». Tipico di Apple, la lista mantiene le N scene aperte più di recente — il limite avviene durante la scrittura nelle impostazioni e non nel builder del menu stesso.

 **IN PAROLE SEMPLICI**

Qui vedi le scene aperte più di recente e con un clic puoi rientrarci, senza dover passare dalla finestra di dialogo del file. Se hai appena iniziato, la lista è vuota e appare in grigio nel menu. Ogni scena che apri tramite «Open Scene...» (M1) finisce automaticamente in questa lista. Se a un certo punto la lista diventa troppo piena o vuoi svuotarla per motivi di privacy, usa «Clear Recent» (M4).

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

DOVE

Barra dei menu → File → Open Recent → Clear Recent.



TECNICO

Svuota la lista dei recenti nelle impostazioni. Effetto immediato, senza finestra di conferma. La voce appare nel sottomenu solo se ci sono in generale voci nella lista dei recenti (si trova sotto un divisore dopo i percorsi).

**IN PAROLE SEMPLICI**

Cancella la lista delle scene aperte più di recente. Pratico quando hai giocherellato con un dataset di test e non vuoi più vedere i percorsi. I file delle scene stessi non vengono eliminati — solo il collegamento nel menu. L'azione ha effetto immediato, senza domanda di conferma; dopo appare nel sottomenu «No Recent Scenes». La voce compare solo se ci sono in generale scene nella lista — con lista vuota non è visibile.

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

DOVE

Barra dei menu → File → Import COLMAP / Metashape Workspace... (⇧⌘I).



TECNICO

Apri un selettore di cartelle. Si aspetta una cartella con il layout del workspace COLMAP (ad es. `sparse/0/cameras.{bin,txt}` più `images/`). Dopo la selezione viene eseguita una pre-verifica del workspace — questa riconosce i tre layout (`sparse/0/`, `sparse/`, `root`) e se la ricostruzione è binaria (`cameras.bin`) o come testo ETH3D (`cameras.txt`). In caso di successo il workspace viene importato; altrimenti appare solo un avviso nel log dell'app. Vedi anche il Capitolo 9 «Backend SfM», Q6 per la logica completa della pipeline.

**IN PAROLE SEMPLICI**

Se usi Metashape, COLMAP, RealityCapture o un software simile per la ricostruzione delle fotocamere e hai un'esportazione, carichi la cartella qui dentro. RadianceKit salta allora lo stadio SfM e inizia direttamente con il training — questo risparmia ore su scene grandi. Il trascinamento sulla finestra principale funziona allo stesso modo. Ci si aspetta una cartella con layout COLMAP (cioè `sparse/0/` con `cameras.*` più cartella `images/`). Maggiori informazioni sui layout supportati e i flussi di lavoro sono nel Capitolo 9 «Backend SfM».

## M6 File > New Project



Barra dei menu → File → New Project (⇧⌘N).



Verifica se ci sono lavori non salvati. In caso affermativo appare una finestra di conferma prima che qualcosa vada perso. Se non c'è nulla da salvare, il reset avviene direttamente — svuota le immagini importate, il risultato SfM, la nuvola gaussiana, lo stato del training e tutti gli indicatori UI dipendenti. Attenzione: una libreria di preset creata dall'utente rimane preservata, perché si trova nelle impostazioni dell'app e non nello stato del progetto.

### IN PAROLE SEMPLICI

Resetta tutto a un inizio vuoto — come se avessi appena aperto l'app. Se hai ancora lavoro non salvato, l'app te lo chiede prima. Usalo quando vuoi iniziare con una scena completamente diversa. Immagini importate, risultato SfM, nuvola gaussiana e stato del training vengono completamente svuotati. I tuoi preset propri rimangono però preservati, perché si trovano nelle impostazioni dell'app e non appartengono alla scena.

## Menu Mode

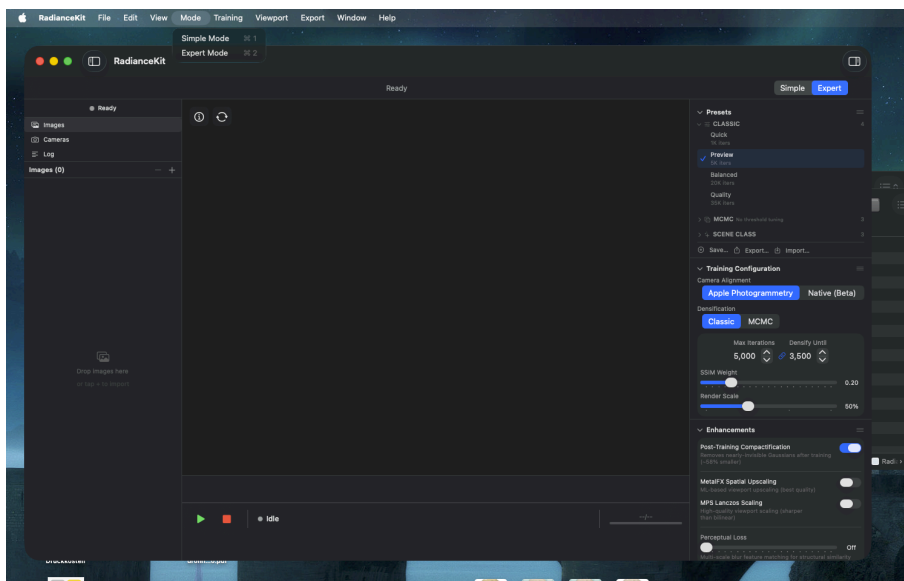


Figura 2: Menu Mode con switch Simple e Expert Mode

Due semplici switch tra la Modalità principiante guidata (tipo-wizard, 4 passaggi) e la Modalità esperto completa (layout Inspector classico con tutti i controlli).

**M7 Mode > Simple Mode**

Barra dei menu → Mode → Simple Mode (⌘1).



Commuta lo stato dell'app sulla Modalità principiante. L'area principale dell'app mostra allora il flusso di lavoro guidato invece del layout esperto. Lo stato della modalità viene salvato nelle impostazioni (vedi S1 «Default Mode» nel Capitolo 3 Impostazioni).

**IN PAROLE SEMPLICI**

Commuta alla variante passo dopo passo, in cui l'app ti guida attraverso importazione, elaborazione, anteprima ed esportazione. Consigliata se hai appena iniziato o se hai bisogno di un risultato veloce. La maggior parte dei controlli di dettaglio è nascosta — lavori con impostazioni predefinite sensate. Se in seguito vuoi approfondire, passa semplicemente alla Modalità esperto (M8). Quale modalità è attiva all'avvio dell'app puoi stabilirlo nelle impostazioni (Capitolo 3, S1).

**M8 Mode > Expert Mode**

Barra dei menu → Mode → Expert Mode (⌘2).



Commuta lo stato dell'app sulla Modalità esperto. In questo modo appare il layout Inspector completo con tutte le sezioni (Presets, TrainingConfig, Enhancements, Metrics, LossChart, ProjectNavigator). Nella Modalità esperto sono accessibili tutti i parametri di training, picker COLMAP, toggle Mid-Compact e diagnostiche. Anche la live preview funziona solo in questa modalità.

**IN PAROLE SEMPLICI**

Commuta alla vista completa con tutti i controlli. Qui vedi i grafici di loss in tempo reale, puoi regolare con precisione tutti i parametri e gestire più config di confronto in parallelo tramite preset. Consigliata se vuoi capire cosa fa il training internamente o se vuoi sperimentare in modo mirato. Anche la live preview, il picker COLMAP e le diagnostiche sono accessibili solo qui. Se ti senti sopraffatto, tramite M7 torna alla Modalità principiante — la tua scena rimane preservata.

## Menu Training

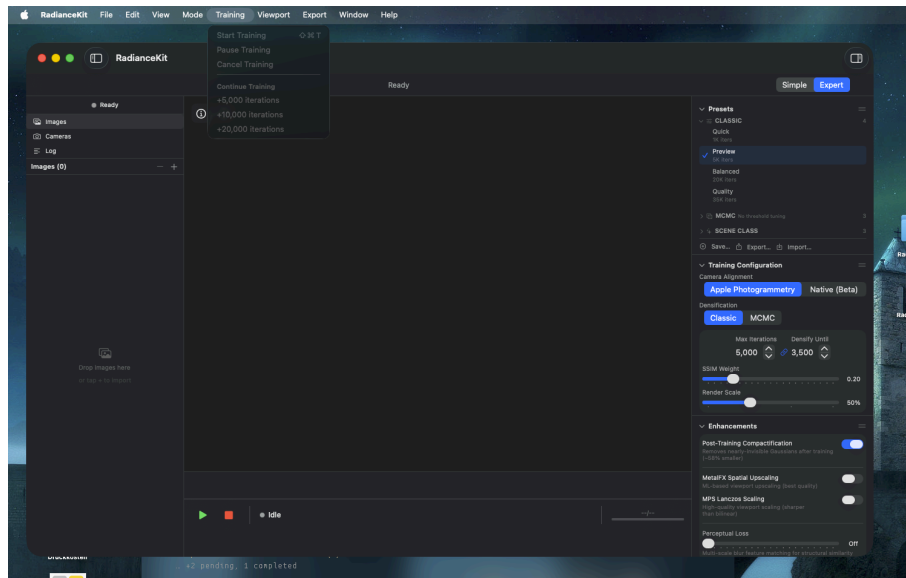


Figura 3: Menu Training con sottomenu Continue — voci da M9 a M14

Quattro azioni attorno all'esecuzione del training: avviare, mettere in pausa, interrompere e prolungare di un numero predefinito di iterazioni. Tutte e tre le voci Continue sono soggette a IAP-gating (non cliccabili nella versione free trial).

### M9 Training > Start Training



DOVE

Barra dei menu → Training → Start Training (⇧⌘T).



TECNICO

Avvia la pipeline di training in modo asincrono. Prerequisito: esiste un risultato SfM e attualmente non è in esecuzione un'altra pipeline. Entrambe le condizioni bloccano la voce se non soddisfatte. All'avvio vengono letti i valori di configurazione attuali, viene creato un nuovo log JSONL sotto `~/Documents/RadianceKit/Logs/training_YYYY-MM-DD_HHmss.jsonl`, e a seconda della scelta della strategia viene eseguito il percorso classico o quello MCMC. Lo stato del training passa da «idle» a «training».

#### IN PAROLE SEMPLICI

Preme il grande bottone verde — non appena hai importato foto e la ricostruzione delle fotocamere è terminata, inizia con esso il vero e proprio training di Gaussian Splatting. Lascia l'app in esecuzione; a seconda del preset tra 1 minuto (Quick) e diverse ore (MCMC Quality). La voce rimane in grigio finché non c'è ancora un risultato SfM o se attualmente è in esecuzione un'altra pipeline. Ogni passaggio scrive in più un log in `~/Documents/RadianceKit/Logs/`, che puoi poi valutare tramite il Pareto Dashboard (M40).

**M10 Training > Pause Training**

Barra dei menu → Training → Pause Training.



Mette in pausa il training in corso. Viene sbloccato solo se lo stato del training è «training». La pausa ferma il loop di iterazione al prossimo punto di sync di sicurezza, mantiene l'intero stato GPU (buffer gaussiani, momenti dell'optimizer, posizione dello scheduler) e passa a «paused». Il resume avviene tramite nuova pressione (il titolo della voce è statico — l'app passa però tra pause/resume nella logica effettiva). I training messi in pausa non sopravvivono a un quit dell'app; in tal caso salva piuttosto la scena ed estendila in seguito tramite la voce Continue Training (M12–M14).

**IN PAROLE SEMPLICI**

Ferma brevemente il training senza perdere i progressi. Pratico quando ti serve brevemente il computer per qualcosa di più importante. Cliccare nuovamente fa proseguire. Non funziona attraverso i riavvii dell'app — se vuoi davvero continuare in seguito, termina il training con Cancel (M11), salva la scena con Save Scene (M2) e poi usa Continue Training (M12–M14). Durante la pausa la GPU riposa completamente; la memoria rimane però occupata.

**M11 Training > Cancel Training**

Barra dei menu → Training → Cancel Training.



Interrompe il training in corso. Attivo se lo stato del training non è «idle». Imposta il flag di cancellazione nel motore di training, il che termina pulitamente il loop di iterazione al prossimo punto di sync, scrive la voce di summary finale nel log JSONL e ripristina lo stato a «idle». La nuvola finora addestrata rimane preservata (può essere salvata o esportata), ma viene contrassegnata come «cancelled».

**IN PAROLE SEMPLICI**

Interrompe definitivamente il training in corso. Lo stato precedente rimane — se dopo qualche migliaio di iterazioni hai già un risultato presentabile, puoi comunque esportarlo dopo. Se vuoi solo interrompere brevemente, usa invece Pause (M10). Nel log del training l'esecuzione viene contrassegnata come «cancelled», il valore di loss finale viene comunque scritto. Una scena interrotta puoi anche continuarla in seguito tramite Continue Training (M12–M14), purché l'app non sia stata terminata nel frattempo.

**M12 Training > Continue Training > +5 000 iterations**

Barra dei menu → Training → Continue Training → +5,000 iterations.



Continua il training di 5 000 iterazioni. Attivo se un training concluso è continuabile e la versione completa è sbloccata. La continuabilità vale se esiste un training concluso e l'intero stato dell'optimizer è ancora in memoria. Al continue, i momenti Adam e lo scheduler LR vengono proseguiti, in modo che la continuazione si comporti come un'esecuzione continua di 25K/45K/60K invece di un nuovo avvio. Il log JSONL riceve una nuova voce di config con il setup incrementale. Disponibile solo nella versione completa.

**IN PAROLE SEMPLICI**

Aggiunge altri 5 000 passi di training. Usalo quando il risultato dopo il primo passaggio è vicino ma non ancora abbastanza nitido. Funziona solo nella versione a pagamento completa. A differenza di un passaggio completamente nuovo, lo stato dell'optimizer rimane preservato, in modo che la continuazione si senta come un'esecuzione continua. Se hai bisogno di più di 5 000 passi, prendi direttamente M13 (+10 000) o M14 (+20 000).

**M13 Training > Continue Training > +10 000 iterations**

Barra dei menu → Training → Continue Training → +10,000 iterations.



Identico a M12, ma con 10 000 iterazioni aggiuntive. Stessi prerequisiti, stesso percorso dello scheduler LR. Consigliato se il training iniziale è stato fatto con un preset di livello intermedio e vuoi vedere un miglioramento significativo della qualità senza dover ricominciare l'esecuzione da zero.

**IN PAROLE SEMPLICI**

Prolunga il training di 10 000 passi — il valore intermedio dei tre valori Continue disponibili. Buona scelta se il primo passaggio è andato bene, ma vuoi diventare chiaramente migliore. Come M12 e M14, il percorso dei learning rate viene proseguito senza interruzioni, invece di ricominciare. Disponibile solo nella versione completa.

## M14 Training > Continue Training > +20 000 iterations



Barra dei menu → Training → Continue Training → +20,000 iterations.



Identico a M12 / M13, ma con 20 000 iterazioni aggiuntive. Il più grande salto Continue predefinito. Per i training MCMC è spesso ciò che fa la differenza tra «funziona» e «adatto al benchmark»; con Classic da 35–40K spesso non viene più aggiunto molto secondo l'esperienza.

### IN PAROLE SEMPLICI

Aggiunge 20 000 passi di training aggiuntivi, il valore Continue massimo. Usalo quando vuoi davvero tirare fuori l'ultimo briciolo di qualità. Con il training classico dopo 40 000 passi spesso non porta più molto — con MCMC invece vale spesso la pena, perché lì la convergenza si manifesta più lentamente. Conta su un tempo di esecuzione decisamente più lungo a seconda della scena. Come M12 e M13, anche questa voce è disponibile solo nella versione completa.

## Menu Viewport

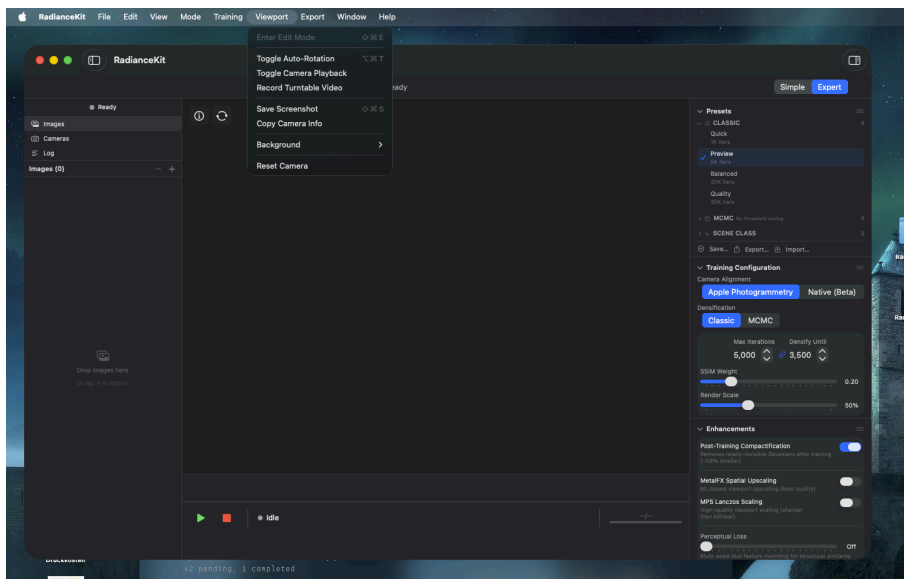


Figura 4: Menu Viewport con modalità modifica, controllo fotocamera e sottomenu sfondo

Controlla il viewport 3D: modalità modifica per selezione e cleanup delle gaussiane, controllo della fotocamera (auto-rotazione, playback, recording), screenshot, colore di sfondo e reset.

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

Barra dei menu → Viewport → Enter Edit Mode (o «Exit Edit Mode», a seconda dello stato). ⌘⌘E.



Il titolo della voce è dinamico e mostra «Exit Edit Mode» o «Enter Edit Mode» a seconda dello stato. Alla pressione, la modalità modifica viene commutata sul renderer del viewport. All'uscita dalla modalità modifica, viene inoltre resettata la selezione attuale. La modalità modifica attiva la selezione tramite clic sulle gaussiane, la selezione tramite box e l'eliminazione delle gaussiane contrassegnate (vedi l'area Editor della UI). Disabilitata finché non è collegato alcun renderer del viewport.

**IN PAROLE SEMPLICI**

Commuta tra la vista 3D normale e una modalità di modifica in cui puoi contrassegnare ed eliminare singole gaussiane (ad es. floater o outlier sullo sfondo). All'uscita la selezione viene automaticamente resettata. La voce rimane in grigio finché non è ancora visibile alcuna scena nel viewport. L'etichetta passa a seconda dello stato tra «Enter Edit Mode» ed «Exit Edit Mode» — vedi quindi sempre in quale modalità ti trovi attualmente.

**M16 Viewport > Toggle Auto-Rotation**

Barra dei menu → Viewport → Toggle Auto-Rotation (⌘⌘T).



Commuta on o off la rotazione continua della fotocamera del viewport attorno a un asse verticale attraverso il centro della scena. L'asse e la velocità provengono dalla configurazione di controllo della fotocamera. L'auto-rotazione è un effetto puro del viewport e non influenza né il training né il recording — se usi in parallelo il registratore video turntable (M18), l'auto-rotazione fornisce però esattamente il percorso che il registratore cattura.

**IN PAROLE SEMPLICI**

Ruota la fotocamera continuamente attorno alla tua scena, in modo che tu possa vederla da tutti i lati senza dover trascinarla con il mouse. Cliccare nuovamente ferma la rotazione. Pratico nella valutazione di scene già addestrate o come animazione di sfondo per una demo dal vivo. Se registri in parallelo un video (M18), l'auto-rotazione fornisce esattamente il movimento che il registratore cattura.

**M17 Viewport > Toggle Camera Playback**

Barra dei menu → Viewport → Toggle Camera Playback.



Commuta il playback del percorso della fotocamera. Se esiste un percorso registrato (ad es. da un recording precedente o perché è stato caricato un `transforms.json`), il percorso viene riprodotto — la fotocamera del viewport non si muove più dunque secondo input di mouse/trackpad, ma riproduce la traiettoria frame per frame. Nuova pressione mette in pausa il playback.

**IN PAROLE SEMPLICI**

Fa partire una corsa della fotocamera registrata o importata in precedenza. Così puoi seguire il percorso originale con cui la scena è stata ripresa, o verificare un movimento orbit pianificato prima dell'esportazione video. Mentre il playback è in esecuzione, gli input di mouse e trackpad sono disabilitati — la fotocamera segue rigorosamente il percorso. Cliccare nuovamente mette in pausa la riproduzione. Se non hai caricato o registrato alcun percorso della fotocamera, non succede nulla.

**M18 Viewport > Record Turntable Video**

Barra dei menu → Viewport → Record Turntable Video.



Commuta la registrazione del viewport. Alla prima pressione inizia una registrazione di frame in un percorso temporaneo; alla seconda pressione la registrazione viene terminata, codificata e scritta in un percorso MP4 (il percorso viene chiesto tramite una finestra di salvataggio). A differenza dell'Export → Media → Orbit Video (M31), che genera un percorso fisso a 360° con una durata impostabile, il registratore turntable cattura *in tempo reale* ciò che vedi nel viewport — puoi quindi registrare anche una corsa manuale della fotocamera.

**IN PAROLE SEMPLICI**

Registra un video direttamente nel viewport. Indipendentemente dal fatto che la fotocamera ruoti automaticamente o che la sposti tu stesso con il mouse — tutto ciò che vedi viene salvato in un file MP4. A differenza dell'esportazione «Orbit Video» (M31), sei tu stesso a indicare la corsa della fotocamera. Il primo clic avvia la registrazione, il secondo clic la termina e ti chiede dove salvare. Pratico se vuoi mostrare ad esempio una specifica inquadratura di dettaglio, che non sarebbe possibile con il movimento orbit rigido.

**M19 Viewport > Save Screenshot**

DOVE

Barra dei menu → Viewport → Save Screenshot (⇧⌘S).



TECNICO

Cattura un singolo frame del viewport a piena risoluzione di rendering (quindi non il layout dei pixel della finestra, ma l'intero contenuto del render target) come file PNG. Il percorso viene chiesto tramite una finestra di salvataggio. Il colore di sfondo (M21–M23) viene impresso. Le impostazioni di upscaling MetalFX/MPS dagli Enhancements (vedi I27/I28) hanno effetto se attive — lo screenshot mostra quindi l'output sovracampionato.

**IN PAROLE SEMPLICI**

Salva uno scatto della tua vista 3D attuale come immagine PNG. Pratico per materiale di marketing o per un confronto veloce. Nota: lo sfondo fa parte dell'immagine — se hai bisogno di trasparenza, esporta piuttosto un file di scena. La risoluzione corrisponde al render target interno, non alla dimensione della tua finestra — l'immagine è quindi spesso più nitida di quanto sembri nella finestra. Eventuali impostazioni di upscaling (Inspector → Enhancements) confluiscono allo stesso modo.

**M20 Viewport > Copy Camera Info**

DOVE

Barra dei menu → Viewport → Copy Camera Info.



TECNICO

Legge la posa attuale della fotocamera del viewport (posizione, punto Look-At, vettore Up) e i valori FOV dal controllo fotocamera e li scrive come testo multi-riga negli appunti. Il formato è leggibile dall'uomo (etichetta = valore per riga), non JSON. Pratico per riprodurre una specifica vista a scopo di debug o condividerla con il supporto.

**IN PAROLE SEMPLICI**

Copia la posizione attuale della fotocamera e la direzione di vista come testo negli appunti. Se vuoi mostrare ad es. a un co-sviluppatore da dove un punto nella scena sembra strano, incollati semplicemente il testo in una mail o in una finestra di chat. Il formato è leggibile dall'uomo (una riga per valore), non JSON. Principalmente pensato per bug report o richieste al supporto.

**M21 Viewport > Background > Dark Gray**

DOVE

Barra dei menu → Viewport → Background → Dark Gray.



TECNICO

Imposta il colore di sfondo del viewport su un grigio scuro (RGB 0.1/0.1/0.1). Il renderer utilizza questo colore come sfondo davanti al quale le gaussiane vengono composte. Il colore predefinito all'avvio dell'app è controllato dall'opzione Settings S3 «Default Viewport Background».

**IN PAROLE SEMPLICI**

Colora lo sfondo del viewport 3D di grigio scuro. Scelta standard per la maggior parte delle scene — offre un buon contrasto sia con gaussiane chiare sia scure, senza che l'occhio si fissi su un piano completamente nero o bianco. Il colore viene anche assunto in screenshot (M19) e video orbit (M31). Se Dark Gray ti sembra troppo poco spettacolare, prova anche Black (M22) o White (M23) per confronto. Quale colore è attivo all'avvio dell'app puoi stabilirlo nelle impostazioni (S3).

**M22 Viewport > Background > Black**

DOVE

Barra dei menu → Viewport → Background → Black.



TECNICO

Imposta il colore di sfondo del viewport su nero puro (RGB 0/0/0). Aiuta se la scena ha molti floater chiari e vuoi identificarli, o per materiale di marketing con look-and-feel scuro.

**IN PAROLE SEMPLICI**

Sfondo nero. Buono per scene molto luminose o quando vuoi guardare nella modalità modifica e cerchi piccole gaussiane chiare (floater) che si perdono nel grigio. Anche ideale per materiale di marketing con look scuro e drammatico. Il colore viene impresso in screenshot e video orbit — se hai bisogno di trasparenza per un compositing successivo, il nero è la scelta peggiore. Per floater scuri passa nuovamente nell'altra direzione a White (M23).

**M23 Viewport > Background > White**

Barra dei menu → Viewport → Background → White.



Imposta il colore di sfondo del viewport su bianco puro (RGB 1/1/1). Utile se la scena ha prevalentemente contenuti scuri e vuoi vedere floater scuri (rumore di sfondo tipico outdoor).

**IN PAROLE SEMPLICI**

Sfondo bianco. Pratico se il soggetto chiaro-su-scuro emerge meglio o per trovare outlier scuri che puoi poi rimuovere in modalità modifica (M15). Nelle scene outdoor il bianco è spesso più utile del nero, perché i tipici floater outdoor sono piuttosto scuri. Come per le altre opzioni di sfondo, il colore viene assunto in screenshot e video.

**M24 Viewport > Reset Camera**

Barra dei menu → Viewport → Reset Camera.



Resetta la fotocamera del viewport, esce dalla vista fotocamera del training e ferma l'auto-rotazione. In questo modo la fotocamera torna alla posizione iniziale (tipica: davanti alla scena, guardando leggermente dall'alto), l'auto-rotazione è spenta, e se il renderer stava mostrando una delle pose SfM (la training camera), torna alla free-camera.

**IN PAROLE SEMPLICI**

Riporta la fotocamera del viewport nella posizione di partenza. Se ti sei perso ruotando o hai spinto la scena fuori dall'inquadratura — clicca qui una volta e vedi di nuovo ciò che dovresti vedere. Allo stesso tempo spegne l'auto-rotazione se attualmente in esecuzione, e torna da una fotocamera del training congelata alla vista libera. Così ottieni in ogni caso un riavvio pulito della vista.

## Menu Export

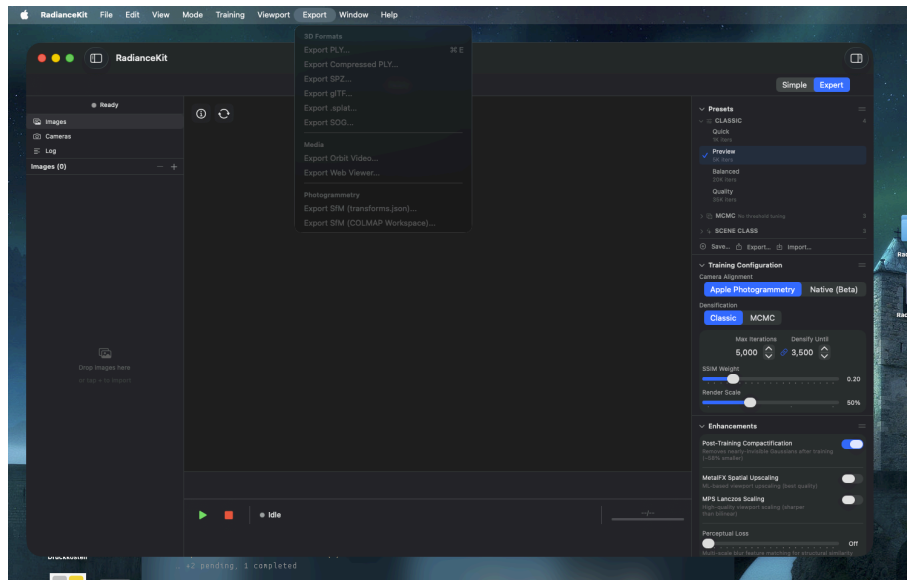


Figura 5: Menu Export con tre gruppi di sottomenu — 3D Formats, Media e Photogrammetry

Otto destinazioni di esportazione più due esportazioni Photogrammetry, raggruppate in tre sezioni (3D Formats, Media, Photogrammetry). I primi sei vengono costruiti tramite una routine helper comune, che apre rispettivamente una finestra di salvataggio e registra l'esportazione nel catalogo dei formati. Le voci Photogrammetry hanno logica individuale. Tutte le voci Photogrammetry e alcune esportazioni 3D sono disponibili solo nella versione completa.

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



Barra dei menu → Export → 3D Formats → PLY (⌘E).



Aprire una finestra di salvataggio con nome file predefinito `gaussians.ply`. Al clic su OK, la nuvola gaussiana attuale viene scritta nel formato PLY ASCII/binario standardizzato — compatibile con SuperSplat, PolyCam, PlayCanvas e tutti i viewer 3DGS comuni. Coefficienti SH completi, precisione completa (Float32 per campo). Dimensione del file spesso diversi centinaia di MB con  $\geq 500K$  gaussiane.

#### IN PAROLE SEMPLICI

Salva la tua scena 3D come file PLY standard. È il formato più universale — quasi ogni software può caricarlo, da SuperSplat passando per PolyCam fino a PlayCanvas. I file diventano però grandi, spesso diverse centinaia di megabyte. Usa PLY se vuoi continuare a piena qualità o archiviare. Se vuoi condividere la scena tramite web, dai un'occhiata piuttosto a SPZ (M27) o Compressed PLY (M26) — sono notevolmente più piccoli.

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

Barra dei menu → Export → 3D Formats → Compressed PLY.



Scrive la nuvola gaussiana nel formato Compressed PLY con quantizzazione personalizzata dei campi posizione, scala, rotazione e SH. File 5–10× più piccoli del PLY non compresso (M25) con perdite visive minime. Compatibile con SuperSplat (che legge lo standard Compressed PLY) e PlayCanvas. Nome file predefinito `gaussians_compressed.ply`.

 IN PAROLE SEMPLICI

Come il PLY normale, ma 5–10 volte più piccolo. La qualità rimane quasi uguale. Usalo se vuoi condividere il file online o lo invii per email. Funziona direttamente con SuperSplat e PlayCanvas. Se il tuo sistema di destinazione ha però bisogno di file ancora più piccoli (mobile, demo browser), prendi invece SPZ (M27) — è compresso in modo ancora più aggressivo. Per la piena qualità di modifica prendi il PLY non compresso (M25).

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

Barra dei menu → Export → 3D Formats → SPZ.



Scrive la nuvola gaussiana nel formato SPZ — il formato splat compresso pubblicato da Niantic con quantizzazione aggressiva (~90 % più piccolo del PLY non compresso). Ottimizzato soprattutto per viewer web e app mobili. Compatibile con Niantic Splatt3R, `gsplat.js` e il viewer browser Niantic.

 IN PAROLE SEMPLICI

Uno dei formati più piccoli. Circa 10× più piccolo di un PLY normale. Usalo soprattutto se vuoi mostrare la scena in un browser o visualizzarla tramite app per smartphone. Per la massima qualità il PLY è la scelta migliore. SPZ è sviluppato da Niantic e funziona direttamente con `gsplat.js`, Splatt3R e il viewer web Niantic. A causa della forte compressione, non puoi più riallenare facilmente i file SPZ — per la modifica prendi PLY.

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

Barra dei menu → Export → 3D Formats → glTF.



Scrive un file `.glb` (Binary-glTF) con l'estensione `KHR_gaussian_splatting`. Conforme allo standard, adatto a pipeline che usano motori glTF come Babylon.js o Three.js e che implementano l'estensione `KHR_gaussian_splatting`.

**IN PAROLE SEMPLICI**

Salva la scena nel formato glTF, che molti programmi 3D e motori web comprendono — a condizione che supportino l'estensione Gaussian Splatting. Se hai una pipeline 3D specifica (ad es. Three.js o Babylon.js) che la comprende, è il tuo formato. Il file esce come `.glb` binario — un singolo pacchetto che contiene tutto. Per i flussi di lavoro Splatting classici la scelta migliore è solitamente PLY o SPZ, perché più strumenti li comprendono direttamente.

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

Barra dei menu → Export → 3D Formats → .splat.



Scrive il formato `.splat` di Antimatter15 — fixed-size 32 byte per gaussiana (posizione come  $3 \times \text{Float32}$ , scala come  $3 \times \text{Float32}$ , rotazione come quaternion normalizzato  $4 \times \text{UInt8}$ , RGB+opacità come  $4 \times \text{UInt8}$ ). Nessun coefficiente SH superiore al DC. Il file più piccolo con compatibilità diretta browser. Per `gsplat.js` e il viewer demo online di `antimatter15`.

**IN PAROLE SEMPLICI**

Il formato di viewer web più semplice. Piccolo e immediatamente visualizzabile in ogni browser. Perde però l'illuminazione di dettaglio (i coefficienti SH superiori vanno persi — lo splat appare uguale da ogni angolazione, invece di reagire alla luce). Buono per le massime prestazioni web, per il foto-realismo piuttosto SPZ o PLY. Funziona con il viewer online di `antimatter15` e `gsplat.js`. Ogni gaussiana occupa fissi 32 byte, il che rende il formato semplice e compatibile — ma al prezzo della profondità di dettaglio.

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

Barra dei menu → Export → 3D Formats → SOG.



Scrive la nuvola gaussiana nel formato SOG. SOG («Self-Organizing Gaussian») è il formato PlayCanvas con layout texture atlas e compressione WebP dei dati quantizzati. Scala con un rapporto di dimensione 15–20× migliore del PLY. L'esportazione invoca internamente `cwebp` come strumento esterno — quindi potenzialmente limitato nella variante sandbox (App Store).

 IN PAROLE SEMPLICI

Formato molto piccolo per i flussi di lavoro PlayCanvas. Circa 15–20 volte più piccolo del PLY, perché i dati vengono impacchettati in un layout texture atlas e compressi con WebP. Se non hai un flusso di lavoro PlayCanvas, SPZ o Compressed PLY sono solitamente la scelta migliore. L'esportazione invoca internamente `cwebp` come strumento esterno — nella versione App Store (sandbox) questo passaggio può essere limitato.

**M31** Export > Media > Export Orbit Video...

Barra dei menu → Export → Media → Orbit Video.



Renderizza un orbit a 360° attorno al centro della scena e lo codifica come MP4 (H.264) o MOV (HEVC, a seconda del default di sistema). A differenza di M18 (recording dal vivo), il percorso qui è predefinito — la durata viene scelta nelle impostazioni o nello step di esportazione della Modalità principiante.

 IN PAROLE SEMPLICI

Genera automaticamente un video di rotazione attorno alla tua scena. Nessun movimento manuale necessario. Buono per i social media o una demo veloce. Se vuoi controllare tu stesso la fotocamera, usa invece Record Turntable Video (M18). Il percorso è fisso: un orbit completo a 360° attorno al centro della scena, durata che scegli nelle impostazioni o nello step di esportazione della Modalità principiante. Il video viene emesso a seconda del sistema come MP4 H.264 o MOV HEVC.

**M32** Export > Media > Export Web Viewer...

Barra dei menu → Export → Media → Web Viewer.



Impacchetta un viewer HTML standalone (basato su gsplat.js) più i dati gaussiani codificati in base64 in un singolo file `.html`. Questo file viene eseguito offline in qualsiasi browser moderno — nessuna dipendenza da server, nessun URL esterno. La dimensione del file è circa fattore 1.3 più grande della variante SPZ (per l'overhead di base64).

**IN PAROLE SEMPLICI**

Salva la tua scena come pagina web auto-avviabile. Doppio clic sul file HTML → si apre il browser → scena 3D interattiva pronta. Funziona senza internet, può essere inviata per mail, è il modo più semplice per condividere il risultato con amici o clienti. Il file contiene l'intero viewer gsplat.js e i dati gaussiani in un singolo documento — nulla viene caricato dal web. La dimensione del file è circa un terzo più grande di un'esportazione SPZ, in cambio non hai bisogno di software aggiuntivo presso il destinatario.

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

Barra dei menu → Export → Photogrammetry → Export SfM (transforms.json).



Percorso di esportazione proprio (non tramite la routine helper comune), perché non viene esportata una nuvola gaussiana ma il risultato SfM. Apre una finestra di salvataggio con `transforms.json` come predefinito e tipo di contenuto `json`. Al clic su OK viene scritto un `transforms.json` compatibile con nerfstudio con intrinseci della fotocamera, pose (come matrice 4x4 in convenzione NeRF) e percorsi dei frame. Il testo di aiuto nell'UI segnala che le immagini di training devono essere copiate come cartella sibling `images/`. Attivo solo se esiste un risultato SfM e la versione completa è sbloccata.

**IN PAROLE SEMPLICI**

Se vuoi continuare a usare il risultato SfM in un altro software come nerfstudio, Brush, gsplat o OpenSplat, esporti qui le posizioni della fotocamera. Metti le tue immagini di training in più in una cartella `images/` accanto al file `transforms.json` — altrimenti il programma di destinazione non può associare le immagini. La voce è in grigio finché non esiste ancora alcun risultato SfM, e bloccata nella versione free trial. Per il flusso di lavoro del workspace COLMAP prendi invece M34.

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



Barra dei menu → Export → Photogrammetry → Export SfM (COLMAP Workspace).



Apri una finestra di salvataggio con nome predefinito `colmap-workspace` (senza estensione, perché è una cartella). Scrive un workspace COLMAP standard con `sparse/0/cameras.bin`, `images.bin`, `points3D.bin`. Permette di aprire una ricostruzione SfM calcolata o importata in RadianceKit in altri strumenti come Postshot, Nerfstudio o Meshroom, o di ricaricarla in caso di re-run A/B come input già calcolato in RadianceKit stesso (tramite M5) — risparmia tempo di calcolo. Attivo solo se esiste un risultato SfM e la versione completa è sbloccata.

#### IN PAROLE SEMPLICI

Come M33, ma nel formato COLMAP invece di nerfstudio. Se usi Postshot, Meshroom, Nerfstudio o un altro strumento con flusso di lavoro COLMAP, è la tua esportazione. Un effetto collaterale pratico: puoi ricaricare questa cartella in seguito tramite M5 in RadianceKit e risparmiarti il tempo di calcolo SfM al prossimo passaggio — soprattutto su scene grandi un risparmio di tempo di ore. Come M33, disponibile solo se esiste un risultato SfM, e bloccata nella versione free trial.

## Menu Help

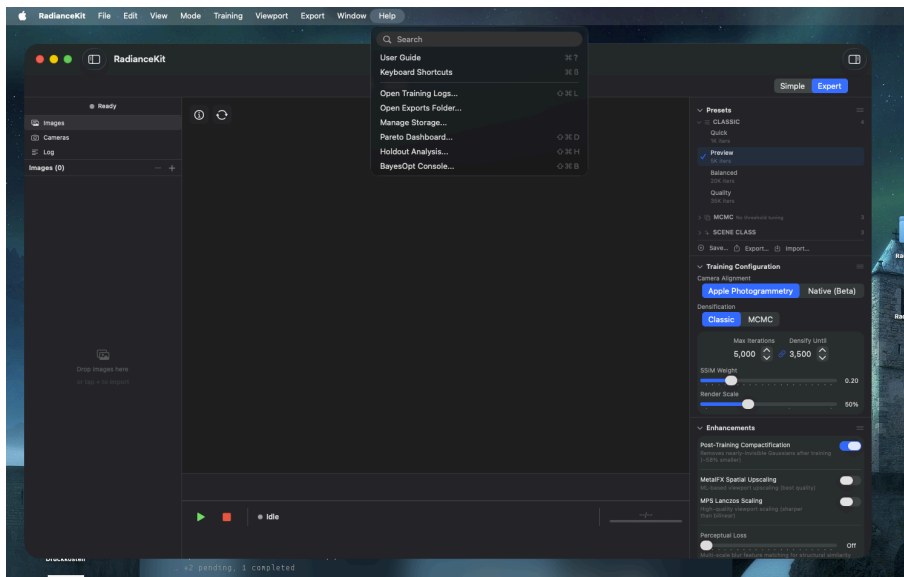


Figura 6: Menu Help con voci di documentazione, cartelle e analisi

Sette voci: due finestre di documentazione (User Guide, Keyboard Shortcuts), tre scorciatoie per cartelle (Training Logs, Exports, Storage) e tre finestre di analisi (Pareto Dashboard, Holdout Analysis, BayesOpt Console). Tipico di Apple, il menu Help appare all'estrema destra. Il menu Help standard viene completamente sostituito dalla variante propria di RadianceKit.

**M35 Help > User Guide**

Barra dei menu → Help → User Guide (⌘?).



Apri la finestra User Guide. Mostra una navigazione con sidebar dei temi e area di dettaglio con scroll a dimensione predefinita 860×640. I contenuti sono inseriti staticamente (non parsati da Markdown).

 **IN PAROLE SEMPLICI**

Apri il manuale interno all'app. Se non vuoi rileggere tutto in questo manuale, ci trovi i passaggi più importanti direttamente nel programma. Il manuale è costruito come finestra propria con sidebar dei temi — puoi quindi saltare in modo mirato a singoli temi. I contenuti sono più brevi di questo manuale e si concentrano sui flussi di lavoro più frequenti.

**M36 Help > Keyboard Shortcuts**

Barra dei menu → Help → Keyboard Shortcuts (⌘/).



Apri la finestra Keyboard Shortcuts — un semplice layout a scorrimento con tutte le scorciatoie dell'app, raggruppate per menu di primo livello. Dimensione predefinita 440×560. I contenuti sono allo stesso modo inseriti staticamente.

 **IN PAROLE SEMPLICI**

Apri una finestra con la lista completa di tutte le scorciatoie da tastiera. Se ad es. non ti ricordi con quale tasto si avvia il training, ci dai un'occhiata. Un riepilogo si trova anche alla fine di questo capitolo. La lista è raggruppata per menu di primo livello, in modo che tu salti rapidamente all'area giusta. Utile quando ti stai abituando dallo stile mouse a quello tastiera.

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

Barra dei menu → Help → Open Training Logs... (⇧⌘L).



Calcola la cartella dei log come `~/Documents/RadianceKit/Logs`, la crea se necessario e la apre nel Finder. Ogni esecuzione del training vi scrive un proprio file JSONL `training_YYYY-MM-DD_HHmss.jsonl`.

 **IN PAROLE SEMPLICI**

Apri nel Finder la cartella con tutti i protocolli di training precedenti. Se qualcosa è andato storto o vuoi controllare quando esattamente il training è convergente su quale valore, lo trovi qui in file JSONL. Per ogni esecuzione del training viene creato esattamente un file con timestamp — puoi leggerlo anche in altri strumenti o inviarlo per mail al supporto. Se vuoi una valutazione grafica, il Pareto Dashboard (M40) è il punto di ingresso migliore.

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

Barra dei menu → Help → Open Exports Folder...



Analogo a M37, ma con `~/Documents/RadianceKit/Exports`. Viene creata alla prima esecuzione del test automatico o al primo clic; dopo vi finiscono i percorsi standard di tutte le esportazioni dei test automatici (ad es. `autotest_<timestamp>.ply`). Le esportazioni selezionate manualmente tramite la finestra di salvataggio NON finiscono necessariamente qui, ma dove l'utente le salva — quindi questa cartella è interessante soprattutto per i test automatici.

 **IN PAROLE SEMPLICI**

Apri la cartella in cui l'app deposita le proprie esportazioni (soprattutto i passaggi di test automatici). Se hai salvato un'esportazione manualmente da un'altra parte con la finestra di salvataggio, è lì e non in questa cartella. Pratico per riordinare o per controllare quanto spazio occupano le esportazioni di test precedenti. Se hai bisogno di una panoramica completa compresi log e bundle di scena, prendi invece Manage Storage (M39).

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

Barra dei menu → Help → Manage Storage...



Apri il browser di archiviazione (vedi il Capitolo 4 Finestre ausiliarie, ID W7–W12). Elenca tutte le scene persistenti, i log di training, le esportazioni e le cache nella cartella `~/Documents/RadianceKit/` con dimensione, permette Reveal-in-Finder e Move-to-Trash per voce.

 IN PAROLE SEMPLICI

Apri un browser a finestra che ti mostra quanto spazio occupa RadianceKit sul tuo disco — per scena, log ed esportazione. Puoi eliminare direttamente singole cose senza dover andare nel Finder. Pratico dopo un uso prolungato, quando il disco si riempie — log precedenti ed esportazioni di test automatici possono sommersi a diversi gigabyte. Tramite Reveal-in-Finder arrivi in qualsiasi momento anche alla vista classica.

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

Barra dei menu → Help → Pareto Dashboard...  
(⇧⌘D).



Apri il Pareto Dashboard (vedi Capitolo 4, ID W13–W22). Il dashboard carica tutti i log JSONL di training da `~/Documents/RadianceKit/Logs/`, li ordina per scena e preset e disegna uno scatter plot di Pareto (default: loss vs gaussiane, opzionale loss vs wallclock o PSNR vs iterazioni).

 IN PAROLE SEMPLICI

Apri una panoramica di tutte le esecuzioni di training precedenti come diagramma. Vedi subito quale esecuzione ha fornito il miglior equilibrio tra qualità e dimensione. Pratico se vuoi confrontare diversi preset tra loro. Per default il diagramma mostra loss vs numero di gaussiane — puoi però passare anche a tempo wallclock o PSNR. I dati provengono dai log JSONL di training (M37); più esecuzioni hai, più significativa diventa la valutazione.

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

Barra dei menu → Help → Holdout Analysis...  
(⇧⌘H).



Apri la finestra di analisi Holdout (vedi Capitolo 4, ID W23–W29). Carica un `transforms.json`, disegna le fotocamere come globo 3D e permette split train/test fold (angolare o lineare, 2–8 fold). L'output è un `fold-assignment.json` che il training può utilizzare nelle rispettive configurazioni di training come set di test.

**IN PAROLE SEMPLICI**

Ti aiuta a suddividere le tue riprese delle fotocamere in set di training e di test — in modo che tu possa misurare in modo oggettivo quanto è buona la tua scena (su immagini che il training non ha visto). Piuttosto uno strumento di ricerca e benchmark. Le fotocamere vengono rappresentate come globo 3D; puoi scegliere tra 2 e 8 fold, uniformemente per angolo o linearmente per ordine. Il risultato è un piccolo file JSON che il training utilizza poi come set di test.

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

Barra dei menu → Help → BayesOpt Console...  
(⇧⌘B).



Apri la console BayesOpt (vedi Capitolo 4, ID W30–W39). Carica spazi di ricerca predefiniti (ad es. «MCMC scale-reg + opacity-reg + ssim»), esegue trial di ottimizzazione bayesiana in modo asincrono e mostra la curva di convergenza e il log dei trial in tempo reale.

**IN PAROLE SEMPLICI**

Una console di auto-tuning integrata. Invece di provare manualmente vari parametri, l'app può eseguirlo da sola durante la notte e proporti alla fine i valori migliori per la tua scena. Strumento molto avanzato — per la maggior parte dei flussi di lavoro un buon preset (vedi Capitolo 7) è sufficiente. Scegli uno spazio di ricerca predefinito (ad es. «MCMC scale-reg + opacity-reg + ssim») e vedi in tempo reale la curva di convergenza così come il log dei trial. Pianifica da diverse ore a giorni a seconda del setup.

**Nota: Cmd-Z nel menu Edit**

Da maggio 2026 il Project Navigator nella Modalità esperto supporta l'eliminazione delle immagini importate tramite il pulsante meno o il tasto backspace e l'annullamento tramite `Cmd-Z`. Questa azione `Cmd-Z` appare nel menu Edit di macOS (fornito da SwiftUI) come «Undo Remove Image» finché un'immagine eliminata è ancora ripristinabile. Viene registrata tramite il sistema standard conforme, non in; quindi non c'è una voce M-ID propria nell'inventario.

## Riepilogo delle scorciatoie da tastiera

Voce di menu	Scorciatoia
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

Menu Edit (fornito dal sistema, in Modalità esperto con selezione attiva del Project Navigator):

Azione	Scorciatoia
Undo Remove Image	⌘Z
Remove Selected Image	Backspace / Delete

## CAPITOLO

## Capitolo 2 — Inspector (Vista Esperto)

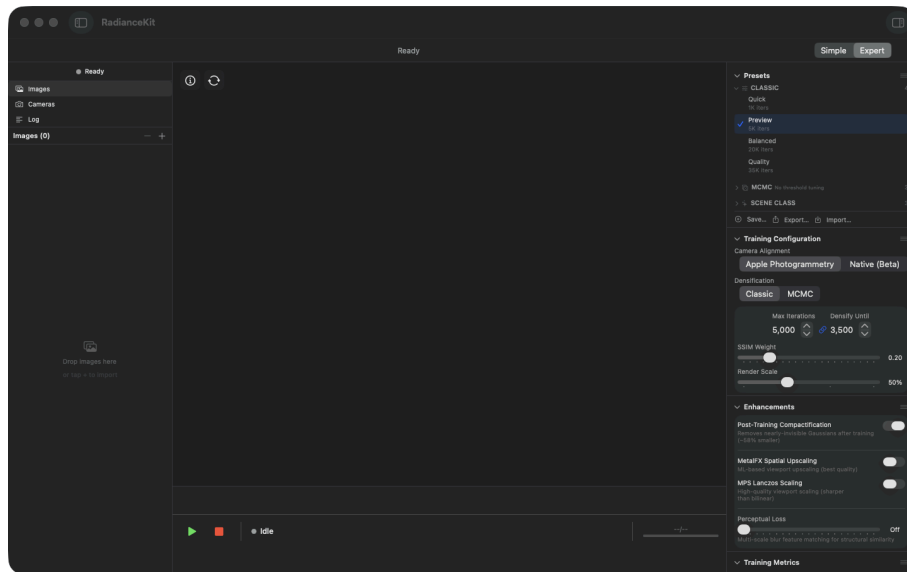


Figura 7: Modalità esperto vuota — Project Navigator a sinistra (Images 0, Cameras, Log), viewport vuoto al centro, Inspector a destra con sezioni Presets/Training Configuration/Enhancements/Training Metrics

**Inspector vuoto prima dell'importazione:** La sidebar a sinistra mostra il contatore Images 0 e il suggerimento «Drop images here / or tap + to import». L'Inspector a destra è completamente funzionale, ma i preset sono solo informativi (nessun training attivo). Il preset predefinito «Preview» (5K iter) è contrassegnato. Camera Alignment su Apple Photogrammetry, Densification Classic, SSIM Weight 0.20, Render Scale 50 %. Stati vuoti in Training Metrics («Start training to see live metrics») e Loss History («Loss curve will appear during training»).

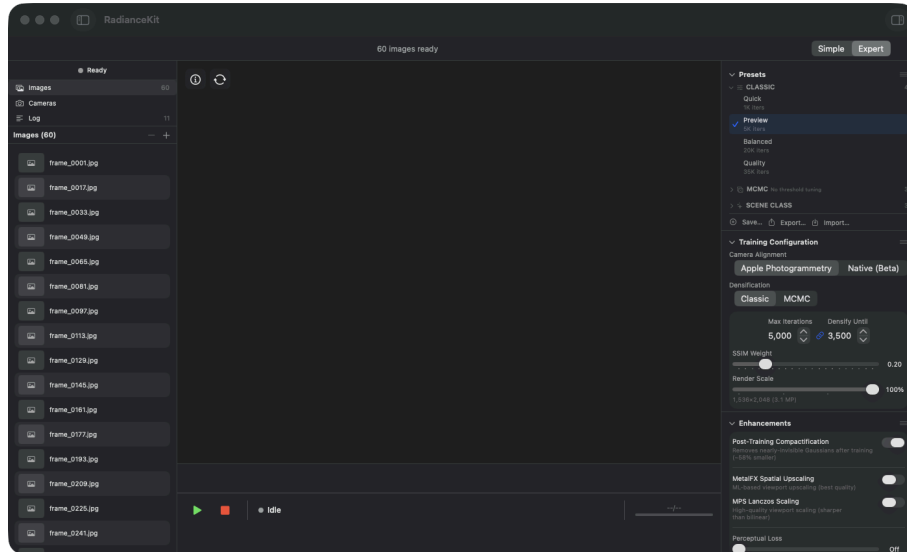


Figura 8: Inspector con 60 immagini di fiori caricate — la sidebar Image mostra i primi nomi di file frame\_0001.jpg ecc., header «60 images ready»

**Inspector dopo l'importazione:** Stato dell'header «60 images ready». La sidebar Image elenca tutti i 60 frame importati ( frame\_0001.jpg fino a frame\_0945.jpg , ogni 16° frame del dataset di 960 cam come sottoinsieme per iterazioni rapide). La logica auto-render-scale verifica la risoluzione dell'immagine ( $1536 \times 2048 = 3.1$  MP) e adatta Render Scale di conseguenza. Il pulsante play (verde, in basso a sinistra) è ora attivo e avvia il training con il preset attivo.

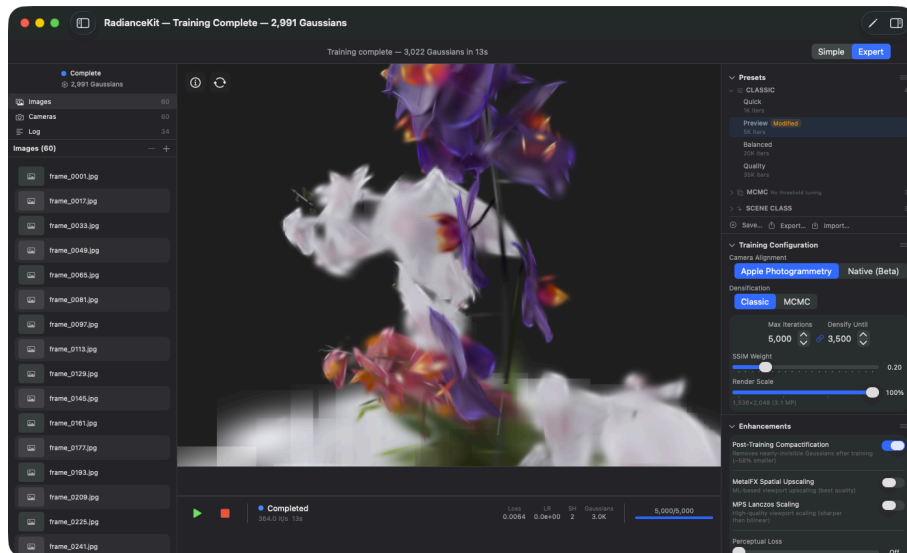


Figura 9: Inspector durante il training — il live viewport mostra la ricostruzione del bouquet di fiori, barra delle metriche in basso (Loss / LR / Gaussian Count / Iterations), scheda preset «Preview» con badge «Modified» se i parametri sono stati modificati

**Inspector durante il training:** La title bar mostra il progresso globale «RadianceKit — Training NN %». Il viewport renderizza la ricostruzione gaussiana in tempo reale (aggiornato ogni 50 iterazioni — intervallo live preview impostabile in Settings → General → Training → Live Preview). Barra delle metriche sotto il viewport: loss attuale, learning rate, gaussian count e contatore di iterazioni (ad es. 1,600/5,000 con preset

Preview). La scheda preset «Preview» dell'Inspector mostra il badge «Modified» non appena qualsiasi parametro si discosta dal default integrato. La sidebar «Log» raccoglie gli eventi degli stadi SfM e training.

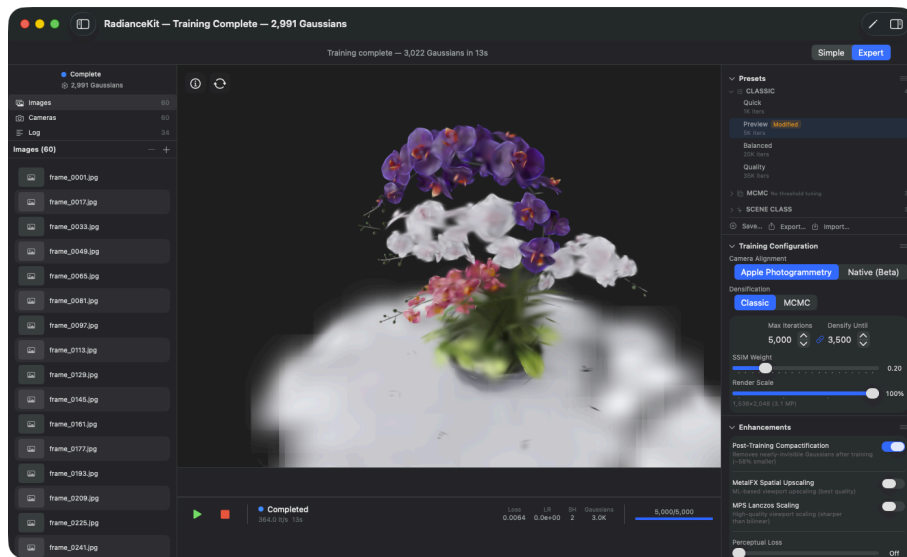


Figura 10: Inspector dopo il completamento del training — il viewport mostra la ricostruzione finita del bouquet di fiori (2.991 gaussiane dopo 5K iterazioni in 13s), title bar «Training Complete — 2,991 Gaussians»

**Inspector dopo il training:** La title bar mostra il numero finale di gaussiane (qui 2 991 — molto compatto, perché la scena sintetica del bouquet su sfondo chiaro ha geometria semplice). Il viewport mostra la nuvola di punti finita — navigazione drag orbitale attiva (ruota attorno al centro della scena). La sezione delle metriche di training è ora riempita con i valori finali, il grafico Loss History mostra l'andamento di tutte le 5 000 iterazioni. La sezione di esportazione in basso è ora attiva (tutti i pulsanti di formato abilitati).

L'Inspector è la sidebar di destra in Modalità esperto (⌘2). Raggruppa tutti i parametri rilevanti per il training in sette sezioni richiudibili. L'ordine predefinito dall'alto verso il basso al primo avvio è: Look, Preset, Configurazione del training, Metriche, grafico di perdita, Enhancements ed Export. La sezione «Look» (regolazioni dell'immagine post-training) è la reale ridenominazione UI della precedente sezione «Finishing» — il suo `rawValue` enum interno rimane «Finishing» per motivi di persistenza, l'intestazione mostrata si chiama «Look». Ogni sezione può essere richiusa cliccando sull'header, l'ordine può essere riorganizzato tramite drag-and-drop (`InspectorView.swift:81-97`).

**Al primo avvio tutte e sette le sezioni sono richiuse** (`inspectorCollapsedSections` di default è `Set(InspectorSection.allCases)`); lo stato dell'app salva poi le preferenze di richiusura e ordine attraverso gli avvii dell'app.

Una serie di controlli dell'Inspector compare in forma quasi identica anche nelle Impostazioni (Capitolo 3) — tipicamente backend SfM, Sky Masking e default simili. La separazione è intenzionale: le Impostazioni forniscono il template a livello di app per i progetti appena creati, l'Inspector sovrascrive questi valori per il progetto attualmente aperto. Chi conosce la logica di una pagina può usare l'altra ad occhi chiusi.

La colonna di sinistra in Modalità esperto — il Project Navigator — non appartiene all'Inspector, ma è il suo vicino diretto. Lì le immagini importate possono essere selezionate con un clic, viste in Quick Look con la barra spaziatrice ed eliminate tramite il

pulsante meno o il tasto Backspace (con Cmd-Z per l'annullamento). L'Inspector segue la selezione attuale della sidebar con informazioni di dettaglio specifiche del contesto, le sette sezioni principali rimangono però sempre disponibili.

## Sezione Look (L1–L5)

La sezione Look ( `rawValue` interno ancora «Finishing») è la sezione più in alto dell'Inspector e raccoglie in un unico posto le regolazioni dell'immagine **post-training**. Tutti i cursori lavorano in modo **non distruttivo**: ogni slider applica nuovamente il `FinishingPass` a uno snapshot pristino immutato (colore-DC, opacità e scala originali) — la regolazione è quindi **idempotente**, non cumulativa. Il risultato appare **dal vivo nel viewport** (WYSIWYG, esattamente come l'export successivo) e viene **incorporato in ogni export**. La sezione è disponibile solo **dopo il completamento di un training** (prima compare «Available after a training run completes.»); i suoi valori vengono **azzerati a ogni nuovo training**. Finché un export è in corso, tutti i cursori sono **bloccati** — compare un avviso di lock «Locked while exporting — the file uses the current settings.» e la GroupBox è disabilitata.

### L1 Slider Saturation



Inspector → sezione Look → GroupBox → Saturation.



Slider 0.5–1.2, visualizzazione a due decimali (ad es. «1.00»). Scala la croma SH-DC di ogni Splat attorno al valore di luminanza: 1.0 = invariato, < 1.0 = desaturato (colore tirato verso la scala di grigi), > 1.0 = più intenso. Matematicamente il colore-DC viene ricalcolato dallo snapshot pristino ( `desaturateDC` ), così che spostamenti ripetuti non si sommino. È stato validato su materiale di drone DJI (viadotto di Pensford), che tende a esagerare — il default per i droni è 0.82. Agisce solo sulla base di colore (SH grado 0), i coefficienti SH superiori restano intatti.

#### IN PAROLE SEMPLICI

Quanto sono intensi i colori dello Splat finito. 1.00 lascia tutto come addestrato, valori inferiori tirano il colore verso il grigio — utile per materiale di drone o video, che spesso esce sovrassaturato. Valori sopra 1.0 lo rendono più intenso. Puoi spostare avanti e indietro quanto vuoi senza che nulla si «accumuli», perché l'app ricalcola sempre dallo stato originale immutato. Visibile dal vivo nel viewport ed esattamente così nell'export.

## L2 Slider Splat length



DOVE

Inspector → sezione Look → GroupBox → Splat length.



TECNICO

Slider 0.3–1.0, visualizzazione a due decimali. Tira i tre assi di scala di ogni gaussiana nello spazio logaritmico verso il loro valore medio ( `shortenScale` , fattore `alpha` ): 1.0 = invariato, valori più piccoli rendono più rotondi gli Splat allungati ad «ago», 0 sarebbero sfere pure. Interviene sugli Splat aghiformi e sovraestesi senza modificare la dimensione complessiva, riducendo così i tipici artefatti «confetti». Applicato a partire dallo snapshot pristine (scala logaritmica originale), quindi idempotente. Commuta con Splat size (L3), perché entrambi operano nello spazio logaritmico.

### IN PAROLE SEMPLICI

Rende più rotondi gli Splat troppo lunghi e schegge. 1.00 lascia la forma come addestrata, valori inferiori comprimono gli «aghi» allungati in macchie più tondeggianti — ciò calma le ricostruzioni granulose, afflitte da artefatti confetti. La dimensione complessiva resta uguale, si tratta solo dell'allungamento. Si può combinare senza problemi con Splat size (L3).

## L3 Slider Splat size



DOVE

Inspector → sezione Look → GroupBox → Splat size.



TECNICO

Slider 0.5–2.0, visualizzazione a due decimali. Scala ogni gaussiana in modo uniforme su **tutti** e tre gli assi ( `sizeScale` ): 1.0 = invariato, < 1.0 = più piccola/più densa/più nitida, > 1.0 = più grande/più «soffice» (riempie i vuoti tra gli Splat). Poiché le scale sono nello spazio logaritmico, la moltiplicazione viene realizzata come offset additivo `log(factor)` — ciò commuta con Splat length (L2), perché un offset costante lascia inalterato lo scostamento dalla media. A partire dallo snapshot pristine, quindi idempotente. Nuovo in questa versione.

### IN PAROLE SEMPLICI

Scala tutti gli Splat in modo uniforme, più grandi o più piccoli. 1.00 è lo stato addestrato, valori inferiori rendono la nuvola di punti più stretta e nitida, valori superiori coprono i vuoti tra gli Splat (risultato più morbido/«soffice»). Pratico per chiudere otticamente una ricostruzione bucherellata o, viceversa, per far emergere più dettaglio. Si combina senza problemi con Splat length (L2) — i due cursori non si influenzano a vicenda.

#### L4 Fade far region (con sub-slider)



DOVE

Inspector → sezione Look → GroupBox → Toggle «Fade far region» più i sub-slider «Fade start xradius» e «Fade floor».



TECNICO

Toggle che attiva una caduta radiale dell'opacità con la distanza dal baricentro delle camere — i «far-confetti» debolmente osservati sullo sfondo vengono dissolti. **Solo per riprese in orbit**: il toggle è disabilitato quando `finishingContext.fadeEligible` è false (voli lineari, troppe poche camere o camere degeneri); allora al posto dei sub-slider compare l'avviso «Far-fade applies only to orbit captures (not this scene).» L'idoneità viene determinata tramite la copertura azimutale delle posizioni delle camere (un orbit gira attorno al baricentro e riempie molti settori della bussola, un volo lineare solo ~2). Due sub-slider controllano la geometria: **Fade start xradius** (1.0–3.0) imposta il raggio interno come multiplo del raggio dell'orbit, entro il quale vale piena opacità; **Fade floor** (0.0–1.0) è il fattore di opacità ben oltre il raggio di fade. Importante: il fade **salta la regione sky-dome** (le gaussiane congelate degli indici [0, frozenCount]), così che la cupola di sfondo intenzionale non venga attenuata.

#### IN PAROLE SEMPLICI

Dissolve i resti sfocati al bordo esterno della scena — proprio quei grumi «far-confetti» che fluttuano lontano sullo sfondo nelle riprese a giro completo. Funziona solo con vere riprese orbit/a giro; con voli di drone rettilinei o troppe poche camere l'interruttore è disattivato e un avviso ne spiega il motivo. Quando è attivo, si aggiungono due regolatori fini: «Fade start xradius» stabilisce a quale distanza (come multiplo del raggio di giro) inizia la dissolvenza, «Fade floor» quanto restano ancora visibili alla fine gli Splat lontani (0 = del tutto via, 1 = invariati). Una sky-dome ricostruita intenzionalmente (I44) non viene mai toccata — il cielo resta intatto.

## L5 Pulsante Reset finishing



Inspector → sezione Look → GroupBox → «Reset finishing» (in basso, pulsante piccolo).



Riporta tutte le impostazioni Look ai default (`FinishingPass.Settings() = Saturation 1.0, Fade off, Splat length 1.0, Splat size 1.0`) e innesca subito un nuovo finishing, così che il viewport torni allo stato addestrato immutato. `controlSize(.small)`. Poiché l'intero stack Look ricalcola in modo idempotente a partire dallo snapshot pristino, «torna al default» è esattamente l'output di training originale — nessuna perdita di qualità per i ripetuti avanti e indietro. Come tutti i cursori della sezione, è bloccato durante un export in corso.

### IN PAROLE SEMPLICI

Riporta con un clic tutti i cursori Look allo standard (Saturation 1.00, Fade off, entrambi gli slider Splat a 1.00) — il viewport mostra poi di nuovo esattamente il risultato appena addestrato. Pratico quando hai pasticciato e vuoi ricominciare da zero in modo pulito. Poiché l'app ricalcola sempre dallo stato originale, non c'è alcuna perdita di qualità. Mentre un export è in corso, il pulsante (come gli slider) è bloccato.

## Sezione Preset (I1–I11)

La sezione Preset è il modo più rapido per applicare una configurazione testata. I preset integrati (Capture Class, Classic, MCMC, Hybrid) forniscono punti di partenza riproducibili da 560+ esperimenti documentati; i preset propri possono essere salvati, esportati, importati e condivisi. La lista è raggruppata per categorie (Capture Class, Classic, MCMC, Hybrid, Custom) e più di una categoria può essere espansa contemporaneamente. Tramite il meccanismo del menu contestuale (clic destro su una riga) sono raggiungibili Export, Duplicate e — per preset propri — Delete.

**I1 Pulsante Save...**

Inspector → sezione Preset → pulsante Save... (barra azioni in basso).



Apri un popover con campo di testo e pulsanti Save/Cancel. Lo stato attuale di TrainingConfig viene persistito come nuovo preset personalizzato (codificato JSON, salvato a livello di app). L'operazione di salvataggio copia tutti gli 81 parametri di training più la strategia di densification attuale. Il preset finisce automaticamente nella categoria Custom, indipendentemente da quale preset integrato sia stato derivato. Nomi vuoti e input di solo whitespace vengono scartati. I nomi già esistenti non vengono rifiutati — ogni preset ha un proprio ID interno, i nomi duplicati sono tecnicamente consentiti ma in pratica confusi.

**IN PAROLE SEMPLICI**

Salva la tua configurazione attuale come preset riutilizzabile. Premi il pulsante, inserisci un nome nel popover e clicca Save — tutti gli 81 parametri inclusa la strategia di densification finiscono sotto il nome scelto nella categoria Custom. Ti serve quando ti sei dato da fare e non vuoi smanettare da capo al prossimo progetto. Particolarmente pratico per setup ricorrenti come «Drone 4K» o «Indoor rapido». I nomi duplicati sono tecnicamente consentiti ma in pratica confusi — prendi piuttosto qualcosa di parlante.

**I2 Campo di testo Preset Name**

Popover Save → campo di testo «Preset Name».



Semplice campo di testo con bordo arrotondato, forma larga. Il valore viene assunto come nome del preset al clic sul pulsante Save. Nessun limite di lunghezza nell'UI, ma il nome salvato deve essere codificabile in JSON e rappresentabile nelle liste UI — emoji e caratteri accentati funzionano. Il contenuto viene automaticamente resettato a stringa vuota all'apertura del popover. Il pulsante Save rimane disabilitato finché il campo dopo trim è vuoto. Non c'è auto-suggest né preimpostazione con il nome del preset attualmente attivo.

**IN PAROLE SEMPLICI**

Qui digiti il nome per il tuo preset. Scegli qualcosa di parlante come «Drone 4K 30fps» o «Interno rapido» — ti aiuta in seguito a ritrovarlo nella categoria Custom. Emoji e caratteri accentati sono consentiti, un limite rigido di lunghezza non c'è. Finché il campo è vuoto o consiste solo di spazi, il pulsante Save rimane in grigio. Alla nuova apertura del popover il campo è di nuovo vuoto — non c'è preimpostazione con il nome del preset attivo.

### I3 Pulsante Cancel (dialogo Save)



Popover Save → pulsante Cancel (a sinistra).



Chiude il popover senza salvare. Scarta il contenuto del campo di testo — alla prossima apertura viene di nuovo resettato a vuoto dalla logica del pulsante Save... (I1). Stile di pulsante standard, nessuna finestra di conferma, nessuna hotkey. La TrainingConfig attuale rimane invariata, poiché il percorso di salvataggio non è stato eseguito.

#### IN PAROLE SEMPLICI

Chiude il popover Save senza salvare nulla. Se hai cambiato idea, hai sbagliato a digitare o hai aperto il dialogo per errore — basta cliccare Cancel. La tua configurazione di training attuale rimane invariata, perché non è stato ancora scritto nulla. Alla prossima apertura del popover il campo nome parte di nuovo vuoto. Nessuna domanda di sicurezza, nessuna hotkey — semplice clic e via.

### I4 Pulsante Save (dialogo Save)



Popover Save → pulsante Save (a destra, stile prominente).



Attiva la persistenza effettiva. Valida nuovamente il nome non vuoto (controllo difensivo) e scrivi poi la TrainingConfig attuale come JSON nella memoria dell'app. Successivamente chiude il popover. Evidenziato in blu, in grigio finché il campo di testo è vuoto. Se il salvataggio fallisce (ad es. perché la memoria dell'app è piena — molto improbabile), al momento non c'è una finestra di errore visibile; il preset semplicemente non apparirebbe al prossimo avvio dell'app.

#### IN PAROLE SEMPLICI

Con un clic su Save assumi il nome e scrivi il tuo setup attuale come nuovo preset. Il popover si chiude, il preset appare immediatamente nella categoria Custom della lista preset e da ora può essere attivato con un clic. Il pulsante è evidenziato in blu ( `borderedProminent` ) e rimane in grigio finché il campo nome è vuoto. Se il salvataggio fallisce (ad es. UserDefaults pieno), non c'è una finestra di errore visibile — il preset semplicemente mancherebbe al prossimo avvio dell'app.

**I5 Pulsante Export...**

DOVE

Inspector → sezione Preset → barra azioni → pulsante Export....



TECNICO

Esporta il preset attualmente selezionato come file `.radiancepreset` (internamente JSON). Disabilitato se non è selezionato alcun preset. Al clic l'app apre una finestra di salvataggio con nome file predefinito (nome preset + estensione `.radiancepreset`). Il formato salvato contiene l'intera TrainingConfig più metadati (nome, categoria, ID, flag built-in). Il doppio clic nel Finder apre l'app — ma **non** automaticamente l'importazione; l'utente deve usare il pulsante Import (I6).

**IN PAROLE SEMPLICI**

Seleziona un preset nella lista e clicca Export — così puoi salvarlo come file `.radiancepreset` e ad es. inviarlo a un collega o trasferirlo su un secondo Mac. Il destinatario lo ricarica dall'altra parte con il pulsante Import... (I6). Funziona allo stesso modo per built-in e per i tuoi preset Custom propri. Il pulsante è in grigio finché nella lista non è cliccato nulla. Suggerimento: tramite il menu contestuale (I8) va ancora più veloce — lì non devi prima selezionare il preset.

**I6 Pulsante Import...**

DOVE

Inspector → sezione Preset → barra azioni → pulsante Import....



TECNICO

Apri una finestra di dialogo dei file che consente solo file `.radiancepreset` (selezione multipla disabilitata). Alla selezione il file JSON viene caricato, validato e inserito nella categoria Custom — con un nuovo ID interno per evitare collisioni con i built-in. L'importazione imposta automaticamente la categoria su Custom, anche se il preset esportato era originariamente ad es. un built-in. File danneggiati o incompatibili con uno schema di versione precedente vengono silenziosamente rifiutati, senza finestra di errore (il log della console fornisce però informazioni).

**IN PAROLE SEMPLICI**

Carica un file `.radiancepreset` da disco. Utile se qualcuno ti invia un setup collaudato o se vuoi mantenere sincronizzati i tuoi preset preferiti su più Mac. I preset importati finiscono sempre nella categoria Custom — anche se sono stati originariamente esportati dai built-in. File danneggiati o obsoleti vengono ignorati silenziosamente; nel log della console c'è poi il motivo. La selezione multipla nella finestra è disabilitata, quindi un solo file per clic.

## I7 Riga preset (attivazione con clic)



Inspector → sezione Preset → ogni riga preset in ogni categoria.



Il clic su una riga preset sostituisce tutti i campi della TrainingConfig con i valori dal preset, memorizza l'ID del preset attivo e resetta lo stato modificato. Il segno di spunta attivo davanti alla riga appare solo se il preset è selezionato e non modificato. Non appena un valore viene modificato nella TrainingConfig (slider, stepper, toggle nelle altre sezioni dell'Inspector), appare un badge arancione «Modified» dopo il nome. I preset integrati non possono essere sovrascritti — in caso di modifica deve essere creata una copia propria tramite il pulsante Save (I1).

### IN PAROLE SEMPLICI

Il clic su una riga attiva il preset e assume tutti i valori lì memorizzati nelle impostazioni di training attuali. Il segno di spunta davanti al nome mostra quale preset è attualmente attivo. Non appena dopo regoli un qualsiasi slider, stepper o toggle nelle altre sezioni, dopo il nome appare un badge arancione «Modified» — perché il tuo setup ora si discosta dal preset. I preset integrati non possono essere sovrascritti; se vuoi mantenere le modifiche, crea una copia propria tramite il pulsante Save... (I1) o duplica il preset (I9).

## I8 Menu contestuale «Export...»



Clic destro su ogni riga preset → prima voce «Export...».



Funzionalità identica a I5 (pulsante Export...), ma raggiungibile più comodamente — senza che il preset debba essere prima selezionato. Esporta direttamente il preset cliccato nella riga. Funziona allo stesso modo per tutte le categorie di preset (built-in o Custom), nessuna restrizione. L'esportazione contiene il flag built-in e la categoria originale, ma al re-import la categoria viene mappata su Custom come descritto in I6.

### IN PAROLE SEMPLICI

Modo rapido per esportare — clic destro sul preset desiderato e selezionare «Export...». Risparmia la deviazione tramite selezione preliminare e poi pressione del pulsante Export... Funziona allo stesso modo per tutte le categorie, anche per i built-in. Il file `.radiancepreset` generato è identico a quello di I5; al successivo re-import finisce automaticamente nella categoria Custom.

## I9 Menu contestuale «Duplicate»



Clic destro su ogni riga preset → seconda voce «Duplicate».



Clona il preset nella categoria Custom. Genera un nuovo ID interno, appende « Copy» al nome e salva la copia. Funziona anche per i preset integrati — il clone è poi editabile. L'originale rimane intatto. La TrainingConfig viene copiata valore per valore (roundtrip JSON), in modo che non esistano collegamenti di riferimento tra originale e copia.

### IN PAROLE SEMPLICI

Genera una copia modificabile di un preset nella categoria Custom. Pratico se vuoi ad es. il preset integrato «Quality» come punto di partenza e poi vuoi solo spostare un po' lo slider SSIM. Flusso di lavoro: duplica, rinomina (menu contestuale o nuovo Save...), adatta, fatto. L'originale rimane intatto — puoi tornarci in qualsiasi momento. Funziona anche per i built-in, che è l'unico modo per assumere i loro valori come base e contemporaneamente renderli modificabili.

## I10 Menu contestuale «Delete»



Clic destro sulle righe preset proprie → ultima voce «Delete» (rosso, distruttivo).



Visibile solo per preset Custom. I built-in non possono essere eliminati. La voce è contrassegnata come distruttiva, appare in rosso nel menu contestuale e viene separata da un divisore in modo da non cliccarla per errore. **Non** c'è alcun dialogo di conferma — un clic elimina immediatamente il preset. Il preset eliminato non è ripristinabile (Cmd-Z non funziona qui — l'annullamento esiste nella build attuale solo per la lista delle immagini, non per le operazioni sui preset). Se il preset eliminato era attualmente attivo, la TrainingConfig attuale rimane invariata, solo la selezione preset attiva viene azzerata.

### IN PAROLE SEMPLICI

Eliminare i preset propri. Sui built-in (Quick, Preview, Balanced, Quality, Ultra Detail, Drone / Aerial, 360° Walkaround, Photo / Object ecc.) la voce «Delete» non è proprio visibile — non puoi eliminarli per errore. Attenzione: non c'è domanda di sicurezza né annullamento, un clic e il preset è andato. Se non sei sicuro, fai prima una copia di sicurezza su disco tramite Export... (I5/I8) — puoi reimportarla in qualsiasi momento. Se il preset era attualmente attivo, la tua TrainingConfig rimane invariata, scompare solo il segno di spunta.

## I11 Header di categoria (espandi/comprimi)



DOVE

Inspector → sezione Preset → ogni header di categoria (Capture Class, Classic, MCMC, Hybrid, Custom).



TECNICO

Stato di richiusura per categoria con default diverso: il gruppo curato Capture Class parte **espanso**, Classic, MCMC, Hybrid e Custom partono **chiusi**. Lo stato non viene persistito — al riavvio dell'app tutte le categorie sono di nuovo nello stato predefinito. La freccia chevron ruota animata. Il numero a destra nell'header mostra il numero di preset in questa categoria. L'area di hit del clic copre l'intera area dell'header.

### IN PAROLE SEMPLICI

Espandere e comprimere le categorie per mantenere ordinata la lista preset. All'avvio dell'app il gruppo Capture Class è aperto, Classic, MCMC, Hybrid e Custom sono chiusi. Clic sull'header (tutta l'area è cliccabile) e la lista si apre o si chiude con breve animazione del chevron. Il piccolo numero a destra mostra quanti preset ci sono nella categoria. Dopo il riavvio dell'app c'è di nuovo lo stato predefinito — l'app deliberatamente non salva questa impostazione di richiusura.

## Sezione Configurazione del Training (I12–I22)

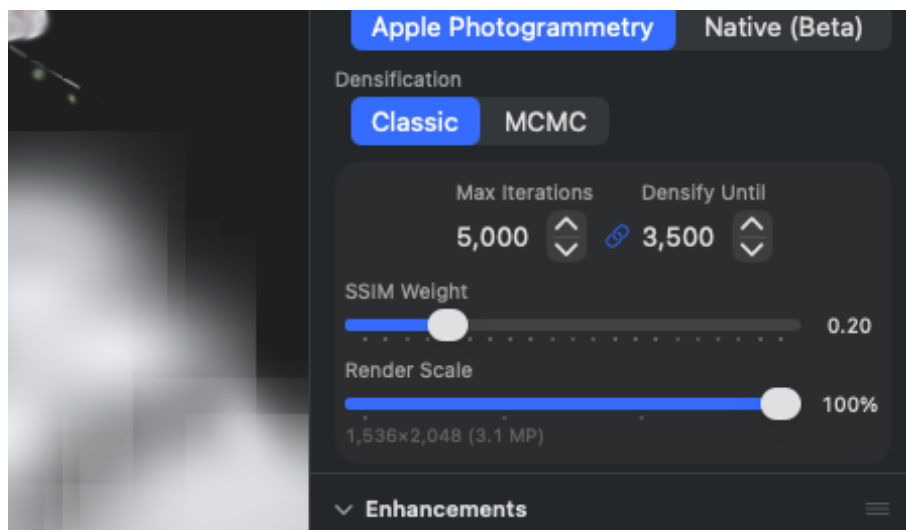


Figura 11: Crop solo della sezione Configurazione del training — Camera Alignment (Apple Photogrammetry attivo, Native (Beta) inattivo), Densification (Classic attivo), Max Iterations 5,000 / Densify Until 3,500 con simbolo di link, slider SSIM Weight 0.20, slider Render Scale al 100 % (1,536×2,048 = 3.1 MP)

Qui si trovano le leve centrali: quale backend SfM deve calcolare, come lavora la densification, quante iterazioni, quanto grande è il peso SSIM. Con la strategia MCMC compaiono due toggle aggiuntivi («MCMC Quality» e «Auto-scale by scene») che vengono nascosti in modalità Classic. Con il backend Native SfM si aggiunge il campo FOV Override, che serve solo per frame video senza lunghezza focale EXIF.

## I12 Picker Camera Alignment



DOVE

Inspector → Configurazione del training → Camera Alignment (picker segmentato in alto).



TECNICO

Picker segmentato con due opzioni: Apple Photogrammetry e Native (Beta). La selezione determina il backend SfM utilizzato alla prossima ricostruzione delle fotocamere. Influenza contemporaneamente quali altri elementi dell'Inspector sono visibili: Native mostra in più il FOV Override (I13), che serve solo con frame video privi di EXIF. Nota: per riprese outdoor molto grandi puoi caricare il risultato di uno strumento esterno (Metashape o COLMAP) tramite Workspace Import — vedi Capitolo 1 (M5) e Capitolo 9 (Q3, Q6).

### IN PAROLE SEMPLICI

Qui scegli come vengono ricostruite le posizioni delle fotocamere — l'interruttore più importante per la qualità finale. Apple Photogrammetry è lo standard veloce e basta completamente per la maggior parte delle scansioni di oggetti. Native (Beta) è lo sviluppo proprio conforme all'App Store, buono per orbite e scene turntable, e con frame video privi di EXIF ha bisogno del FOV Override (I13). Con set outdoor molto grandi puoi in alternativa calcolare le fotocamere in Metashape o COLMAP e caricare il risultato tramite l'importazione Workspace. Dettagli e raccomandazioni per tipo di scena li trovi nel Capitolo 9.

## I13 Campo FOV Override (SfM Native)



DOVE

Inspector → Configurazione del training → FOV Override (visibile solo con Camera Alignment = Native).



TECNICO

Campo di testo numerico (range 0-170°), default 0 = determinazione automatica da EXIF o euristica. L'input manuale è necessario se le immagini di input sono state estratte da un video che non contiene metadati di lunghezza focale. Valori tipici: iPhone Wide  $\approx 73^\circ$ , DJI Mavic Wide-Crop  $\approx 70^\circ$ , drone con sensore full-frame  $\approx 84^\circ$ . Il valore viene limitato a [0, 170] — valori fuori vengono riportati direttamente al limite. Ha effetto solo sulla pipeline SfM nativa (Q4/Q5); Apple Photogrammetry ignora completamente questo valore.

### IN PAROLE SEMPLICI

Se le tue immagini non hanno EXIF (tipico con frame estratti da video), inserisci qui l'apertura orizzontale della fotocamera in gradi. Valori indicativi: iPhone Wide  $\approx 73^\circ$ , DJI Mavic Wide-Crop  $\approx 70^\circ$ , drone con sensore full-frame  $\approx 84^\circ$ . Uno 0 lascia che l'app indovini da sola — spesso va bene, ma può andare storto con obiettivi rari. Valori sopra  $170^\circ$  vengono automaticamente riportati al limite. Il campo è visibile e ha effetto solo se hai scelto Native come Camera Alignment (I12) — Apple Photogrammetry lo ignora completamente.

## I15 Picker Densification



DOVE

Inspector → Configurazione del training → Densification (picker segmentato, sempre visibile).



TECNICO

Commuta tra le due strategie di densification: Classic (metodo 3DGS originale con clone/split/prune e soglia di gradiente) e MCMC (Stochastic Gradient Langevin Dynamics con rilocalizzazione, NeurIPS 2024). Al passaggio da Classic a MCMC l'app imposta automaticamente i campi specifici MCMC su valori di default collaudati (reg-weight = 0, MCMC cap-multiplier 3.0, sample/noise schedule). Senza questa inizializzazione automatica le sessioni con preset vecchi soffrivano del bug MCMC collapse 1.4.4 (460K→5 gaussiane, kill del watchdog). La selezione del picker determina inoltre quali elementi dell'Inspector sono visibili — con MCMC compaiono I16/I17. Effetto dettagliato dei campi nel Capitolo 6, T11–T16 (Classic) e T61–T73 (MCMC).

### IN PAROLE SEMPLICI

La scelta di strategia centrale per la crescita del numero di gaussiane. Classic è ben sintonizzato da 459 esperimenti, genera risultati rapidi e di alta qualità e non ha bisogno di conoscere i campi MCMC. MCMC è l'approccio più recente (NeurIPS 2024), più riproducibile e rinuncia alla regolazione manuale della soglia — in cambio impiega circa 6x più tempo a qualità paragonabile. Al passaggio a MCMC l'app imposta automaticamente default sicuri, in modo che il training non vada in collapse 1.4.4. I dettagli sui campi di strategia sono nel Capitolo 6 (T11–T16 Classic, T61–T73 MCMC).

## I16 Toggle MCMC Quality



DOVE

Inspector → Configurazione del training → MCMC Quality (solo con Densification = MCMC).



TECNICO

Commuta l'accumulo del gradiente su 2 passi (attivo) o 1 passo (inattivo). Accumula i gradienti da due viste di fotocamera consecutive prima dell'esecuzione dello step dell'optimizer. Empiricamente (sessione 33, V544a) ciò riduce l'errore L1 finale di circa il 6% (0.0246 con Quality vs 0.0261 senza, in media su 3 trial su Horse-Full-MCMC). Il prezzo: tempo di training raddoppiato. Su training molto lunghi (200K iterazioni) ciò porta a oltre 10 minuti di attesa aggiuntiva — quindi vale la pena solo se gli ultimi punti percentuali di qualità sono davvero necessari. Ha effetto solo sul training, non sul formato di esportazione o sulla visualizzazione del viewport.

### IN PAROLE SEMPLICI

Modalità Quality per MCMC con accumulo del gradiente su due viste. Rende empiricamente il risultato finale circa il 6% migliore (L1 0.0246 invece di 0.0261 nel test Horse), ma costa il doppio del tempo. Se stai già eseguendo un training MCMC 200K (volentieri 2 ore), si aggiunge un'altra ora circa. Vale la pena per rendering showcase finali o alla fine di una sessione di quality sweep, nel flusso di lavoro quotidiano piuttosto no. Visibile solo se Densification è su MCMC (I15).

## I17 Toggle Auto-scale by scene



DOVE

Inspector → Configurazione del training → Auto-scale by scene (solo con MCMC).



TECNICO

Se attivo, scala il limite superiore effettivo di Max-Gaussians con SfM-init-point-count × MCMC-cap-multiplier (default 3.0). Esempio: SfM fornisce 250K initpoint, base cap = 150K, multiplier 3.0 → limite superiore effettivo =  $\max(150K, 750K) = 750K$ . Se disattivato, vale rigorosamente solo la base. Introdotto per v1.4.5, perché grandi riprese outdoor con oltre 1000 frame e corrispondente alta densità di punti SfM facevano morire di fame la densificazione con il default rigido di 150K cap — i punti superflui rimanevano, quelli nuovi non potevano nascere. Default OFF nei preset Custom, ON nei built-in MCMC. Ha effetto solo durante il training, non nell'esportazione.

### IN PAROLE SEMPLICI

Fa crescere il numero massimo di gaussiane con la dimensione della scena (più precisamente: con il numero di initpoint SfM). Per scene piccole noti appena una differenza, per grandi scene outdoor è spesso decisiva per la qualità — altrimenti il training «soffoca», perché il limite di default di 150K è troppo basso per la scena. Introdotto specificamente per v1.4.5 dopo che set outdoor molto grandi (oltre 1000 frame) erano visibilmente appesi al cap. Nei preset built-in MCMC è già acceso in anticipo; nei preset propri di default spento.

## I18 Stepper Max Iterations



DOVE

Inspector → Configurazione del training → Group-Box → Max Iterations.



TECNICO

Stepper con range 1 000–100 000, passo 1 000. Determina il numero totale di iterazioni dell'optimizer. Correlato linearmente al tempo di training (dimezzamento = circa 50% del tempo). Sweet spot empirici: 20K (Classic Balanced,  $L1 \approx 0.028$ ), 40K (Classic Quality,  $L1 \approx 0.023$ ), 200K (MCMC Full,  $L1 \approx 0.0246$ ). Oltre 40K con Classic in media non porta quasi alcun miglioramento — diminishing returns. Alla modifica, se la funzione di link (I19) è attiva, Densify Until viene tirato proporzionalmente (rapporto di default: 0.5, ossia Densify Until =  $\text{Max}/2$ ).

### IN PAROLE SEMPLICI

Quanti passi di training vengono eseguiti — più è meglio, ma costa anche linearmente più tempo. Regola pratica: 20 000 per buona qualità, 40 000 per l'ottimo con strategia Classic (oltre non porta quasi nulla in media). MCMC ha bisogno decisamente di più, qui 200 000 è standard. Raddoppiare le iterazioni raddoppia grossolanamente il tempo di training. Con pulsante link attivo (I19) Densify Until viene tirato proporzionalmente — praticamente sempre quello che vuoi.

## I19 Pulsante Link/Unlink (Densify ↔ Iterations)



Inspector → Configurazione del training → Group-Box → piccolo pulsante Link tra Max Iterations e Densify Until.



Pulsante toggle che congela il rapporto tra Densify Until e Max Iterations. Con attivo (icona link evidenziata) a ogni modifica di Max Iterations Densify Until viene tirato proporzionalmente. Con unlink (icona link-plus) i valori rimangono indipendenti. Default è linked, perché ciò riflette la correlazione tipica — se tiri il training su iterazioni doppie, di solito vuoi anche far girare la densification proporzionalmente più a lungo. Il rapporto viene calcolato all'impostazione del pulsante link dal valore attuale; un rapporto tipico è 0.5 (Densify Until = metà del numero di iter).

### IN PAROLE SEMPLICI

Piccolo pulsante a forma di graf-fa tra Max Iterations e Densify Until. Quando attivo (icona link evidenziata), i due valori si muovono insieme — se raddoppi le iterazioni, Densify Until raddoppia nello stesso rapporto. Quando no (icona `link.badge.plus`), puoi impostarli in modo indipendente. Lo standard è linkato, perché ciò riflette la correlazione tipica — un training più lungo di solito vuole anche una fase di densification più lunga. Per il 99% dei casi lascialo agganciato.

## I20 Stepper Densify Until



Inspector → Configurazione del training → Group-Box → Densify Until.



Stepper con range 500–50 000, passo 500. Determina l'indice di iterazione a partire dal quale non vengono più aggiunte nuove gaussiane tramite clone/split (Classic) o rilocazione (MCMC). Al raggiungimento vengono raffinate solo posizione e colore. Valori più alti = più gaussiane = file più grande, tempo per iterazione più lungo (+30–60% di tempo GPU per passo). Valori tipici: 15K (per 30K Max Iter), 20K (per 40K), 100K (per 200K MCMC). Con link attivo (I19) automaticamente scalato. Ha effetto diverso con Classic vs MCMC: Classic ferma completamente la crescita, MCMC ferma la logica di rilocazione, ma l'adattamento di sample/noise prosegue.

### IN PAROLE SEMPLICI

Fino a quale iterazione possono essere aggiunte nuove gaussiane — con Classic tramite clone/split, con MCMC tramite rilocazione. Dopo si tratta solo del affinamento di colore e forma dei punti esistenti. Più alto = più dettaglio, ma anche file più grande e +30–60% di tempo GPU per passo. Valori tipici: 15K (per 30K Max Iter), 20K (per 40K), 100K (per 200K MCMC). Di solito appeso tramite link (I19) a Max Iterations — raramente sensato disaccoppiarlo manualmente.

**I21 Slider SSIM Weight**

Inspector → Configurazione del training → Group-Box → SSIM Weight.



Slider 0.0–1.0 in passi di 0.05, visualizzazione come «0.20». Mescola loss L1 (0.0) e loss SSIM (1.0). L1 stringe la luminosità per pixel, SSIM la somiglianza strutturale (bordi, statistiche locali). Il default 0.2 è il valore del paper 3DGS originale (Kerbl 2023) e reverse-engineered come compromesso robusto in numerose sessioni. Valori più alti (0.5+) preferiscono la conservazione del dettaglio, ma possono ignorare errori locali di luminosità. Valori più bassi (< 0.1) portano alla perdita di dettaglio sui bordi netti. Il calcolo SSIM avviene nello shader con una finestra Gaussian 11×11. Prestazioni: a 0.0 (solo L1) il training è circa 8-12% più veloce, perché il calcolo SSIM viene saltato nello shader.

 IN PAROLE SEMPLICI

Quanto fortemente viene pesata la somiglianza strutturale dell'immagine (bordi, pattern locali) rispetto al puro confronto di luminosità. 0.2 è lo standard dal paper 3DGS originale e basta per quasi tutte le scene. Più alto (0.5+) per strutture fini come capelli, pelliccia o vegetazione — lì aiuta più peso strutturale. Più basso (0.0) rende il training circa 8-12% più veloce, perché il calcolo SSIM viene saltato nello shader, ma costa dettaglio sui bordi netti. Chi non ha un buon motivo per un cambiamento, lascia 0.2.

**I22 Slider Render Scale**

Inspector → Configurazione del training → Group-Box → Render Scale.



Slider 0.25–1.0 in passi di 0.25, visualizzazione come «100%». Scala la risoluzione di rendering di training relativa alla dimensione dell'immagine sorgente. La più grande leva sulle prestazioni: 50% riduce il tempo GPU di circa il 75% (perché 4× meno pixel), 25% di circa il 94%. La soglia del gradiente viene scalata automaticamente. Sotto lo slider appare una visualizzazione live della risoluzione in MP (ad es. «2304×1296 (3.0 MP)»). Se il valore attuale si discosta da quello raccomandato, viene mostrato in arancione «— recommended: 50%». La raccomandazione mira a ~3 MP di risoluzione effettiva — l'intervallo elaborato più efficientemente dalle GPU Apple Silicon. Le immagini sorgente 4K ricevono ad es. automaticamente 25% raccomandato, le immagini FullHD 100%. Una modifica attiva inoltre la riallocazione del buffer.

 IN PAROLE SEMPLICI

Con quale risoluzione il training renderizza — una delle più grandi leve di prestazione. Pieno (100%) dà la migliore qualità, ma costa molto tempo GPU con immagini grandi. Metà (50%) risparmia circa il 75% di tempo GPU, perché vengono calcolati quattro volte meno pixel — perfetto per sorgenti 4K. Sotto lo slider vedi la risoluzione effettiva in megapixel; l'app mira a circa 3 MP, perché è ciò che funziona più efficientemente su Apple Silicon. Se il tuo valore si discosta da ciò, l'app mostra una nota arancione «recommended» — di solito vale la pena seguirla.

## Sezione Enhancements (I26–I29, I42–I44)

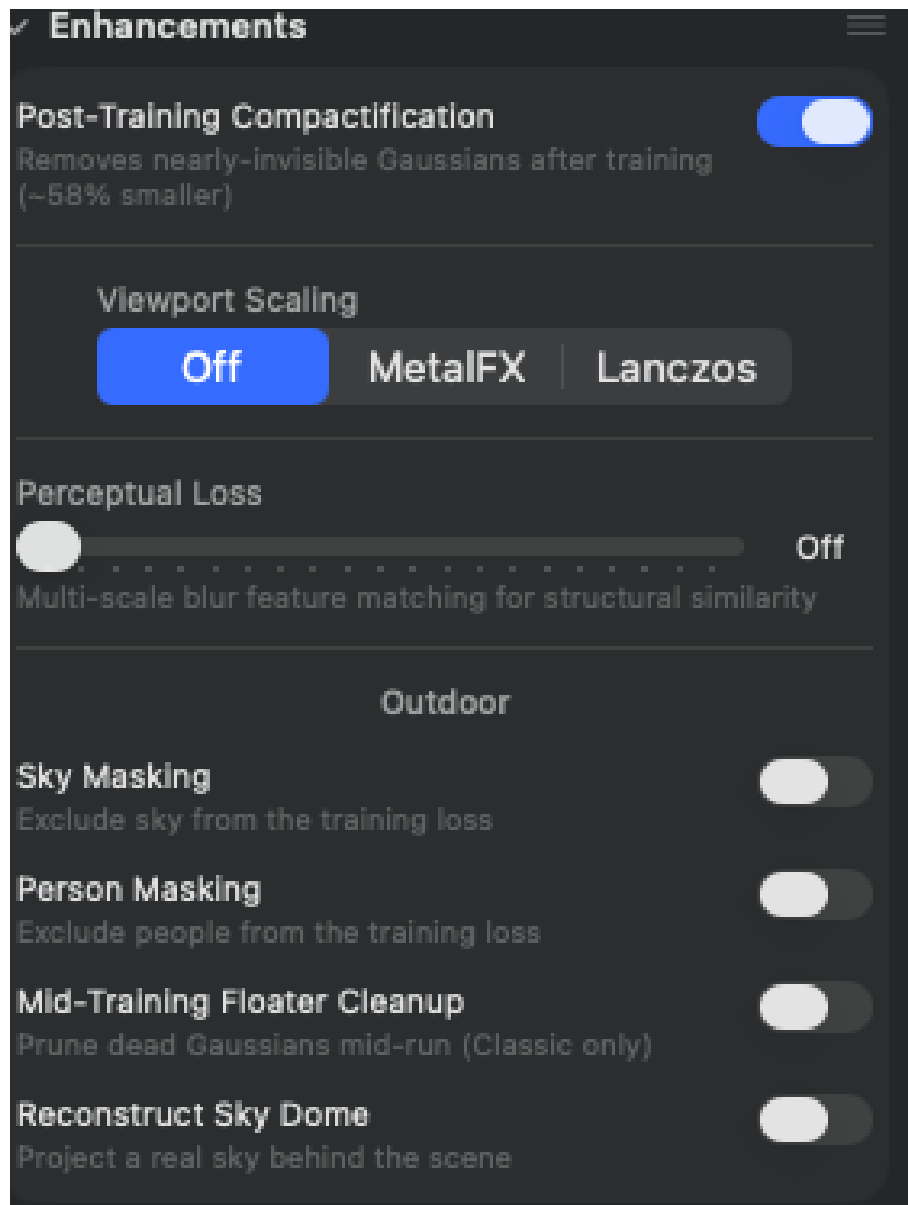


Figura 12: Crop solo della sezione Enhancements — tre righe: Post-Training Compactification (toggle on), Viewport Scaling (picker segmentato Off/MetalFX/Lanczos), Perceptual Loss (slider su «Off»). Ogni riga con sottotitolo spiega la funzione

La sezione Enhancements raggruppa tre funzionalità che migliorano la qualità dell'immagine senza modificare il loop di training core. Le prime due (I26-I27) sono stadi **post-training** o **viewport**: la compactification fa pulizia dopo la fine del training, il Viewport Scaling è un puro renderer del viewport che non influenza il training in corso. La Perceptual Loss (I29) è nonostante l'appartenenza alla sezione una componente del training — viene attivata durante il training come termine di loss aggiuntivo, da qui la separazione dai controlli del viewport tramite un divisore. A partire dalla v1.6 la sezione ha anche un gruppo Outdoor (I42–I44: Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome) — opzioni di training contro i floater nel cielo che un tempo si trovavano nella finestra delle Impostazioni e ora risiedono qui per progetto.

**I26 Toggle Post-Training Compactification**

Inspector → Enhancements → Post-Training Compactification.



Attiva il post-processing V443: dopo la conclusione delle iterazioni di training vengono eliminate le gaussiane con opacità sotto 0.01 (1% di visibilità). Empiricamente ciò riduce la dimensione del file di ~55-58% senza perdita di qualità visibile — perché queste gaussiane non contribuiscono comunque visivamente. La compactification viene eseguita come pass compatto GPU e dura a seconda del gaussian count frazioni di secondo a pochi secondi. Non influenza le prestazioni del training. Quando questo toggle è spento, vengono esportate anche le gaussiane invisibili — rilevante solo se vuoi usare il formato per un ulteriore stadio di training (Continue Training), altrimenti spreco di memoria.

 IN PAROLE SEMPLICI

Pulisce dopo il training le gaussiane che comunque non puoi vedere (opacità sotto 1%). Rende i file di esportazione circa la metà più piccoli (55-58% di riduzione di dimensione) senza perdita di qualità visibile. Viene eseguito come breve pass GPU dopo l'ultima iterazione, dura solo frazioni di secondo a pochi secondi. Praticamente dovrebbe essere sempre attivo — l'unico motivo per spegnerlo è se vuoi continuare il training in seguito tramite Continue Training e devi mantenere anche le gaussiane invisibili. Per i normali flussi di lavoro di esportazione, lascialo semplicemente attivo.

**I27 Picker Viewport Scaling**

Inspector → Enhancements → Viewport Scaling (picker segmentato con tre opzioni: Off, MetalFX, Lanczos).



Un unico picker segmentato che sceglie l'upscaler del viewport — le tre opzioni sono **mutuamente esclusive**. Se la risoluzione di training (tramite I22 Render Scale) è inferiore alla dimensione del viewport, la modalità scelta scala il frame renderizzato alla dimensione di visualizzazione. **Off** = semplice stiramento bilineare. **MetalFX** = l'upscaler spaziale di Apple basato su ML, l'opzione più nitida (il modello ML è ottimizzato per bordi netti), overhead circa 1-2ms per frame su GPU M3. **Lanczos** = i Metal Performance Shaders di Apple con ricampionamento sinc 8-tap, classico senza ML, overhead minimo (< 0.5ms), qualità inferiore a MetalFX, ma senza il tipico «addolcimento» ML delle strutture lineari fini. La pipeline del renderer viene riconfigurata in diretta al cambio — visibile immediatamente, senza riavvio. **Contesto:** in passato erano due toggle separati (MetalFX + Lanczos) che potevano essere attivi contemporaneamente — uno stato contraddittorio in cui MetalFX prevaleva silenziosamente su Lanczos. Il picker elimina questo stato; un eventuale stato «entrambi attivi» ereditato da sessioni più vecchie si risana da solo al prossimo cambio passando a MetalFX. Ha effetto **solo** sul live viewport, non sulle esportazioni renderizzate (video orbit, screenshot) — che vengono renderizzate a piena risoluzione sorgente.

 IN PAROLE SEMPLICI

Migliora la nitidezza del live picture nel viewport — particolarmente utile se lavori con risoluzione di training ridotta (Render Scale 50%, vedi I22). Tre livelli, di cui solo uno è attivo alla volta: «Off» si limita a stirare i pixel, «MetalFX» usa il machine learning di Apple per i bordi più nitidi (praticamente sempre la scelta migliore), «Lanczos» è il filtro classico senza ML — prendilo come fallback se in una scena MetalFX ti leviga le linee o mostra artefatti. Agisce dal vivo, senza riavvio. Ha effetto solo nel live viewport, non sui video orbit esportati o screenshot — vengono renderizzati a piena risoluzione sorgente. Diversamente da prima, non puoi più scegliere per errore due modalità contemporaneamente.

**I29 Slider Perceptual Loss**

Inspector → Enhancements → Perceptual Loss.



Slider 0.0–0.2 in passi di 0.01, visualizzazione a 0.0 come «Off», altrimenti come «0.05» ecc. Attiva un termine di loss aggiuntivo che confronta il Gaussian blur multi-scala del rendering con l'immagine ground truth (3 scale di blur). Cattura differenze strutturali che L1+SSIM da soli non riconoscono. Implementazione V460. Empiricamente un valore di 0.05-0.1 migliora lo score L1 nelle sessioni di qualche punto percentuale, ma costa ~5% di tempo di training (forward pass aggiuntivo attraverso i blur kernel). Sopra 0.15 il training diventa instabile e L1 peggiora di nuovo (il termine di loss domina l'ottimizzazione). Ha effetto **durante** il training, non nel post-processing — nonostante la posizione nella sezione «Enhancements» non è dunque un puro miglioramento successivo.

**IN PAROLE SEMPLICI**

Un termine di loss aggiuntivo che verifica la somiglianza strutturale dell'immagine attraverso tre diversi livelli di sfocatura. Aiuta soprattutto con scene con strutture fini come capelli, tessuto o vegetazione, perché cattura pattern che L1+SSIM da soli non vedono. Valori più piccoli sono più sicuri — da 0.05 a 0.1 è il sweet spot, sopra 0.15 il training diventa instabile e il loss peggiora di nuovo. A 0 (Off) la funzione è completamente spenta e non costa nulla; attivo ingoia circa il 5% di tempo di training per il forward pass extra attraverso i blur kernel. Ha effetto nonostante la sezione «Enhancements» direttamente durante il training, non solo nel post-processing.

**I42 Sky Masking**

Inspector → Enhancements (gruppo Outdoor) → Toggle «Sky Masking». Bound: `AppState.trainingConfig.skyMaskingEnabled` (per progetto, `@DefaultFalse`). Default: `false`.



Attiva la segmentazione pre-training dei pixel del cielo basata su Apple Vision. Prima dell'inizio del training, per ogni fotocamera di input viene estratta la regione del cielo tramite la maschera di foreground di Apple Vision (Sky = Background) e assegnata come maschera per-pixel alla rispettiva fotocamera. Durante il training, il contributo del loss per pixel viene moltiplicato per il complemento della maschera del cielo — i pixel del cielo contribuiscono 0 al gradiente, così che le gaussiane che si proiettano nel cielo non ricevono segnali di ottimizzazione e quindi non diventano «più dense» o «più luminose». Riduce significativamente i floater (grumi scuri nel cielo) in scene outdoor/drone. Costa ~3% di regressione L1 nel training classico 40K (vedi [memory/dev\\_outdoor-floater-reduction.md](#)). Utile solo per scene outdoor con cielo chiaramente riconoscibile; per scene interne o sfondi bianchi la segmentazione del cielo identifica aree sbagliate e blocca segnali di loss validi. Il valore viene ora memorizzato per progetto (non più a livello di app globale) e segue il preset / il file di scena.

 IN PAROLE SEMPLICI

Nelle riprese outdoor con cielo nell'inquadratura, si formano spesso grumi neri o colorati nel cielo — i cosiddetti «floater». Questa opzione rileva automaticamente dove si trova il cielo e dice al training: «Lascia stare il cielo.» Funziona molto bene per voli di drone e scene paesaggistiche. Per interni o sfondi scuri può peggiorare l'immagine — quindi attivala solo quando è visibile cielo reale. Dettagli: [memory/dev\\_outdoor-floater-reduction.md](#).

## I43 Mid-Training Floater Cleanup



Inspector → Enhancements (gruppo Outdoor) → Toggle «Mid-Training Floater Cleanup». Bound: `AppState.trainingConfig.floaterCleanupEnabled` (per progetto, `@DefaultFalse`). Default: `false`.

### TECNICO

Abilita due passaggi aggiuntivi di density control durante il training Classic 40K (preset «P4 Quality»): all'iterazione 20,000 e all'iterazione 30,000. Entrambi i passaggi scansionano tutte le gaussiane secondo tre criteri: (a) opacità molto bassa (default 0.005), (b) dimensione minuscola in screen-space, (c) nessun contributo al loss nelle ultime 1000 iterazioni. Le gaussiane che soddisfano tutte e tre le condizioni vengono purgate. Effetto: ~5–15% di gaussiane in meno alla fine del training, visibilmente meno grumi scuri nel cielo per scene drone/outdoor. Costa ~1–3% di regressione L1 su scene indoor close-up, quindi non abilitato come default. Il valore viene memorizzato tra i riavvii (a differenza di S7). Le due iterazioni di cleanup (20K, 30K) sono hard-coded e attualmente non possono essere modificate tramite UI; per training più brevi (ad es. P2 Preview 5K) il toggle non ha effetto, perché non raggiunge mai i marker di iterazione. **Nuovo:** il toggle è ora azionabile solo quando il preset attivo usa il densifier **Classic** (`densificationStrategy == .classic`). Con MCMC o Hybrid viene **disabilitato** e compare un avviso inline, perché queste strategie gestiscono comunque da sole le gaussiane morte (MCMC tramite relocation, Hybrid tramite logica combinata di reloc/noise) — lì i passaggi manuali di cleanup sarebbero inutili o controproducenti. Riferimento codice: `RadianceKitApp.swift`, tab General. Dettagli: `memory/dev_outdoor-floater-reduction.md`.

### IN PAROLE SEMPLICI

Durante il training, a volte si formano punti gaussiani «morti» che non contribuiscono più alla qualità dell'immagine ma occupano memoria. Questa opzione pulisce due volte durante un training lungo (a 20K e 30K iterazioni) e rimuove questi cadaveri. Per scene outdoor con cielo è particolarmente utile, perché la maggior parte dei floater si raccoglie lì. Per piccoli training o primi piani di mobili, non davvero necessario. L'interruttore si può accendere solo quando il tuo preset usa il densifier Classic — con preset MCMC o Hybrid è in grigio (con una breve spiegazione), perché quelli ripuliscono da soli i loro punti morti.

**I44 Reconstruct Sky Dome**

DOVE

Inspector → Enhancements (gruppo Outdoor)  
→ Toggle «Reconstruct Sky Dome». Bound:  
AppState.trainingConfig.skyDomeEnabled (per progetto, @DefaultFalse ). Default: false .



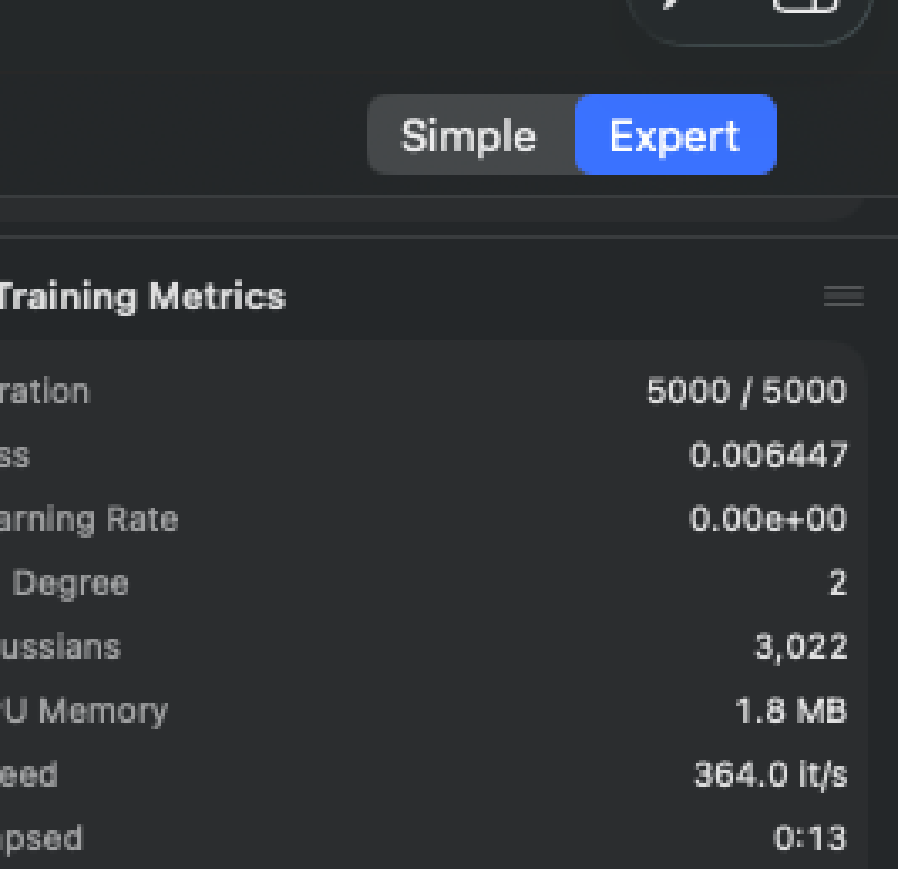
TECNICO

Attiva la proiezione pre-training del Sky Dome (V549e MVP). Dopo l'SfM e prima dell'inizio del training, per ogni fotocamera di input viene estratta dall'immagine la maschera del cielo Apple Vision condivisa con S7, i pixel del cielo vengono de-proiettati utilizzando gli intrinseci della fotocamera su una superficie sferica virtuale (raggio predefinito 8x il raggio della scena). Su questa sfera vengono inizializzate ~5000 nuove gaussiane con valori medi di colore dai pixel del cielo proiettati, scala molto grande (1.0 in unità di scena) e opacità iniziale 0.95. Queste 5000 gaussiane non sono una maschera del cielo in senso classico — vengono addestrate come tutte le altre, ma l'opacità iniziale alta le mantiene in un guscio sottile. Risultato: per viste novel a 360° in scene outdoor/drone, appaiono il colore reale del cielo e strutture nuvolose invece di grumi di coriandoli scuri. Il valore viene memorizzato tra i riavvii. Utile solo per scene outdoor con almeno copertura della fotocamera a 360°; per catture oggetti puri senza vista del cielo non ha alcun effetto. Stato: sperimentale, validazione A/B più ampia su più set outdoor ancora in sospeso.

**IN PAROLE SEMPLICI**

Invece di lasciare che il training cerchi di «indovinare» il cielo dai pochi pixel visibili (cosa che porta a floater), RadianceKit proietta i pixel del cielo direttamente su una sfera virtuale attorno alla scena prima che il training inizi. Quando poi ruoti la scena finita a 360°, vedi cielo reale invece di grumi neri. Funziona solo per riprese outdoor in cui è effettivamente presente cielo nell'inquadratura. Per scansioni di soggiorno o riprese in studio non porta nulla.

## Sezione Metriche (I30–I38)



Training Metrics	
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: Crop solo della sezione Training Metrics dopo training completato sul bouquet (5K iterazioni, 2 991 gaussiane finali) — tabella con metriche di training (Iteration, Loss, SSIM Loss, Combined Loss, Gaussian Count, Learning Rate, Elapsed, ETA)

Mentre un training è in esecuzione, la sezione Metriche mostra nove valori live dal loop di training. Prima dell'avvio di un training la sezione è vuota («Start training to see live metrics»). Tutti i valori vengono aggiornati ogni ~30 iterazioni (frequenza di aggiornamento dello stream). La sezione è in sola lettura — nessun elemento è cliccabile o modificabile. Per un'analisi più approfondita consulta i log JSONL di training sotto `~/Documents/RadianceKit/Logs/` (script python3 `scripts/analyze_logs.py best 5`).

**I30 Iteration**

Inspector → Metriche → Iteration. Sola lettura.

 **TECNICO**

Visualizzazione nel formato «4523 / 40000» — iterazione attuale su iterazioni totali pianificate. Conta sincrono con il loop di training, che spinge i valori ogni ~30 iter. Il secondo numero corrisponde al valore Max Iterations al momento dell'avvio; non cambia più anche se l'utente regola lo stepper dopo — l'esecuzione in corso usa la sua copia snapshot. Se l'app aggiunge iterazioni tramite il menu Training (Continue Training +5K/+10K/+20K), il denominatore aumenta.

 **IN PAROLE SEMPLICI**

Dove si trova attualmente il training. «4523 / 40000» significa: 4523 di 40 000 passi sono fatti, quindi circa l'11% finito. Il numero a sinistra conta secondo dopo secondo; se si ferma per minuti, il training è bloccato — di solito un indizio di GPU throttling o un'app concorrente. Il numero a destra corrisponde al valore Max Iterations (I18) all'avvio del training e non cambia più, anche se regoli lo stepper dopo. Con Continue Training (+5K/+10K/+20K) cresce con i passi aggiuntivi.

**I31 Loss**

Inspector → Metriche → Loss. Sola lettura.

 **TECNICO**

Valore float con sei decimali (ad es. «0.024385»). Misura il loss combinato L1+SSIM (mix controllato tramite I21 SSIM Weight) più opzionalmente Perceptual Loss (I29) e altri regolarizzatori. La scala non è assoluta, ma dipende dalla scena — richiede lo stesso dataset per la maggior parte dei confronti. Valori finali tipici con buone configurazioni: - Classic Quality 40K iter: 0.022–0.025 (Horse, Truck, Garden) - MCMC Full 200K iter: 0.024–0.028 - Outdoor drone 30K: 0.030–0.060 (peggiore per motivi di geometria) - Indoor appartamenti: 0.018–0.025

Valori sopra 0.10 dopo 5K iterazioni indicano problemi SfM (pose della fotocamera scadenti) — interrompi e ricalcola l'SfM.

 **IN PAROLE SEMPLICI**

Quanto il rendering si discosta ancora dall'originale — combinato da L1, SSIM ed event. Perceptual Loss. Più piccolo è meglio. Sotto 0.03 è solitamente davvero buono, sotto 0.05 ancora ok, le scene outdoor sono per motivi di geometria piuttosto a 0.03-0.06. Sopra 0.10 dopo diverse migliaia di iterazioni è un segnale di allerta — di solito è la ricostruzione delle fotocamere (l'SfM non ha funzionato correttamente). La scala non è assoluta, ma dipende dalla scena; fai confronti solo all'interno dello stesso dataset. Se il numero salta improvvisamente in alto, è di solito accaduto un evento di gradient explosion.

**I32 Learning Rate**

DOVE

Inspector → Metriche → Learning Rate. Sola lettura.



TECNICO

Visualizzazione in notazione scientifica (ad es. «1.60e-04»). Learning rate attuale per i parametri di posizione (3DGS ha sei LR indipendenti per posizione, SH-DC, SH-Rest, opacità, scala, rotazione — qui viene mostrata la posizione LR come dimensione rappresentativa). Valore iniziale predefinito 1.6e-4, che attraverso un decadimento esponenziale scende a ~1.6e-6 alla fine del training. Il decadimento è regolabile tramite il campo LR-schedule nella configurazione del training (campo T nel Cap. 6). Se la LR rimane insolitamente alta (ad es. 1e-3 o più dopo 10K iterazioni), potrebbe indicare una configurazione caricata in modo errato.

**IN PAROLE SEMPLICI**

Quanto sono grandi i passi di ottimizzazione — in concreto il learning rate per le posizioni delle gaussiane. Parte da 1.60e-04 e scende esponenzialmente a circa 1.60e-06 alla fine del training («1.60e-06» = 0.0000016). L'andamento avviene automaticamente, non devi regolare nulla qui. Se il valore dopo 10 000+ iterazioni è ancora maggiore di 1e-3, è probabilmente stata caricata una config errata — interrompi il training e seleziona di nuovo il preset. Internamente 3DGS ha sei learning rate indipendenti (posizione, SH-DC, SH-Rest, opacità, scala, rotazione); qui vedi solo la posizione LR come rappresentante.

**I33 SH Degree**

DOVE

Inspector → Metriche → SH Degree. Sola lettura.



TECNICO

Intero 0-3. Grado di spherical harmonics per la rappresentazione del colore. Parte da 0 (solo la componente DC, ossia colore indipendente dalla direzione per gaussiana — quindi solo una costante RGB) e cresce progressivamente fino a 3 nel corso del training. Lo schedule standard solleva il grado di 1 a 1000/2000/3000 iterazioni. SH-3 corrisponde a 48 coefficienti di colore per gaussiana (3 canali RGB × 16 funzioni base SH). Grado SH più alto = più riflessione dipendente dalla direzione (le superfici lucide appaiono correttamente diverse da angoli di visualizzazione differenti), ma anche più memoria e training più lento.

**IN PAROLE SEMPLICI**

Quanto è complessa attualmente la rappresentazione del colore per gaussiana. Parte da 0 (solo un colore indipendente dalla direzione per punto) e viene portato gradualmente a 3 — tipico alle iterazioni 1000, 2000 e 3000. Lo stadio 3 significa 48 coefficienti di colore per gaussiana e permette riflessioni dipendenti dalla direzione, ossia che le superfici lucide appaiono correttamente diverse da angoli di visualizzazione differenti. Non hai bisogno di toccarlo attivamente, lo schedule avviene automaticamente. Un grado più alto costa più memoria e rallenta leggermente il training — ma questo è il prezzo per luci speculari realistiche.

## I34 Gaussians



DOVE

Inspector → Metriche → Gaussians. Sola lettura.



TECNICO

Numero attuale di gaussiane nel modello, formattato con separatore di locale (ad es. «524.318»). Crescita: - Classic: parte dagli initpoint SfM (tipico 50K-300K), cresce tramite clone/split fino a poco prima di Densify Until, poi statico fino alla fine del training (modulo pruning) - MCMC: i sample point vengono aggiunti fino al cap MCMC, poi solo rilocalizzazione

Valori finali sani: - Classic Quality: 400K-700K (Horse 524K, Garden 800K) - MCMC Full: esattamente al cap (default 150K, con auto-scale multiplier × SfM count a seconda della scena 500K-1.5M)

Con MCMC il numero scende al < 60% del cap → anomalia (indicatore di collapse, indica regolarizzatori troppo aggressivi).

### IN PAROLE SEMPLICI

Quanti punti gaussiani ha attualmente il modello 3D. Cresce durante il training, finché non viene raggiunto Densify Until (I20); dopo il numero rimane praticamente costante. Più punti = più dettaglio, ma anche file più grande e rendering più lento nel viewport. 500.000 gaussiane è un valore intermedio tipico per Classic Quality su una scena media; MCMC Full a seconda dell'auto-scale (I17) finisce tra 500K e 1.5M. Se il numero in MCMC scende improvvisamente sotto il 60% del cap, è un indicatore di collapse — di solito regolarizzatori troppo aggressivi.

## I35 GPU Memory



DOVE

Inspector → Metriche → GPU Memory. Sola lettura.



TECNICO

Stima del consumo di memoria del buffer gaussiano come gaussian count × 616 byte (formattato in stile memoria). 616 byte è la dimensione empirica di una gaussiana completamente equipaggiata (posizione, scala, rotazione, opacità, coefficienti SH grado 3, accumulatore di gradiente). La visualizzazione **non** cattura l'overhead del renderer (tile buffer, sort buffer, backward buffer) — il consumo reale di memoria GPU è tipicamente 2-3× sopra questo valore. A 500K gaussiane: visualizzato ~290 MB, reale ~700 MB. A 1.5M gaussiane: visualizzato ~880 MB, reale ~2.5 GB. Su M3 Max con 64+ GB di memoria unificata non critico, su M3 Pro con 18 GB già un limite.

### IN PAROLE SEMPLICI

Una stima di quanta memoria GPU occupano le gaussiane stesse — circa 616 byte per punto. Il consumo GPU reale è 2-3× più alto di quanto visualizzato, perché il renderer aggiunge ancora i propri tile, sort e backward buffer. Su un MacBook con 16-18 GB di memoria unificata dovresti rimanere sotto 500K gaussiane; con M3 Max o Studio (64+ GB) puoi tranquillamente guidare 1.5M e oltre. Se il training crasha improvvisamente o il sistema fa swap, di solito qui è raggiunto il limite — abbassa Render Scale (I22) o riduci Densify Until (I20).

**I36 Speed**

Inspector → Metriche → Speed. Sola lettura.

 **TECNICO**

Iterazioni al secondo con una cifra decimale («24.3 it/s»). Calcolato dal trainer come media mobile sulle ultime ~100 iterazioni. Valori tipici: - Preset Quick (1K iter): 80-120 it/s (breve, nessuno steady state) - Classic 20K @ 1.0 Render Scale (scena 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 (più lento per la rilocazione) - Con 1M+ gaussiane e risoluzione piena: < 10 it/s

Speed in calo nel corso del training è normale — più gaussiane = più compute per iterazione. Crolli improvvisi (ad es. da 30 → 5 it/s) indicano thermal throttling della GPU o app concorrenti.

 **IN PAROLE SEMPLICI**

Quanto velocemente sta girando il training, in iterazioni al secondo. Si attesta tipicamente a 20-50 it/s, con Render Scale ridotta (50%) e piccole scene anche 80-120 it/s. Scende durante il training in modo del tutto naturale, perché più gaussiane = più lavoro per iterazione. Crolli improvvisi (ad es. 30 → 5 it/s) indicano thermal throttling della GPU o app concorrenti — tab del browser con video, backup Time-Machine, indicizzazione Photos. Tenere l'app in primo piano e chiudere i programmi in background spesso aiuta. Con 1M+ gaussiane e risoluzione piena, sotto 10 it/s è normale.

**I37 Elapsed**

Inspector → Metriche → Elapsed. Sola lettura.

 **TECNICO**

Tempo già trascorso come «4:23» (m:ss) o «1:23:45» (h:mm:ss). Cambio di formato a partire da 1 ora. Misura solo il puro tempo di training, non le fasi precedenti (calcolo SfM, importazione immagini). Con pausa/resume il cronometro continua — è quindi wall clock, non tempo CPU.

 **IN PAROLE SEMPLICI**

Da quanto tempo gira il training, come puro cronometro (tempo wall clock). Il formato è «m:ss» fino a un'ora, poi «h:mm:ss». Non «tempo CPU», ma «da quanto stiamo aspettando» — quindi anche i tempi di pausa contano. Misura solo la fase pura di training, non il calcolo SfM precedente o l'importazione delle immagini. Utile per il confronto con l'ETA (I38) — se Elapsed supera nettamente l'ETA originale, il training è diventato più lento da qualche parte rispetto al previsto.

**I38** ETA

Inspector → Metriche → ETA. Sola lettura.

 **TECNICO**

Tempo residuo stimato come «17:42» o «1:12:35». Calcolo:  $(\text{Max Iterations} - \text{iterazione attuale}) / \text{iterazioni-al-secondo}$ . Mostra «-» se la velocità è attualmente zero (proprio all'inizio o in pausa). La stima **non** viene adattata al tipico rallentamento verso la fine del training — soprattutto con MCMC e Classic con grandi valori di Densify Until il training tende a diventare più lento, perché entrano sempre più gaussiane nell'immagine. In realtà rimane tipicamente 10-20% sopra l'ETA iniziale.

 **IN PAROLE SEMPLICI**

Quanto tempo si dovrà ancora attendere — calcolato dalle iterazioni residue e dalla speed attuale (I36). Una stima approssimativa: se il Mac diventa improvvisamente più lento (più gaussiane dalla fase di densify, thermal throttling, altre app), può durare più a lungo di quanto visualizzato. L'app non calcola il tipico rallentamento verso la fine del training, quindi la fine reale finisce di solito 10-20% sopra l'ETA iniziale. Calcola +15%, allora di solito quadra. Mostra «-» se la velocità è attualmente 0 (inizio training o pausa).

## Sezione Grafico di perdita (I39–I41)

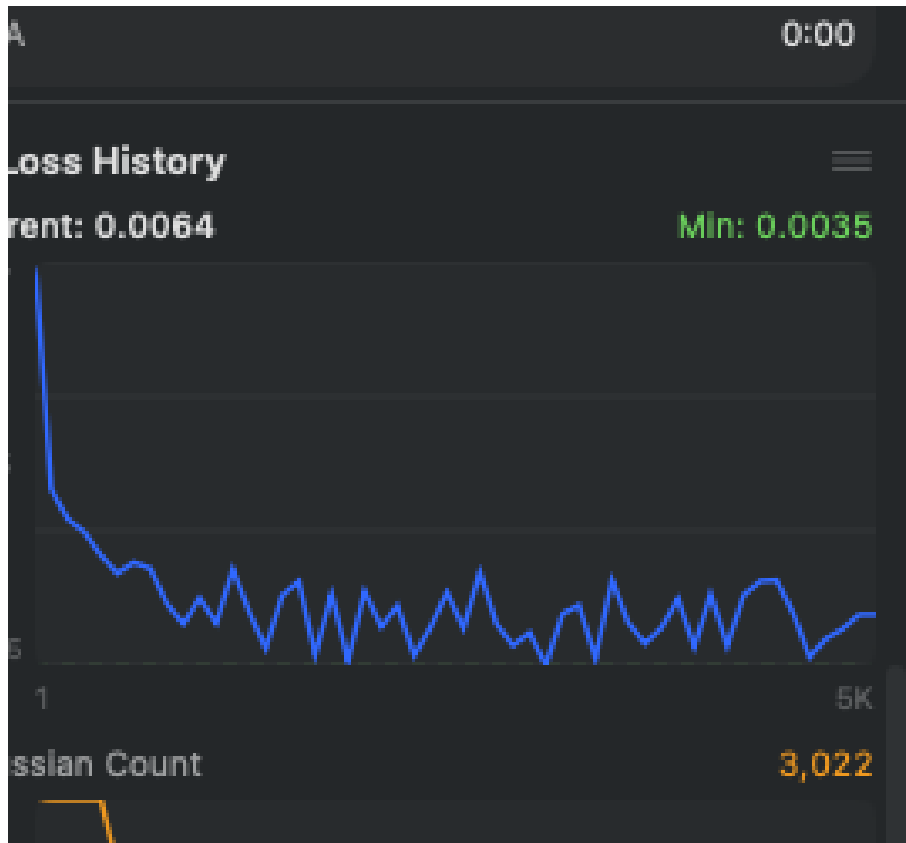


Figura 14: Crop solo della sezione Loss History dopo training completato — Current 0.0064, Min 0.0035 (verde), andamento blu da 0.027 (iterazione 1) a 0.0035 (iterazione 5K) con caratteristico gomito attorno a iter 200, sotto grafico Gaussian Count arancione

La sezione Grafico di perdita visualizza l'andamento del training nel tempo. È costituita da due grafici: un grafico Loss-Curve (grande, in alto, blu) e un grafico Gaussian Count (più piccolo, in basso, arancione). Entrambi vengono costruiti in tempo reale durante il training e persistono fino al prossimo avvio del training. Prima del primo training l'area è vuota («Loss curve will appear during training»). I grafici sono puri disegni di SwiftUI Path (nessun framework Swift Charts), in modo che renderizzino fluidamente anche con 100K+ punti.

**I39** Current Loss (visualizzazione)

Inspector → Grafico di perdita → area etichetta sinistra «Current: 0.0287». Sola lettura.



Valore float dell'ultimo loss sample point, formattato con quattro decimali. Identico a I31 (Loss nella sezione Metriche), solo qui formattato in modo più compatto. La fonte è la Loss History — una lista che riceve una voce ogni ~30 iterazioni. Solo valori finiti vengono inseriti nella lista — NaN/Infinity (molto raro, in caso di un bug di gradient explosion) vengono filtrati.

**IN PAROLE SEMPLICI**

Il valore loss attuale in notazione più breve rispetto alla sezione Metriche (quattro decimali). Contutisticamente identico a I31, ma qui la visualizzazione si trova direttamente sul grafico di loss e ti dà il valore numerico esatto mentre osservi la curva. Viene aggiornato come tutte le metriche live ogni 30 iterazioni. Valori NaN o Infinity (estremamente rari con bug di gradient explosion) l'app li filtra automaticamente. Utile per non dover saltare nell'altra sezione mentre guardi il diagramma.

**I40** Min Loss (visualizzazione)

Inspector → Grafico di perdita → area etichetta destra «Min: 0.0245» (verde). Sola lettura.



Minimo di tutti i valori loss mai visti dell'esecuzione di training attuale. Viene ricalcolato dal vivo dalla Loss History — nessuna persistenza separata. Viene rappresentato in verde, perché «Min» = «Best so far». La linea verde tratteggiata sul bordo inferiore del grafico segna visivamente questa posizione Y. In sessioni di Continue Training il tracking del minimo riparte — la vecchia history nell'UI viene sostituita da quella nuova (non appesa). Se il training attuale procede peggio del precedente, la visualizzazione Min può quindi essere più grande del risultato finale precedente.

**IN PAROLE SEMPLICI**

Il valore loss più basso che questo training ha visto finora — rappresentato in verde, perché «best so far». La linea verde tratteggiata sul bordo inferiore del grafico segna anche visivamente questa posizione. Se la curva attuale è attualmente nettamente sopra, c'è ancora con fortuna un miglioramento; di solito Min è però l'indicazione del risultato finale che ti interessa in seguito. In sessioni di Continue Training il tracking del Min riparte, perché la vecchia history nell'UI viene sostituita da quella nuova — il valore Min può quindi apparire peggiore del risultato finale precedente.

## I41 Grafico Gaussian Count



DOVE

Inspector → Grafico di perdita → secondo grafico sotto (arancione). Sola lettura.



TECNICO

Grafico a linee del numero di gaussiane sulle iterazioni di training. Fonte: la Gaussian Count History (lista di coppie (iter, count), riempita dal trainer ogni ~30 iter). Scala Y dinamica tra minimo e massimo della history. Con strategia Classic la curva appare tipicamente così: in costante aumento fino a Densify Until, poi piatta (con piccole oscillazioni di pruning). Con MCMC: salita ripida fino al cap, poi linea orizzontale (la rilocazione mantiene il numero costante). Se la curva **scende** nonostante il training attivo, la densification fa pruning troppo aggressivo — indizio di default sbagliati o di un noto bug MCMC collapse (tema dell'hotfix 1.4.4).

### IN PAROLE SEMPLICI

Come si sviluppa il numero di gaussiane nel tempo di training — il grafico arancione più piccolo sotto la curva Loss. Con strategia Classic la linea sale costantemente, finché non viene raggiunto Densify Until (I20), dopo rimane piatta con piccole oscillazioni di pruning. Con MCMC schizza ripida sul cap e poi rimane orizzontale, perché la rilocazione mantiene il numero costante. Se la curva improvvisamente piega in giù nonostante il training attivo, la densification è troppo aggressiva nel pruning — classico segno del bug MCMC collapse di v1.4.4. Allora aiuta l'aggiornamento dell'app o un ritorno a Classic.

### Come si legge la curva di perdita?

Il grafico Loss è lo strumento diagnostico più importante nell'Inspector — nessun altro indicatore mostra in modo così diretto se il training procede utilmente o si è bloccato. La forma sana tipica è una rapida discesa nelle prime 1000-3000 iterazioni (da ~0.15 a ~0.05), seguita da una discesa lenta e uniforme fino alla fine del training (a 0.020-0.030). Logaritmicamente la curva appare in tal caso come una diagonale liscia.

**Cosa significa un plateau nel Loss?** Se la curva rimane piatta per diverse migliaia di iterazioni, ci sono due possibili letture: (a) Il training è «convergente» — il loss non può più scendere significativamente, perché il modello è buono quanto può essere con i dati e le impostazioni date. Ciò è desiderato; è «finito». (b) Il training si «blocca» — il loss potrebbe in realtà ancora scendere, ma l'ottimizzazione ristagna (minimo locale, learning rate troppo piccolo, densification spenta). Distinguere: se il valore loss si trova in un intervallo tipicamente buono (0.020-0.030 con indoor/oggetto, 0.040-0.060 con outdoor) e la curva è piatta da 5K iterazioni, è convergente. Se il valore è nettamente più alto rispetto a scene paragonabili (ad es. 0.08), si blocca.

**Attenzione: plateau gaussiane ≠ plateau loss.** Un plateau nel numero di gaussiane **non** significa «il training è finito». Significa solo che la densification ha smesso di aggiungere nuovi punti — o perché è stato raggiunto (Classic) o perché il cap MCMC è pieno. Il training prosegue dopo e raffina solo i punti esistenti. Il vero segnale di «finito» lo leggi nella curva Loss e nella visualizzazione di iterazione (I30), non qui.

**Regola pratica per interrompere:** Se la curva Loss dopo 5000+ iterazioni è sopra 0.08 e scende a malapena, è molto probabile che la ricostruzione SfM sia storta. Interrompi

il training, consulta il Capitolo 9 se il backend SfM scelto si adatta alla scena, eventualmente passa a COLMAP/Native, poi riavvia. Meglio investire 10 minuti in un SfM migliore che 2 ore di training con allineamento delle fotocamere scadente.

## **Quando ricorrere all'Inspector?**

Riferimento rapido: quale sezione + quali controlli per quale caso d'uso tipico?

Compito comune	Sezione	ID controllo
Desaturare i colori dello Splat finito	Look	L1 (Saturation)
Arrotondare Splat ad ago/confetti	Look	L2 (Splat length)
Riempire nuvola bucherellata / ingrandire Splat	Look	L3 (Splat size)
Dissolvere i «far-confetti» lontani negli orbit	Look	L4 (Fade far region)
Scartare le regolazioni Look	Look	L5 (Reset finishing)
Caricare setup preconfezionato	Preset	I7 (cliccare riga)
Salvare setup proprio	Preset	I1 → I2 → I4
Condividere setup con colleghi	Preset	I5 (Export) risp. I6 (Import)
Cambiare backend SfM (ad es. perché Apple PG troppo instabile)	Configurazione del training	I12 (vedi Cap. 9)
Elaborare frame video senza lunghezza focale EXIF	Configurazione del training	I13 (FOV Override)
Prestazioni COLMAP: GLOMAP invece di classico	Configurazione del training	I14
Passare da Classic a MCMC	Configurazione del training	I15
Far girare il training più a lungo	Configurazione del training	I18 (Max Iter) + I20 (Densify Until) — accoppiati tramite I19
Dimezzare il tempo GPU	Configurazione del training	I22 (Render Scale al 50%)
+6% qualità training (MCMC)	Configurazione del training	I16 (MCMC Quality)
Scena outdoor con molti punti SfM	Configurazione del training	I17 (Auto-scale by scene)
Impostare/cambiare percorso COLMAP	Configurazione del training	I23 / I24 / I25
Rimpicciolire i file di esportazione	Enhancements	I26 (lasciare sempre acceso)
Viewport più nitido senza più tempo di training	Enhancements	I27 (Viewport Scaling → Metal-FX)
MetalFX leviga troppo → alternativa	Enhancements	I27 (Viewport Scaling → Lanczos)
Ultimo briciolo di dettaglio con strutture fini	Enhancements	I29 (Perceptual Loss 0.05-0.1)
Per un migliore bilanciamento tra qualità e velocità	Enhancements	I30 (Perceptual Loss 0.05-0.1) + I31 (Perceptual Loss 0.1-0.2) + I32 (Perceptual Loss 0.2-0.3) + I33 (Perceptual Loss 0.3-0.4) + I34 (Perceptual Loss 0.4-0.5) + I35 (Perceptual Loss 0.5-0.6) + I36 (Perceptual Loss 0.6-0.7) + I37 (Perceptual Loss 0.7-0.8) + I38 (Perceptual Loss 0.8-0.9) + I39 (Perceptual Loss 0.9-1.0)

## CAPITOLO

## Capitolo 3 — Impostazioni

---

La finestra delle Impostazioni si apre tramite `RadianceKit` →

`Impostazioni...` o la scorciatoia da tastiera standard `⌘, .`. Contiene due tab: **General** e **AI Helpers**. A differenza dei valori dell'Inspector del Capitolo 2, le impostazioni di questa finestra hanno effetto **a livello di app globale** (su tutti i progetti) — vengono persistite e sopravvivono ai riavvii dell'app. Il tab General raggruppa tre sezioni di contenuto: Interface, Viewport e Training. (I tre toggle Outdoor-Floater — Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome — che un tempo si trovavano qui sono stati spostati, a partire dalla v1.6, nella sezione Enhancements dell'Expert Inspector, dove ora vengono memorizzati per progetto; vedi Capitolo 2, I42–I44.) Il tab AI Helpers attiva gli helper di machine learning on-device (Vision, CoreML) per il preprocessing di SfM e training.

I controlli precedenti per attivare o disattivare collettivamente tutti gli AI Helpers non esistono più nella versione attuale — non sono quindi documentati qui. Anche l'area precedente «Coming Soon» per gli helper non ancora distribuiti è stata rimossa e non è referenziata qui.

## Tab General

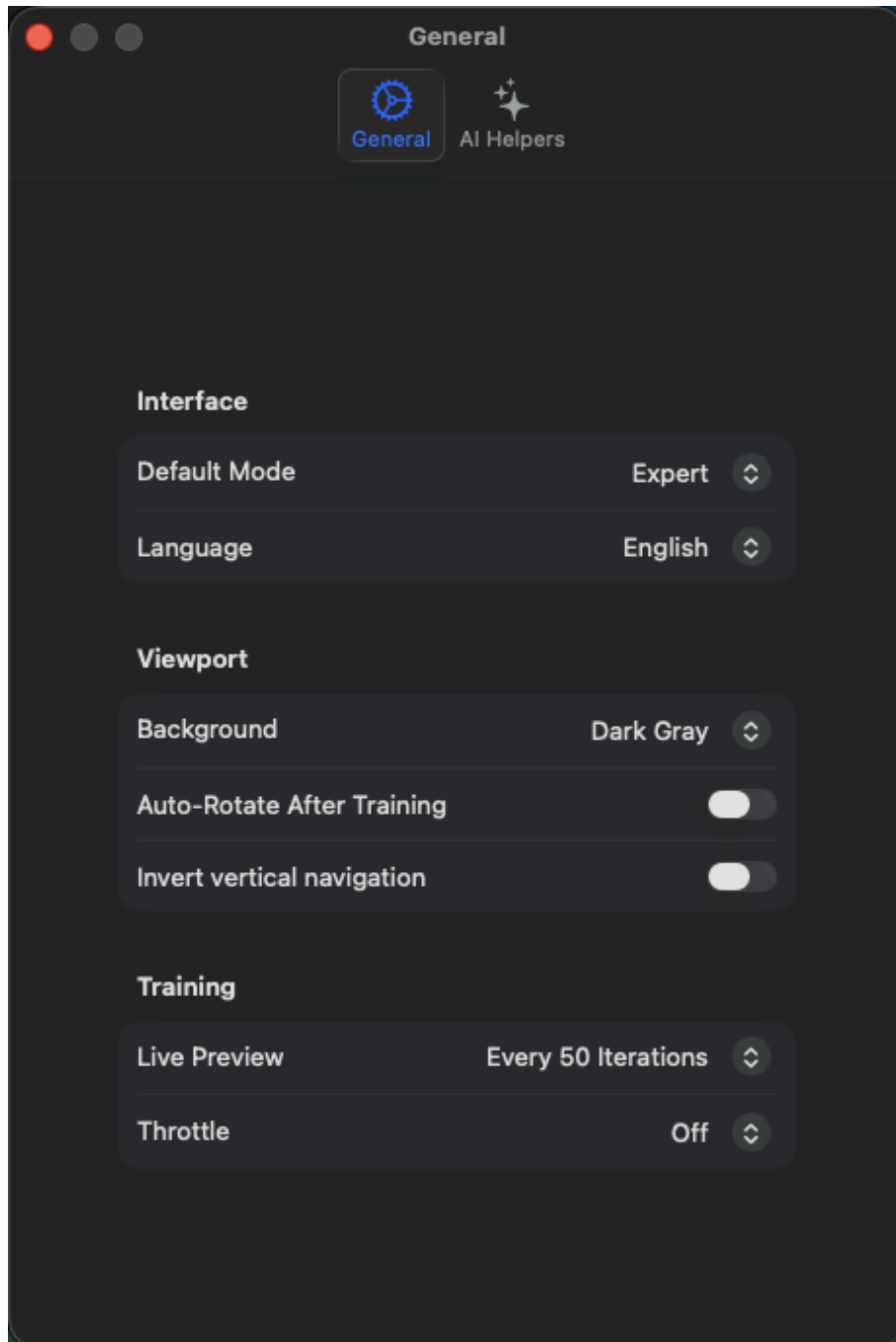


Figura 15: Impostazioni → tab General con sezioni Interface, Viewport, Training e Experimental

## S1 Default Mode



DOVE

Settings → General → Interface → picker Default Mode. Bound: Default: `.simple`.



TECNICO

Controlla in quale delle due modalità UI l'app si apre dopo il prossimo avvio. «Simple Mode» è il flusso di lavoro guidato a wizard in 4 passaggi (Importazione → Elaborazione → Anteprima → Esportazione, documentato nel Capitolo 10 sotto Z1–Z4), «Expert Mode» il layout classico a tre pannelli con Navigator, Viewport 3D ed Expert Inspector del Capitolo 2. Il valore viene memorizzato tra i riavvii. Effetto identico al menu Mode → Simple Mode (⌘1) / Mode → Expert Mode (⌘2), tranne che il menu cambia la sessione in corso mentre questo picker imposta il default per le sessioni future. Entrambe le modalità accedono allo stesso stato del progetto — progetti, fotocamere e configurazione del training vengono preservati al cambio di modalità. I pulsanti della toolbar specifici della modalità vengono rigenerati immediatamente.

### IN PAROLE SEMPLICI

Qui scegli con quale interfaccia RadianceKit si avvia al prossimo lancio. «Simple Mode» è la modalità principianti: quattro passi chiari, preset predefiniti, quasi nessuna opzione. «Expert Mode» è il layout completo della cassetta degli attrezzi con tutti i controlli che vedi nel Capitolo 2. Puoi passare avanti e indietro in qualsiasi momento tramite il menu «Mode» senza perdere immagini o progressi del training.

## S2 Language



DOVE

Settings → General → Interface → picker Language. Bound: Default: `.system` (segue la lingua di macOS).



TECNICO

Seleziona la lingua di visualizzazione dell'intera UI dell'app, indipendentemente dalla lingua di sistema macOS. RadianceKit è localizzato in 17 lingue (`de`, `en`, `pl`, `en-AU`, `ar-SA`, più altre 12). Con «System» l'app segue la lingua macOS. Con una scelta esplicita l'impostazione della lingua viene memorizzata tra i riavvii; l'effetto completo richiede di solito un riavvio dell'app, perché i bundle di localizzazione vengono caricati solo all'avvio. Le 298 chiavi di localizzazione documentate nel progetto sono tutte considerate, comprese le voci nelle sub-view e nei tooltip della guida.

### IN PAROLE SEMPLICI

Se il tuo Mac è in inglese ma preferisci l'interfaccia italiana di RadianceKit (o viceversa), lo imposti qui. La maggior parte dei testi cambia subito. Alcuni dialoghi appaiono nella nuova lingua solo dopo un riavvio dell'app.

### S3 Viewport Background



Settings → General → Viewport → picker Background. Bound: `.darkGray` (RGB 0.1, 0.1, 0.1).



Imposta il colore di sfondo predefinito per il viewport 3D. Tre opzioni: «Dark Gray» (RGB 0.1, 0.1, 0.1 — default), «Black» (0, 0, 0) e «White» (1, 1, 1). L'impostazione persiste il default per nuovi progetti e sessioni tra i riavvii e contemporaneamente aggiorna immediatamente il renderer Metal in esecuzione. Opzioni identiche si trovano nel menu Viewport → Background (M21, M22, M23), ma il picker Settings imposta il default mentre il menu cambia la visualizzazione in corso. Importante per screenshot e video demo: gli sfondi bianchi evidenziano più fortemente i floater verdi/blu, gli sfondi scuri sono migliori per cattura di rendering pulite.

#### IN PAROLE SEMPLICI

Il colore dietro i tuoi modelli 3D nella finestra di anteprima. Grigio scuro è il default e si adatta alla maggior parte delle scene. Bianco va bene per gli screenshot, nero risulta più elegante per le catture di rendering. Puoi cambiare il colore in qualsiasi momento tramite il menu «Viewport → Background» per la scena in corso — questa impostazione definisce solo quale colore sarà nuovamente attivo alla prossima apertura.

### S4 Auto-Rotate After Training



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



Avvia una rotazione continua a piatto rotante della fotocamera del viewport attorno al centroide della scena immediatamente dopo la fine del training (velocità di rotazione predefinita ~0.3 rad/s). Praticamente utile per sessioni demo, confronti A/B e per valutare immediatamente da una vista a 360° se si sono formati «floater» ai bordi della scena. L'effetto è visivamente identico al menu Viewport → Toggle Auto-Rotation (M16, ⌘⌥T), tranne che questo toggle attiva il comportamento automaticamente dopo la fine del training invece che manualmente. Può essere interrotto in qualsiasi momento tramite il menu o cliccando nel viewport (cosa che mette in pausa la rotazione). Non ha alcuna influenza sulle prestazioni del training — la rotazione parte solo quando il training è terminato.

#### IN PAROLE SEMPLICI

Quando attivato, la scena 3D ruota automaticamente non appena il training è finito — come un carosello. Bello quando fai training notturni e vuoi vedere il risultato già in movimento al mattino senza dover cliccare. Per le sessioni lunghe in cui monitori solo il training, lascialo spento.

**S5 Live Preview Interval**

Settings → General → Training  
→ picker Live Preview. Bound:  
`AppState.trainingConfig.livePreviewInterval`.  
Default: 0 (Off).



Determina l'intervallo di iterazione con cui lo snapshot del training in corso viene renderizzato nel viewport 3D. Quattro valori discreti: 0 («Off»), 50, 250, 1000 iterazioni. Con Live Preview attivo, il trainer copia il buffer Gaussian dalla GPU in un buffer di rendering separato e attiva un redraw del viewport. Con «Off», il viewport viene aggiornato solo dopo il completamento del training. Costo prestazionale: ogni 50 iterazioni ~5–10% più lento su M3 Ultra, ogni 250 iterazioni ~1–2% più lento, ogni 1000 iterazioni non misurabile. Overhead di memoria costante ~2 GB per il buffer dello snapshot, indipendentemente dall'intervallo. Il valore serve come default per i nuovi training; dopo l'inizio del training, il Training Inspector mostra il valore live effettivo del training in corso. Con intervallo 50 l'impressione visiva è una «crescita» fluida della nuvola di punti, con 1000 risulta a scatti.

**IN PAROLE SEMPLICI**

Mentre il training è in corso, puoi scegliere quanto spesso la vista 3D viene aggiornata. «Off» significa: nessun aggiornamento durante il training (il più veloce). «Every 50 Iterations» mostra quasi in tempo reale come si compone la tua scena (leggermente più lento). Per una visione comoda di piccoli training, «Every 250» è un buon compromesso.

## S6 Throttle Delay

### DOVE

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

### TECNICO

Inserisce un ritardo artificiale in millisecondi tra le iterazioni del training. Quattro valori discreti: 0 («Off»), 2 («Light»), 5 («Moderate»), 10 («Eco»). Scopo: durante training più lunghi (diverse ore) la GPU verrebbe altrimenti utilizzata al 100%, il che porta a una UI di sistema notevolmente più lenta (cursore del mouse che salta, altre app diventano lente). Il ritardo throttle dà alla GPU pause durante le quali possono essere eseguiti altri compiti. Il costo prestazionale è considerevole: con 5 ms di throttle un tipico training 40K dura circa 50–80% più a lungo che senza throttle. In modalità prestazione «Eco» (10 ms) il ritardo per iterazione è più lungo dell'iterazione stessa — fattore 2–3× più lento. Con throttle attivo appare sotto il picker una nota: «Throttle is on. Training will be slower than usual.» L'app stessa non reagisce in modo notevolmente migliore — solo le altre app ne traggono beneficio.

### IN PAROLE SEMPLICI

Se il tuo Mac diventa troppo caldo durante un training lungo o altri programmi diventano troppo lenti, attiva qui un freno. «Off» dà gas alla GPU (il più veloce). «Light» fa una piccola pausa tra ogni passo (leggermente più lento, ma il sistema reagisce meglio). «Eco» è il freno più forte — buono per training notturni su un MacBook che non deve scaldarsi troppo.

## Tab AI Helpers

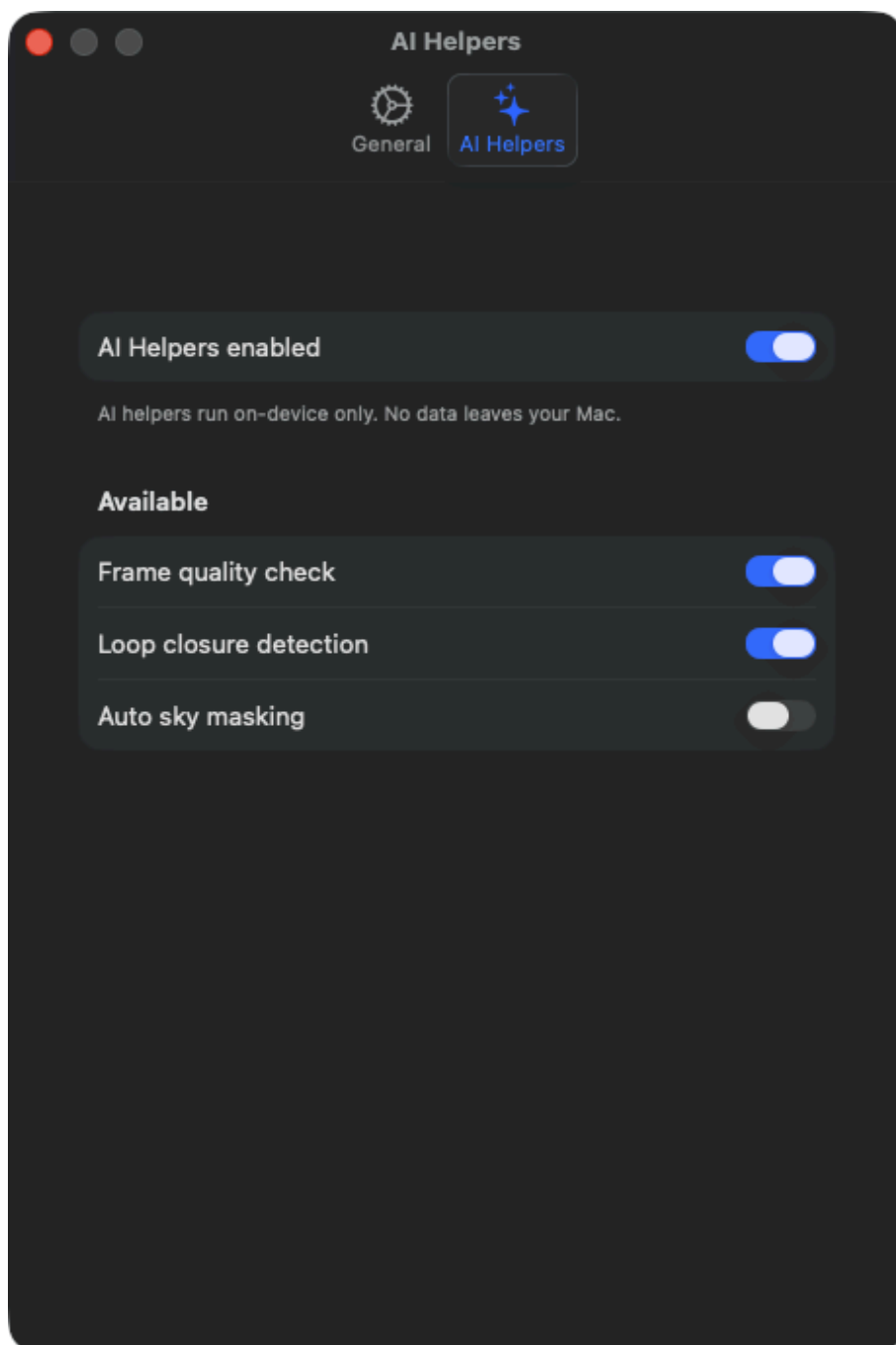


Figura 16: Impostazioni → tab AI Helpers con master switch e sub-toggle

**S11 AI Helpers enabled (Master)**

Settings → AI Helpers → prima sezione → Toggle «AI Helpers enabled». Bound:.. Default: `true`.



Master switch su tutte le funzionalità AI Helpers nella pipeline. Quando spento, la pipeline di importazione e SfM salta completamente tutti gli stadi di preprocessing basati su ML — nessuna chiamata Apple Vision, nessun caricamento di modello CoreML, nessun risveglio della NPU. Quando acceso, vengono consultati i singoli sub-toggle (S12–S13). Il valore viene memorizzato tra i riavvii. Si applica ai seguenti stadi: (a) pre-check di qualità dei frame prima di SfM (S12), (b) rilevamento di loop closure (S13). Importante: quando spento, i due sub-toggle sono disabilitati e visivamente in grigio. La nota a piè di pagina sottolinea che tutti gli AI Helpers vengono eseguiti rigorosamente on-device — nessun upload di immagini, nessuna elaborazione cloud. La garanzia di privacy deriva dall'uso esclusivo del framework Apple Vision (localmente sulla Neural Engine) e dai modelli CoreML che vengono distribuiti direttamente nel bundle dell'app.

**IN PAROLE SEMPLICI**

L'interruttore principale per tutte le funzioni che internamente usano AI/machine learning. Il default è «on», perché gli helper risparmiano molto tempo senza che le tue immagini lascino il Mac. Se li vuoi completamente spenti (ad es. per risparmiare energia o perché il tuo Mac non ha NPU), spegnili qui — le due sub-opzioni qui sotto si grigiano allora automaticamente e non fanno più nulla.

**S12** Frame quality check DOVE

Settings → AI Helpers → sezione Available → Toggle «Frame quality check». Bound:.. Default: `true`.

 TECNICO

Attiva lo screener di qualità dei frame (Fase 3.11), che analizza ogni frame importato prima della chiamata SfM. Passi della pipeline per frame: (a) filtro di varianza laplaciana da Apple Vision (rilevamento sfocatura — soglia ~150), (b) controllo di sovra/sotto-esposizione basato su istogramma (soglia: >5% pixel a 0 o 255), (c) rilevamento frame vuoti (deviazione standard < 5 su tutti i pixel). I frame che superano tutti e tre i controlli passano direttamente. I frame che falliscono almeno un controllo attivano un dialogo modale di conferma che elenca ogni frame problematico con miniatura e motivo e chiede se debba essere rimosso. Importante: nessuna eliminazione automatica — il dialogo è sempre richiesto, l'utente mantiene la decisione finale. Prestazioni: ~50 ms per frame su M3 Ultra, eseguito in parallelo. Quando spento, tutti i frame vengono inoltrati all'SfM senza controllo. Con il master (S11) disabilitato, questo toggle è visivamente in grigio e senza effetto. Stato distribuito secondo la memoria: SHIP-PED 2026-05-23.

 IN PAROLE SEMPLICI

Prima del training vero e proprio, l'app esamina ogni foto: è sfocata? completamente scura o bianca? vuota? In tal caso, ti chiede se vuoi scartare l'immagine — non rimuove mai nulla automaticamente. Questo fa risparmiare molte ore in seguito, perché una singola immagine totalmente sfocata può a volte rovinare l'intero training. Il default è «on», perché il costo è quasi nullo e il beneficio grande.

**S13 Loop closure detection**

DOVE

Settings → AI Helpers → sezione Available → Toggle «Loop closure detection». Bound: Default: `true`.



TECNICO

Attiva il rilevamento di loop closure basato sul feature print di Apple Vision. Per ogni frame importato viene calcolato un vettore di feature di ~768 dimensioni, che rappresenta un embedding neurale del contenuto dell'immagine. Tutti i feature print vengono poi confrontati a coppie tramite similarità coseno. Le coppie con similarità > 0.85 e distanza nell'indice dei frame > 50 (ossia frame non adiacenti) vengono identificate come «candidate di loop closure» e scritte in un file sidecar JSONL nella cartella del progetto. Solo informativo — la sequenza di immagini importate non viene modificata. Scopo: dà al risolutore SfM (in particolare COLMAP) un suggerimento che questi frame si raggruppano insieme nello spazio 3D. Per l'SfM nativo le informazioni sidecar sono attualmente solo documentative; COLMAP utilizza i suggerimenti internamente tramite file di match personalizzato (integrazione manuale possibile, non collegata automaticamente). Prestazioni: ~200 ms per frame su M3 Ultra, eseguito in parallelo. Quando spento, non vengono generati feature print. Con il master (S11) disabilitato, visivamente in grigio.

**IN PAROLE SEMPLICI**

Quando giri attorno a un oggetto mentre scatti e finisci di nuovo al punto di partenza, aiuta enormemente il computer a saperlo. Questa opzione rileva automaticamente quali delle tue foto sono state scattate «quasi dallo stesso punto» e lo scrive in un piccolo file di aiuto. Gli strumenti SfM (specialmente COLMAP) possono usare queste informazioni per fornire una ricostruzione 3D più pulita. Il default è «on», perché viene eseguito senza il tuo intervento e non cambia nulla nelle tue immagini.

## Impostazioni Specchio dell'Inspector

Le restanti voci di Settings (S17–S33) dalla tabella inventario sono specchi dell'Expert Inspector e sono documentate nel Capitolo 2 (controlli Inspector I12–I29). Non appaiono fisicamente nella finestra delle Impostazioni, ma sono state elencate nell'inventario solo perché passano attraverso proprietà di `TrainingConfig` che vengono persistite tramite, e quindi hanno formalmente carattere di impostazioni. Per spiegazioni di contenuto, vedi lì.

## Quando cosa?

Impostazione	Ambito	Persistenza
S1 Default Mode	App-Globale	Riavvio app
S2 Language	App-Globale	Riavvio app
S3 Viewport Background	App-Globale (default) + Runtime	Riavvio app
S4 Auto-Rotate After Training	App-Globale	Riavvio app
S5 Live Preview Interval	Default per nuovi training	Riavvio app
S6 Throttle Delay	Default per nuovi training	Riavvio app
S11 AI Helpers Master	App-Globale	Riavvio app
S12 Frame quality check	App-Globale	Riavvio app
S13 Loop closure detection	App-Globale	Riavvio app

App-Globale = ha effetto su tutti i progetti. Default per nuovi training = ha effetto solo sul prossimo training creato; le sessioni in corso rimangono invariate. Training corrente = ha effetto immediato sulla configurazione del training in corso, ma non persiste senza re-importazione esplicita.

## CAPITOLO

## Capitolo 4 — Finestre ausiliarie

---

Oltre alla finestra principale (viewport 3D più Inspector), RadianceKit gestisce altre sette finestre, tutte aperte tramite il menu Help. La lista da cima a fondo: User Guide (⌘?), Keyboard Shortcuts (⌘/), Open Training Logs... (non apre una finestra dell'app ma il Finder; quindi qui non trattata oltre), Manage Storage..., Pareto Dashboard... (⇧⌘D), Holdout Analysis... (⇧⌘H), BayesOpt Console... (⇧⌘B). Tre di esse — Dashboard, Holdout, BayesOpt — sono strumenti di analisi autonomi. Hanno ciascuno un proprio stack di view model, leggono o scrivono file JSON su disco e per ognuna c'è un argomento CLI con cui far puntare la finestra direttamente a un determinato file all'avvio dell'app ( `--dashboard-dir` , `--holdout-file` , `--bayesopt-autorun` ).

Le quattro finestre semplici (User Guide, Keyboard Shortcuts, Manage Storage, più le voci di sottomenu Open Training Logs / Open Exports Folder) ricevono una breve voce per controllo. Le tre finestre di analisi sono documentate in modo più approfondito — ciascuna con un'introduzione che spiega cosa vedi nella finestra, quando dovresti aprirla e come interpretare l'immagine visualizzata.

Alla fine del capitolo c'è una sezione di rimandi all'Inspector della finestra principale: cosa puoi leggere in modo sensato dal grafico Loss live e dal display Gaussian Count durante un training in corso.

## User Guide (W1–W4)

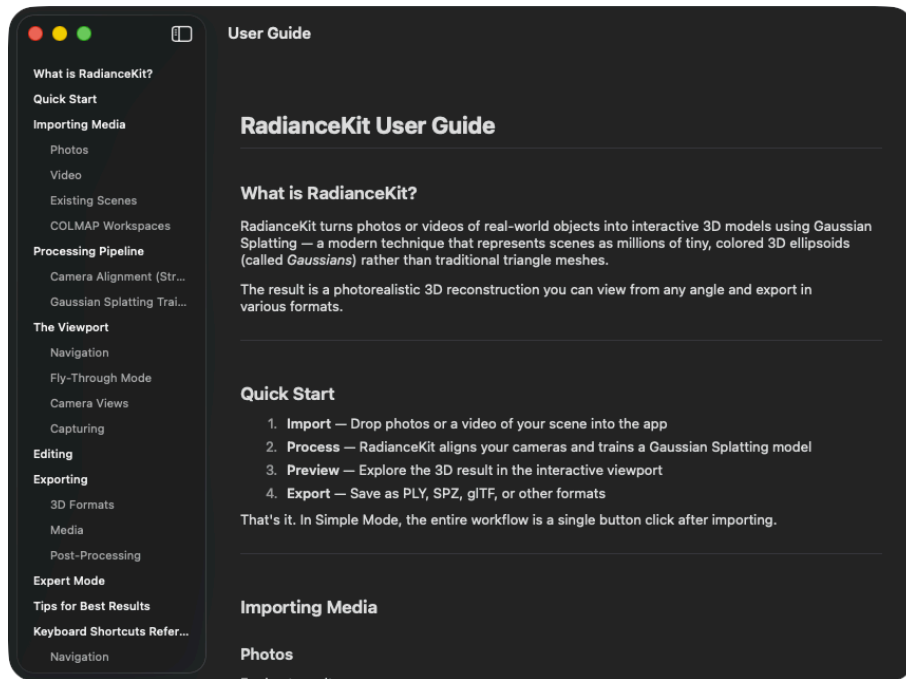


Figura 17: Finestra User Guide con sidebar a sinistra e contenuto Markdown renderizzato a destra

**Cos'è:** Una finestra di aiuto integrata che renderizza la `guide_<lingua>.md` fornita con l'app. La lingua viene derivata dalle Impostazioni (tab General → Language) o, se lì c'è «System», dalle preferenze di lingua di macOS. Il layout è classico: a sinistra la sidebar con tutte le intestazioni, a destra il testo corrente.

**QUANDO APRIRE** Quando hai bisogno di un rapido promemoria su un singolo punto — come sostituto di una parola chiave. Il riferimento esaustivo è questo manuale; la finestra di aiuto integrata è piuttosto ciò che sarebbe un `--help` sulla riga di comando. Viene aggiornata ad ogni release dell'app, ma mantenuta contenutisticamente più superficiale.

## W1 NavigationSplitView (Sidebar + Detail)



DOVE

Help → User Guide (⌘?)..



TECNICO

Layout a due colonne con sidebar stretta (almeno 180 pt di larghezza) per l'albero dei contenuti e un'area di dettaglio scrollabile per il contenuto Markdown vero e proprio. La finestra ha una dimensione minima di 700 × 500 pt. Alla prima apertura la finestra carica la `guide_<lang>.md` corrispondente dal bundle dell'app (fallback `guide_en.md`), la parifica in blocchi (intestazioni H1–H4, paragrafi, liste, tabelle, divisori) ed estrae separatamente la struttura delle intestazioni per la sidebar. La formattazione inline (bold, italic, code-span) viene renderizzata tramite il motore Markdown integrato. La lingua viene letta dalle impostazioni dell'app, con il caso speciale del cinese ( `zh-Hans` ) e del portoghese brasiliano ( `pt-BR` ), che vengono mantenuti come tag locale completi perché queste varianti si distinguono da zh risp. pt.

### IN PAROLE SEMPLICI

Il testo di aiuto integrato, a sinistra la lista dei temi, a destra il contenuto. La lingua si imposta automaticamente in base alle tue impostazioni di sistema. Funziona offline, ma è volutamente solo una versione breve — il riferimento completo è questo manuale.

## W2 List (sidebar delle intestazioni)



DOVE

Colonna sinistra nella finestra User Guide..



TECNICO

Lista su tutte le intestazioni H2 e H3 del documento Markdown attuale. Le voci H2 appaiono senza rientro con peso font medium, le voci H3 con rientro a sinistra di 16 pt e stile foreground ridotto. H4 e profondità superiori vengono ignorate, perché la profondità altrimenti rende la sidebar disordinata. Gli ID di ancoraggio vengono generati dal testo dell'intestazione tramite slugificazione (lowercase + spazi a dash + filtraggio su lettere/numeri/dash — stesso algoritmo che GitHub usa per i suoi ancoraggi Markdown, in modo che anche URL esterni alla documentazione potenzialmente finirebbero sullo stesso ancoraggio). La lista utilizza lo stile macOS nativo.

### IN PAROLE SEMPLICI

La barra di navigazione sul lato sinistro. Tocca una voce e salti alla sezione.

### W3 Button (intestazione → salto all'ancoraggio)



Un pulsante per riga della sidebar..



Ogni voce della sidebar è un pulsante che imposta l'ancoraggio attuale, otticamente però appare come voce di lista. Una variabile osservatore attiva poi il salto di scroll all'ancoraggio corrispondente con un'animazione morbida di 0,3 s. Dopo il salto il valore dell'ancoraggio viene resettato, in modo che il prossimo clic sullo stesso ancoraggio si attivi di nuovo (altrimenti l'osservatore non scatterebbe di nuovo, perché il valore non è cambiato).

#### IN PAROLE SEMPLICI

Il clic ti porta al punto corrispondente nel testo a destra.

### W4 ScrollView (contenuto di dettaglio)



Colonna destra..



Area di contenuto scrollabile e impilata verticalmente con lazy rendering, perché guide più lunghe possono facilmente avere oltre 200 blocchi Markdown — una variante non lazy le istanzierebbe tutte contemporaneamente. Ogni blocco riceve un proprio ID, o l'ancoraggio dell'intestazione (per H1–H3 saltabili) o un placeholder di indice. La larghezza massima è 720 pt, padding 32 orizzontale / 24 verticale, in modo che le righe lunghe mantengano un layout ben leggibile. Le tabelle vengono renderizzate cella per cella con stack orizzontali e divisori; il codice inline tramite il motore Markdown integrato. I veri blocchi di codice vengono attualmente trattati come paragrafo — una limitazione nota della finestra di aiuto.

#### IN PAROLE SEMPLICI

Il vero testo di aiuto. Scrollabile, larghezza ben leggibile, tipografia chiara.

## Keyboard Shortcuts (W5–W6)

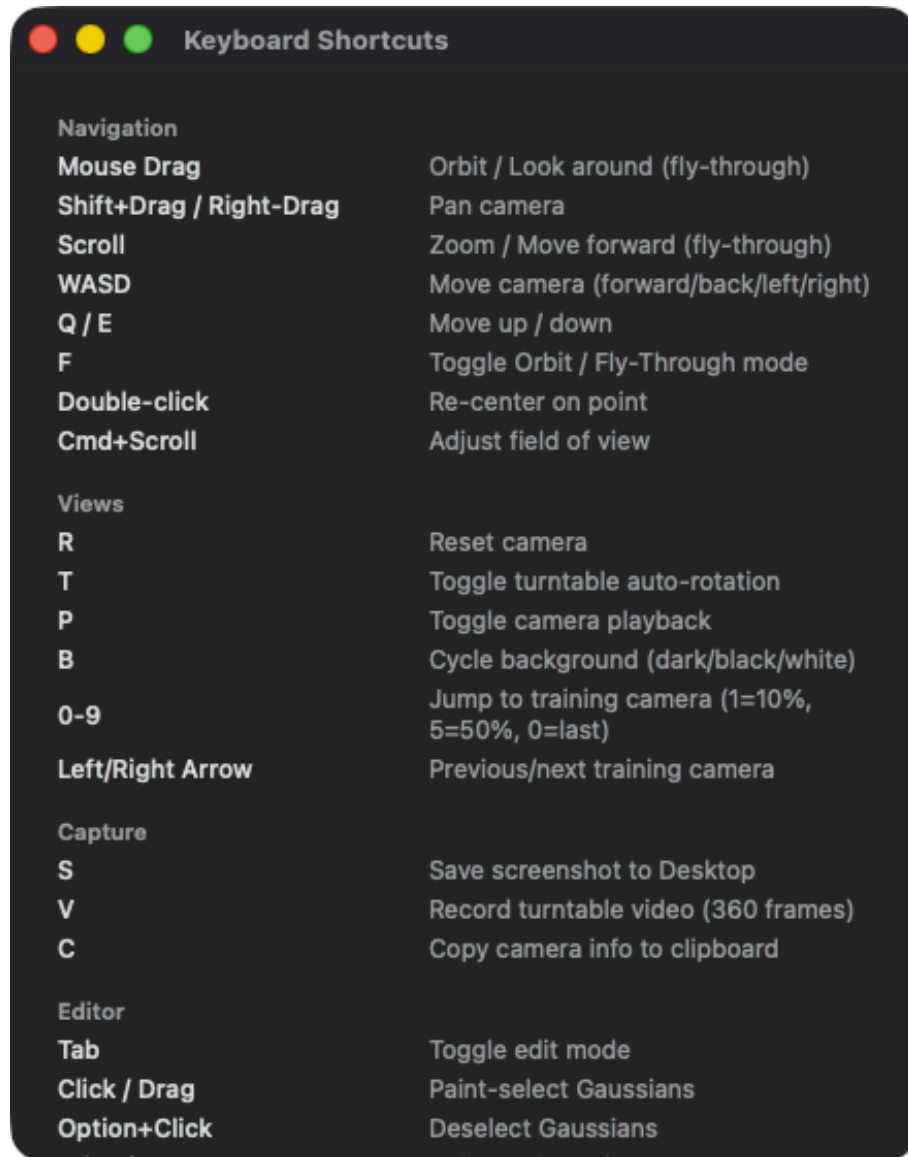


Figura 18: Finestra Keyboard Shortcuts — cinque gruppi Navigation/Views/Capture/Editor/Training con colonna hotkey a sinistra e descrizione a destra

**COSA MOSTRA L'IMMAGINE** Lista di riferimento statica in cinque sezioni. **Navigation:** Mouse Drag (Orbit/Fly), Shift+Drag/Right-Drag (Pan), Scroll (Zoom), WASD (movimento fly-through), Q/E (Up/Down), F (Toggle Orbit/Fly), Double-click (Re-center), Cmd+Scroll (regolazione FoV). **Views:** R (Reset Camera), T (Auto-Rotation), P (Camera Playback), B (ciclo sfondo), 0–9 (salta a training cam 1=10%/5=50%/0=ultima), Left/Right Arrow (Prev/Next Cam). **Capture:** S (Screenshot to Desktop), V (Turntable Video), C (Copy Camera Info). **Editor:** Tab (modalità modifica), Click/Drag (paint-select), Option+Click (deselect), X / Delete (elimina selezione), Cmd-Z (annulla ultima eliminazione), [ / ] (dimensione pennello più piccola/più grande), Esc (deseleziona). **Training:** Start, Pause/Resume, Cancel, Continue +5K/+10K/+20K tramite scorciatoie di menu in M9–M14.

**Cos'è:** Una semplice panoramica statica di tutte le scorciatoie da tastiera — Navigation, Views, Capture, Editor, Training. Il contenuto è hard-coded, nessun caricamento Markdown.

**QUANDO APRIRE** Quando cerchi il modo più rapido per fare qualcosa nel viewport. WASD fly-through, R per reset fotocamera, B per il ciclo dello sfondo — tutti sono qui.

### W5 ScrollView (area di contenuto)

 DOVE

Help → Keyboard Shortcuts (⌘/)..

 TECNICO

Una semplice area di scorrimento con una lista verticale all'interno. Padding 20 tutt'intorno, nessuna sidebar di navigazione (la lista è abbastanza corta). I contenuti sono raggruppati in cinque sezioni (Navigation, Views, Capture, Editor, Training). Per combinazione di tasti una riga con testo traducibile in entrambe le colonne. Colonna sinistra (codice tasto) fissata a 180 pt di larghezza, in modo che le descrizioni a destra rimangano allineate verticalmente. Nessuna interazione oltre allo scorrimento — cliccare su una riga non attiva nulla, le scorciatoie sono veri modificatori di tastiera nel menu e nel viewport.

#### IN PAROLE SEMPLICI

Tabella di tutti i tasti di scorciatoia. Cheat sheet statico per una rapida consultazione.

### W6 VStack (sezioni delle scorciatoie)

 DOVE

All'interno dello ScrollView..

 TECNICO

Sezioni impilate allineate a sinistra con distanza di 16 pt. All'interno delle cinque sezioni rispettivamente intestazione + sequenza di righe. Le intestazioni usano uno stile subheadline secondario — deliberatamente nessun formato title, perché le sezioni non devono essere navigabili. Il contenuto è deliberatamente piatto (nessuna disclosure, nessuna ricerca, nessun filtro), affinché il componente funzioni invariato su ogni versione macOS e il file rimanga leggibile.

#### IN PAROLE SEMPLICI

Il raggruppamento dei tasti per funzione (Navigation, Views, Editor e così via).

## Manage Storage (W7–W12)

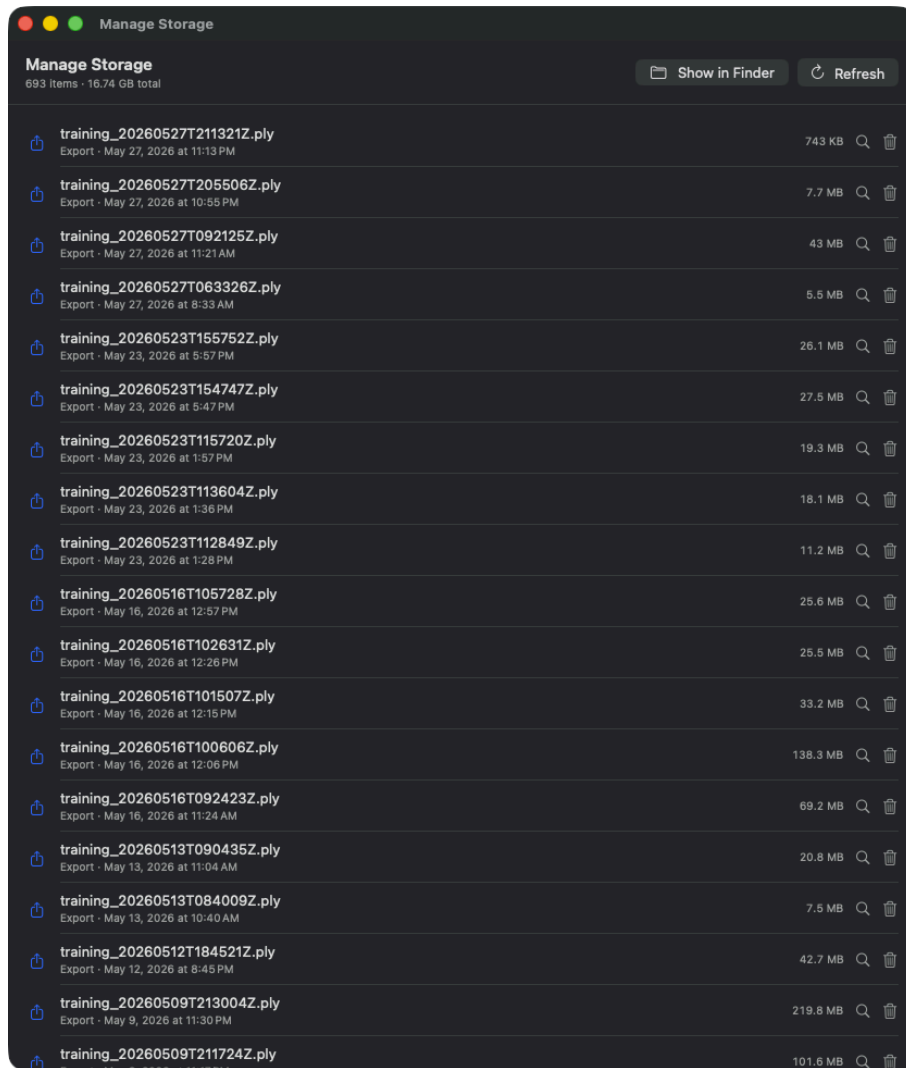


Figura 19: Finestra Manage Storage — header mostra «693 items · 16.74 GB total», tabella con file PLY esportati ordinati per data, ciascuno con pillola formato + nome file + dimensione + data

**COSA MOSTRA L'IMMAGINE** Vista a tabella di tutti i file gestiti da RadianceKit. L'header conta 693 voci, dimensione totale 16.74 GB. Toolbar in alto: «Show in Finder» + «Refresh». Ogni riga: icona PLY, nome file (ad es. `training_20260527T211321Z.ply`), data esportazione, dimensione (varia da 7 KB a 218 MB), icona lente (Reveal) e icona cestino (Move to Trash). I file sono ordinati per data, più recenti in alto. In questa registrazione demo dominano le esportazioni PLY, perché si è lavorato molto con `--benchmark`.

**Cos'è:** Una panoramica dell'uso del disco per tutto ciò che RadianceKit deposita sotto `~/Documents/RadianceKit/` — log, esportazioni, scene, bundle di capture (dal companion iOS), importazioni (copie di staging delle immagini di input). Per voce una dimensione in byte e due pulsanti: «Mostra nel Finder» e «Sposta nel Cestino». NON è una pulizia automatica — l'app non elimina nulla da sola; decidi tu per voce.

**QUANDO APRIRE** Quando il disco si riempie. Soprattutto i log si accumulano (un JSONL per tentativo di training, più il `_qualityMetrics.json`); le esportazioni naturalmente

anche (PLY 100% dati grezzi, una per esportazione). Utile anche dopo un crash, quando la directory di staging Imports ha ancora vecchie copie delle immagini di input (vedi «Disk-pressure incident» in `dev_v549f-needle-reduction.md`).

### W7 Pulsante «Show in Finder»

 DOVE

Header in alto a destra nella finestra Storage Browser..

 TECNICO

Apri l'intera directory RadianceKit (`~/Documents/RadianceKit/`) nel Finder, in modo che tu possa vedere direttamente la struttura delle cartelle e manipolarla anche con il Finder stesso. L'azione apre una nuova finestra del Finder e non passa al container sandbox dell'app — `~/Documents/RadianceKit/` è il dominio Documents regolarmente accessibile alle app, non un percorso di container sandboxed.

#### IN PAROLE SEMPLICI

Apri la directory nel Finder, in modo che tu possa maneggiare i file da solo.

### W8 Pulsante «Refresh»

 DOVE

Header, accanto al pulsante Finder..

 TECNICO

Attiva una scansione in background, che gira su un task asincrono iniziato dall'utente, in modo che la scansione di grandi alberi di directory non blocchi l'UI. Il walk vero e proprio attraversa ogni sottocartella nota (Logs, Exports, Scenes, Captures, Imports) e crea una voce di archiviazione per ogni figlio diretto. Per voce viene determinata la dimensione ricorsiva — preferibilmente il consumo reale del disco (incluso APFS hardlinks sharing) con fallback sulla dimensione logica del file.

#### IN PAROLE SEMPLICI

Legge di nuovo la lista, nel caso tu abbia eliminato o aggiunto qualcosa nel Finder nel frattempo.

**W9 List (voci di archiviazione)**

DOVE

Contenuto principale sotto l'header..



TECNICO

Lista con per riga questo layout: icona SF Symbol specifica per categoria (documento per Logs, freccia upload per Exports, cubo per Scenes, vassoio per Imports), nome + sottotitolo (etichetta tipo + data di modifica formattata), contatore byte a destra (allineato a destra, monospaziato), pulsante Reveal (simbolo lente), pulsante Trash (cestino). Ordinamento: primario per tipo (Scenes prima, poi Exports, Logs, Captures, Imports, Other), secondario per data di modifica decrescente (più recenti in alto). Se la scansione è ancora in corso, in quella posizione viene mostrato invece un progresso «Scanning...». Se non è stato trovato nulla, una visualizzazione empty state con icona vassoio.

**IN PAROLE SEMPLICI**

Lista di tutti i tuoi dati Radiance-Kit, ordinati per tipo e attualità. Per voce vedi la dimensione e puoi eliminare direttamente.

**W10 Pulsante riga «Reveal in Finder»**

DOVE

Per riga, simbolo lente a destra..



TECNICO

Apri il Finder e seleziona la voce specifica (file o cartella). Differenza rispetto a W7: W7 apre la directory root; W10 contrassegna esattamente questa singola voce. Flusso di lavoro pratico: identifica una voce grande, clicca sulla lente, poi copiala ad esempio su un volume esterno.

**IN PAROLE SEMPLICI**

Salta nel Finder direttamente su questa voce, in modo che tu la trovi rapidamente.

### W11 Pulsante riga «Move to Trash»



Per riga, simbolo cestino a destra accanto alla lente..



Attiva la finestra di conferma (W12). Solo dopo conferma viene eseguita l'operazione standard di macOS «sposta nel cestino» (quindi reversibile, nessuna eliminazione diretta). Dopo il successo del trash la voce viene rimossa dalla lista e il contatore byte totale aggiornato. In caso di errori viene visualizzata una finestra di errore modale.

#### IN PAROLE SEMPLICI

Sposta la voce nel cestino. La finestra chiede prima.

### W12 ConfirmationDialog (conferma di eliminazione)



Viene attivata da W11, visualizzata come sheet macOS..



Finestra di conferma standard con titolo dinamico «Delete <name>?» e una riga di messaggio che indica esplicitamente che la voce finisce nel cestino ed è da lì ripristinabile (finché il cestino non viene svuotato). Due pulsanti: «Move to Trash» come azione distruttiva (rappresentata in rosso) e «Cancel» con binding automatico di Esc. La finestra è non modale nel senso che blocca solo questa finestra, non l'intera app — è standard macOS per eliminazioni reversibili.

#### IN PAROLE SEMPLICI

Domanda di sicurezza prima dell'eliminazione. «Move to Trash» è reversibile — finché il cestino non viene svuotato.

## Pareto Dashboard (W13–W22)

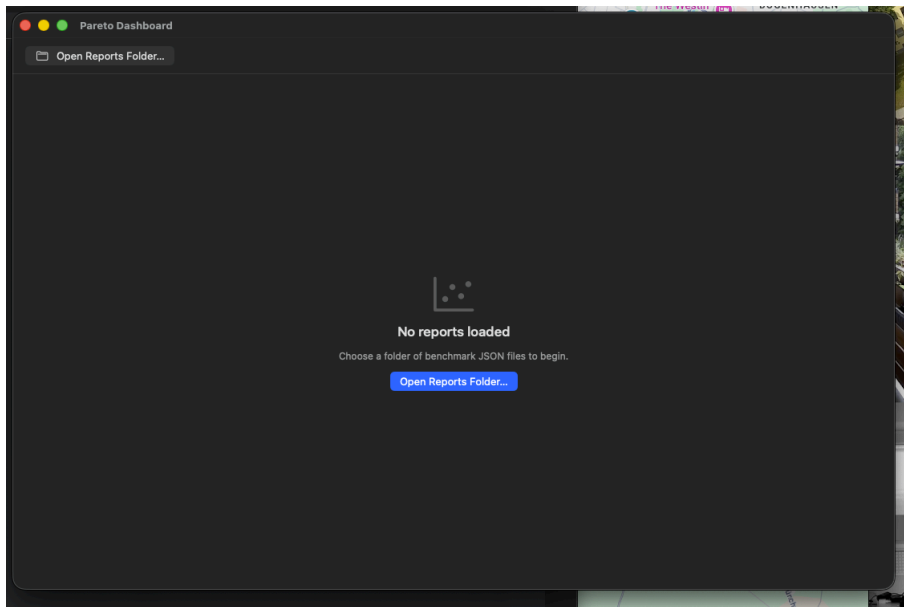


Figura 20: Pareto Dashboard — stato vuoto prima dell'importazione del report

Stato vuoto (dopo la prima apertura) — empty state con call-to-action «Open Reports Folder...». I punti dati appaiono non appena i report di training sono caricati, vedi il prossimo scatto.

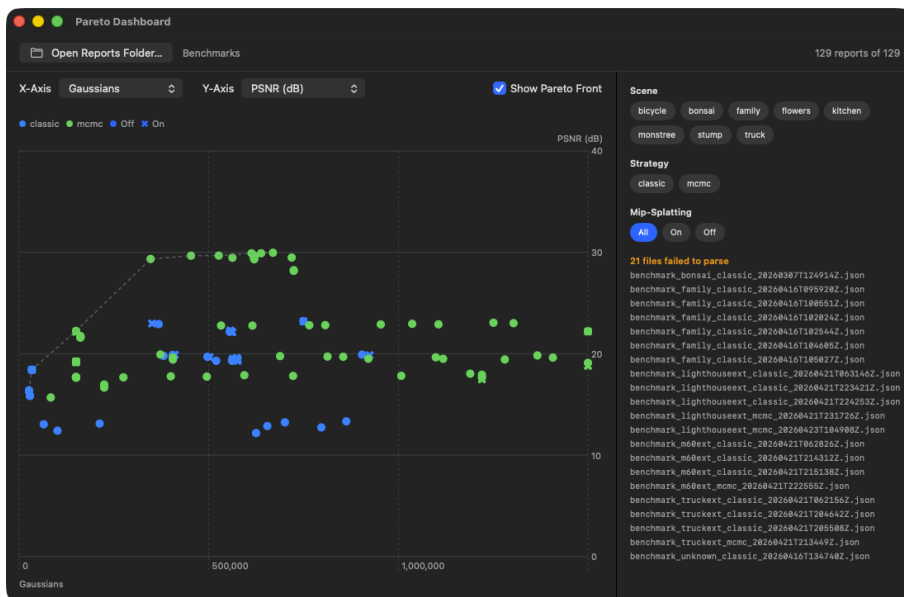


Figura 21: Pareto Dashboard con 129 report di benchmark caricati — Gaussiane vs PSNR con frontiera di Pareto, filtri Scene/Strategy/Mip

**COSA MOSTRA L'IMMAGINE** La toolbar dell'header mostra «129 reports of 129» (tutti i report nella cartella scelta sono stati parsati con successo — 21 file aggiuntivi non sono stati parsati per formato più vecchio, vedi lista di note a destra). Assi: picker X su Gaussians, picker Y su PSNR (dB). Scatter plot: punti verdi = strategia Classic, punti blu = MCMC. La linea tratteggiata della frontiera di Pareto corre lungo i valori PSNR


migliori raggiunti e si appiattisce attorno a  $\text{PSNR} \approx 30$  dB da circa 500K gaussiane. Chip filtro a destra: 7 scene (bicycle, bonsai, family, flowers, kitchen, stump, truck), 2 strategie (classic, mcmc), 3 opzioni Mip-Splatting (All, On, Off). Attualmente tutti i filtri sono aperti, da qui il denso cluster di punti.

**Cos'è:** Uno strumento di confronto multi-run. Hai addestrato in passato più scene o la stessa scena con preset diversi — ognuna di queste esecuzioni di training produce (se hai passato `--benchmark` o chiamato tramite la funzione benchmark) un file di report JSON che contiene tra l'altro PSNR, SSIM, LPIPS, gaussian count e tempo wallclock finali. Il dashboard legge un'intera cartella di tali report contemporaneamente e li traccia come scatter 2D con assi selezionabili. Inoltre la frontiera di Pareto (l'insieme dei punti non dominati) viene tracciata come linea tratteggiata.

**QUANDO APRIRE** Dopo aver creato almeno tre o quattro report di training. Con meno punti la linea della frontiera non è significativa. Caso d'uso tipico: hai provato a ricostruire una scena outdoor e hai eseguito in sequenza P3 Balanced (Classic), P4 Quality (Classic), P7 MCMC Quality e P9 Outdoor (tuned) — ora vuoi sapere quale configurazione fornisce il miglior PSNR per secondo di tempo di training o quale ha bisogno di meno gaussiane per un dato PSNR.

**COME INTERPRETARE** Entrambi gli assi sono liberamente selezionabili (asse X:,, `psnr`, `ssim`, `lpips`, ...; asse Y allo stesso modo). La logica della frontiera di Pareto in `ParetoFront2D.indices` sa per ogni metrica se «più piccolo = meglio» (ad es. LPIPS, Loss, Time) o «più grande = meglio» (PSNR, SSIM) — la linea corre quindi a seconda della scelta degli assi da basso sinistra ad alto destra o da alto sinistra a basso destra, sempre lungo la migliore combinazione raggiunta. Un punto è ottimale di Pareto se NESSUN altro punto è almeno altrettanto buono in ENTRAMBE le dimensioni (cioè nessun altro lo domina). I punti ottimali di Pareto si trovano sulla linea, gli altri punti a destra/sopra (a seconda dell'orientamento degli assi). I punti SULLA linea sono i veri candidati per «miglior preset»; i punti LONTANI dalla linea sono tempo di training sprecato.

**CHIP FILTRO** Puoi restringere la selezione a una scena specifica (se vuoi confrontare ad es. solo le esecuzioni outdoor), a una strategia specifica (Classic o MCMC) o a Mip-Splatting on/off (rilevante dopo la Fase Q1.5, dove Mip rimane come opt-in advanced flag).

 Hai tre report per la scena «truck» sotto `~/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). Imposta asse X su `trainingTime`, asse Y su `PSNR`. Run B sta in alto a destra, Run C ancora più in alto a destra, Run A in basso a sinistra. La frontiera di Pareto collega A e C — entrambi non dominati. Run B è «perso» (C è migliore in Time E PSNR). Conclusione: per «truck» non vale la pena il default MCMC; o veloce+ok (A) o lungo+molto buono (C). Salva la configurazione di C come preset proprio (Inspector → I1 Save Preset).

**Prossima azione:** Salvare la migliore configurazione come preset. In concreto: guarda i punti di Pareto (l'hover mostra PSNR/SSIM/LPIPS/Gs/Time nel tooltip), decidi quale si adatta meglio al trade-off tempo-vs-qualità, apri il report corrispondente (il nome file

contiene il timestamp del run), copia la sua configurazione di training in un nuovo run o salvala come preset dopo la prossima sessione di training tramite l'Inspector.

### W13 Pulsante «Open Reports Folder...»



Toolbar in alto a sinistra..



Apri una finestra di selezione cartella con la richiesta «Select a folder containing benchmark .json reports». Dopo la conferma viene eseguito un task in background che parsifica sequenzialmente tutti i file `.json` nella cartella. I report errati (JSON danneggiato, schema sbagliato) vengono raccolti e visualizzati in fondo alla sidebar come «N file failed to parse» — nessun crash. Se avviene un secondo clic mentre un primo load è ancora in corso, il task precedente viene interrotto, in modo che due risultati non scrivano contemporaneamente nello stato.

Anche via CLI: `--dashboard-dir /path/to/reports` carica la cartella direttamente all'avvio dell'app.

#### IN PAROLE SEMPLICI

Seleziona la cartella in cui si trovano i tuoi report di benchmark. Il percorso standard è `~/Documents/RadianceKit/Reports/`. Carica poi tutti i JSON in una volta.

### W14 Picker «X-Axis»



Sopra il grafico, a sinistra..



Picker di menu con tutti gli assi metrici disponibili del modulo dashboard (PSNR, SSIM, LPIPS, gaussian count, tempo di training e così via). Il default è gaussian count. Al cambio il punto in hover viene resettato, perché una posizione finora evidenziata nel vecchio sistema di coordinate degli assi non ha più senso dopo il cambio di asse. Il picker è limitato alla larghezza del contenuto, in modo che non si estenda su tutta la larghezza.

#### IN PAROLE SEMPLICI

Quale metrica deve stare sull'asse orizzontale. Di solito «tempo di training» o «numero di gaussiane», perché sono i «costi» che vuoi confrontare.

**W15 Picker «Y-Axis»**

Sopra il grafico, accanto a X-Axis..



Identico a W14, solo che il default è PSNR. La scelta dell'asse viene salvata in modo indipendente, quindi l'utente può anche scegliere combinazioni senza senso (X=PSNR, Y=PSNR — porterebbe tutti i punti su una diagonale). Tali combinazioni non vengono però intercettate; decisione consapevole, perché un confronto «SSIM vs PSNR» è ben interessante per vedere quanto coerentemente si comportano le metriche.

**IN PAROLE SEMPLICI**

Cosa sta sull'asse verticale. Normalmente «PSNR» o «SSIM» come metrica di qualità.

**W16 Toggle «Show Pareto Front»**

A destra accanto ai picker degli assi..



Toggle macOS standard. Se attivo, nel grafico Pareto viene tracciata, oltre alla nuvola di punti, una linea con la frontiera di Pareto 2D calcolata. Stile: tratteggiata (pattern tratti 4-4), grigia semitrasparente, spessore linea 1,5 pt. Il calcolo Pareto avviene nel main thread — con il numero tipico di report ( $\leq \sim 50$ ) è senza problemi veloce. Se il toggle è spento, la linea viene omessa, in modo che ci siano solo i punti nudi.

**IN PAROLE SEMPLICI**

Mostra la linea che passa attraverso i punti «finora migliori». Se la linea ti dà fastidio (ad es. perché vuoi confrontare solo i singoli trade), spegnila.

**W17** Chip filtro «Scene»

Sidebar destra nella finestra del dashboard..



Chip filtro per ogni scena presente nei report caricati. Layout flow proprio che riimpacchetta automaticamente i chip su più righe non appena la larghezza è esaurita. I chip attivi ricevono lo sfondo di accento, quelli inattivi uno sfondo materiale standard neutro. La selezione multipla è possibile (semantica di set); se nessun chip è selezionato, valgono tutte le scene come «fatte passare» — ovvero la logica di set è «selezione vuota = tutto», non «selezione vuota = nulla».

 IN PAROLE SEMPLICI

Il clic su un nome di scena filtra i punti solo su questa scena. Selezione multipla possibile. Vuoto = tutte le scene.

**W18** Chip filtro «Strategy»

Sotto il filtro Scene nella sidebar..



Esattamente come W17, ma per le strategie di training — tipicamente i due valori «classic» e «mcmc», derivati dal campo strategy dei JSON dei report di benchmark. Utile se hai mescolato report di entrambe le strategie e vuoi vedere solo una sorta (ad es. «mostra solo run MCMC, perché Classic l'ho già escluso»).

 IN PAROLE SEMPLICI

Filtro per Classic o MCMC. Per default entrambi sono attivi.

**W19 Chip filtro «Mip-Splatting»**

Sotto il filtro Strategy nella sidebar..



Filtro a tre valori (invece di set come W17/W18): «All» / «On» / «Off». Sfondo: Mip-Splatting è stato valutato nella Fase Q1.5 come miglioramento multi-scala sperimentale e il verdetto finale è stato «nessun vantaggio piacevole in modo continuativo; mantenuto come opt-in flag». Quando fai confronti Mip on/off, spesso vuoi poter separare nettamente. Da qui il filtro ternario dedicato con gli stati «lascia passare tutto», «solo Mip on», «solo Mip off». La sezione della sidebar viene mostrata solo se ci sono almeno un report Mip E almeno un report non Mip nel set di dati (altrimenti il filtraggio non ha senso).

 IN PAROLE SEMPLICI

Se vuoi confrontare Mip-Splatting on/off, qui filtro a tre parti. Altrimenti ignora.

**W20 ChipButton (toggle filtro, all/on/off)**

Componente helper, viene usato in W17/W18/W19..



Wrapper minimalista per pulsante. Contenuto: testo etichetta con grado font caption e padding 10 orizzontale / 5 verticale. Sfondo condizionale: se attivo → colore di accento dell'app con testo bianco; altrimenti sfondo materiale standard neutro con testo nero. La forma è una capsula (a forma di pillola). Stile pulsante plain, in modo che il materiale capsula non sia sovrapposto a un bordo di sistema.

 IN PAROLE SEMPLICI

I pulsanti rotondi del filtro stessi. Otticamente come un tag iOS.

**W21 Chart (scatter Pareto)**

Area centrale del dashboard..



Grafico Swift Charts con due layer: 1. un punto per report — posizione dalle metriche X e Y scelte, colore per strategia, simbolo per stato Mip. Dimensione simbolo normale 80, evidenziato 200 (se l'ID corrisponde al report attualmente in hover). 2. una linea per la frontiera di Pareto, solo se il toggle è attivo.

Overlay del grafico: un rettangolo trasparente registra il movimento del mouse; per frame viene determinata la posizione del punto più vicina euclideanamente nel frame del grafico e il report in hover aggiornato, se la distanza è sotto 24 px (altrimenti resettato). Così ottieni il tooltip senza cliccare — basta l'hover.

**IN PAROLE SEMPLICI**

Il vero diagramma di dispersione. Ogni punto è un'esecuzione di training. Hover per il tooltip di dettaglio.

**W22 Tooltip (dettaglio hover)**

Sotto il grafico, mostrato in hover..



Stack orizzontale: nome scena (headline), tag strategia (caption), divisore, poi metriche PSNR/SSIM/LPIPS/Gs/Time ciascuna in un piccolo gruppo verticale (etichetta + valore monospaziato). Se Mip era attivato, in più un tag capsule «Mip» in colore di accento. Sfondo blur semitrasparente, rettangolo arrotondato con raggio 8 pt. Viene mostrato solo se il mouse è effettivamente su un punto. Scompare automaticamente all'uscita.

**IN PAROLE SEMPLICI**

La scheda di dettaglio in basso, quando passi con il mouse su un punto. Mostra tutte le metriche di qualità e la configurazione del run in una volta.

## Holdout Analysis (W23–W29)

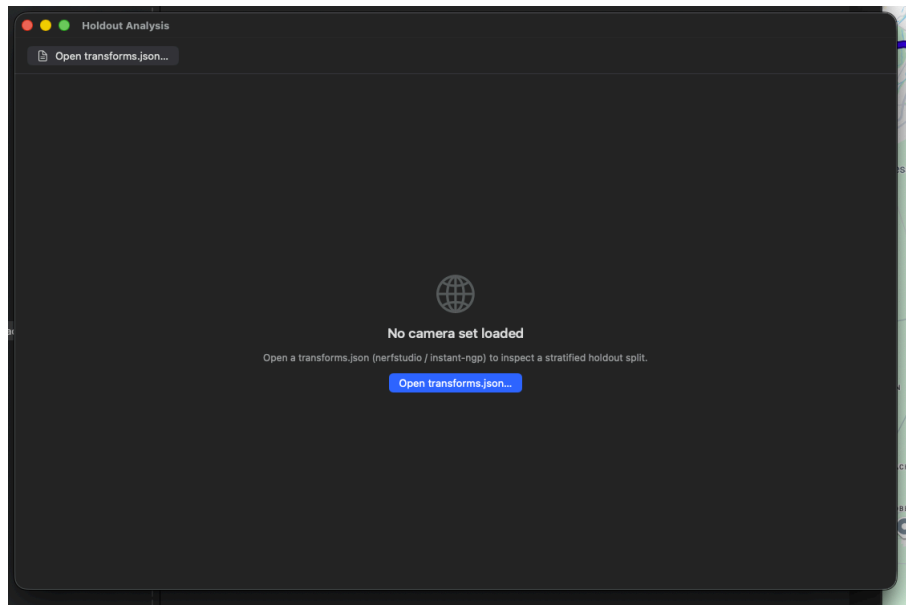


Figura 22: Holdout Analysis — stato vuoto prima del caricamento di un transforms.json

Stato vuoto con empty state e call-to-action «Open transforms.json...». Accetta i formati NeRF Studio e Instant NGP.

Stato vuoto (dopo la prima apertura) — i marker delle fotocamere appaiono non appena un `transforms.json` è caricato, vedi il prossimo scatto.

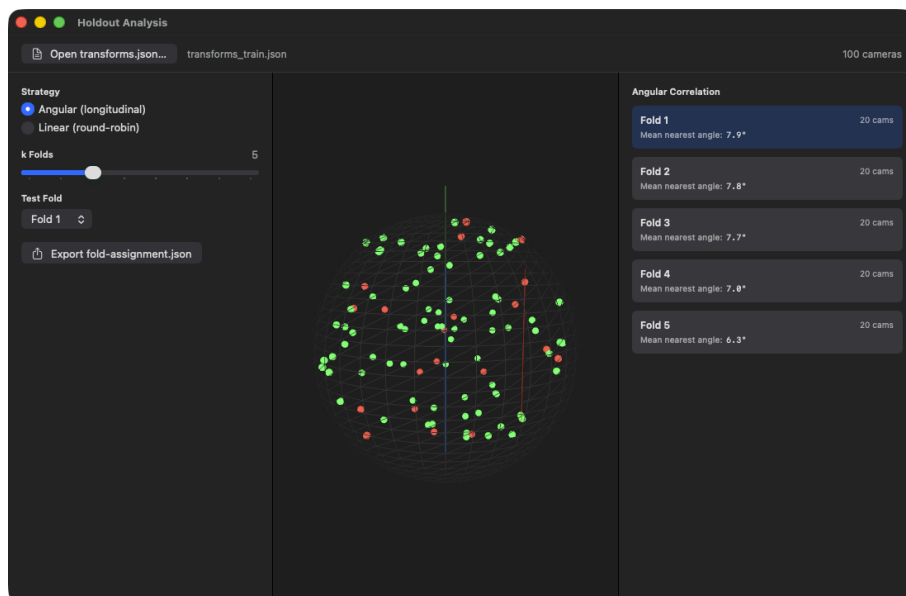


Figura 23: Globo Holdout con 100 fotocamere NeRF Blender Mic, 5 fold da 20 fotocamere ciascuno, strategia angolare attiva

**COSA MOSTRA L'IMMAGINE** L'header mostra il file caricato (`transforms_train.json`) e il conteggio cam («100 cameras»). Sidebar sinistra: picker della strategia con due opzioni — Angular (longitudinal) attivo (allinea i fold per settori di longitudine/latitudine sulla sfera, in modo che ogni test fold sia geometricamente denso) vs Linear (round-robin)


(basato sull'ordine, ogni k-esimo frame come test set). Lo slider k-Folds è su 5, il picker Test Fold su Fold 1. Il pulsante Export genera un `fold-assignment.json` per Nerfstudio/Instant-NGP. Pannello centrale: proiezione su globo 3D di tutte le 100 fotocamere — punti verdi = Train, punti rossi = test fold attuale (Fold 1 con 20 fotocamere). Sidebar destra (Angular Correlation): per fold 20 cam + Mean Nearest Angle (Fold 1: 7.9°, Fold 2: 7.8°, Fold 3: 7.7°, Fold 4: 7.0°, Fold 5: 6.3°) — un valore più piccolo significa che le fotocamere all'interno di questo fold sono strettamente vicine, quindi lo split holdout è spazialmente coerente.

**Cos'è:** Un visualizzatore 3D per la disposizione delle tue fotocamere con logica di cross-validation. Carichi un `transforms.json` (il formato standard di Nerfstudio/Instant-NGP per le pose delle fotocamere), l'app legge tutte le fotocamere, proietta le loro direzioni di vista su una sfera unitaria e le mostra come piccoli marker sferici su un globo virtuale. Poi divide le fotocamere in `k` fold (secondo la strategia scelta: angolare o lineare), contrassegna in verde la quota di training e in rosso la quota di test (holdout), e calcola per fold un punteggio Angular Correlation, che ti dice quanto lontano il test fold è dal training fold nello spazio degli angoli di visualizzazione.

**QUANDO APRIRE** Quando vuoi fare valutazione holdout — ovvero: quanto bene generalizza il tuo modello su angoli di visualizzazione mai visti? Lo standard nel training è «every-8th view come holdout» (convenzione Mip-NeRF360), ma è una divisione molto lineare. Se le tue immagini sono ad esempio raggruppate temporalmente (prima un lato dell'oggetto, poi l'altro), allora «every-8th» non è rappresentativo — una posizione casuale della sequenza finisce nel test, ma tutti i suoi vicini sono nel training, è troppo facile. Con «angular» si stratifica invece sullo spazio degli angoli di visualizzazione: ogni fold contiene fotocamere da tutte le aree dell'orbit, in modo che il test testi davvero le lacune di generalizzazione.

**COME INTERPRETARE** Angular vs Linear: - Angular (standard): divide le fotocamere per angolo longitudinale (coordinata  $\phi$  attorno all'asse Y) in `k` settori uguali. Fold 0 sono fotocamere con  $\phi \in [0^\circ, 360/k^\circ)$ , Fold 1 i successivi, e così via. Vantaggio: ogni fold copre una sezione dell'orbit; il test fold è spazialmente compatto ma ampiamente distribuito sul dataset mondiale. Buono per riprese orbit classiche. - Linear (round-robin): indice di fold =  $(\text{image\_index} \bmod k)$ . È la semplice divisione «every-k-th». Funziona se l'ordine dell'immagine NON ha bias spaziale (ad es. riprese di drone ordinate casualmente). Funziona male se le immagini si raggruppano temporalmente.

Nel globo 3D vedi subito: punti verdi (training) e punti rossi (test). Se i punti rossi si raggruppano tutti in un angolo, l'holdout è cattivo (nessun buon test di generalizzazione). Se sono distribuiti uniformemente tra i verdi, è buono. Il punteggio Angular Correlation per fold (sidebar destra, in gradi) dice in più: valore più piccolo = il test è vicino al training (ogni fotocamera di test ha una fotocamera di training vicina, test facile); valore più grande = il test è lontano dal training (generalizzazione più dura).

 Hai registrato la tua scena Truck con 251 immagini, esporti tramite la voce di menu M33 (Export SfM transforms.json) un file nerfstudio. Apri la finestra Holdout (🔗🔍), carica il JSON tramite «Open transforms.json...», guarda il globo. `k=5` (default) ti dà 5 fold. Clicca su «Fold 3» — vedi se i marker rossi sono ragionevolmente uniformi. Se sì: «Export fold-assignment.json», metti il file esportato nella cartella Reports, e alla prossima esecuzione di training con `--benchmark` (o le corrispondenti impostazioni

dell'Inspector) viene usata esattamente questa divisione di fold come holdout di test — al posto dello standard «every-8th».

### W23 Pulsante «Open transforms.json...»



DOVE

Toolbar in alto a sinistra..



TECNICO

Apri una finestra di selezione file limitata ai file JSON. Dopo la conferma il modulo Holdout carica il file. Il loader parsifica sia il formato nerfstudio (intrinseci della fotocamera più lista di frame con percorso immagine e matrice trasformazione) sia il formato instant-ngp (stessa struttura). Per frame, la direzione di vista viene estratta dalla matrice trasformazione (asse z della base locale della fotocamera) e salvata. Se il parsing fallisce, viene mostrato un messaggio di errore nell'area di stato.

Anche via CLI: `--holdout-file /percorso/a/transforms.json` avvia la finestra direttamente con il file caricato.

#### IN PAROLE SEMPLICI

Carica il tuo JSON delle pose della fotocamera. Lo standard sono le esportazioni Nerfstudio e Instant-NGP. RadianceKit stesso può esportare transforms.json tramite menu → Export → SfM.

### W24 Picker «Strategy» (angular/linear)



DOVE

Sidebar sinistra, in alto..



TECNICO

Picker radio con due opzioni: Angular e Linear. Il cambio di strategia attiva automaticamente un ricalcolo dei fold. Le direzioni di vista sono una lista di vettori unitari 3D sulla sfera; la strategia angolare li proietta sull'angolo longitudinale  $\phi$  e ordina, la strategia lineare fa semplicemente una divisione modulo sull'indice del frame.

#### IN PAROLE SEMPLICI

Angular per riprese orbit uniformi (standard, sicuro), Linear solo se le tue immagini non si raggruppano spazialmente.

**W25 Slider «k Folds»**

Sidebar sinistra, al centro..



Slider da 3 a 10, passo 1. Alla modifica il calcolo dei fold viene riavviato automaticamente, in modo che la lista dei fold, gli indici di training/test e il punteggio per fold vengano ricalcolati immediatamente. Il valore scelto viene visualizzato come testo monospaziato a destra accanto all'etichetta.

Regola pratica:  $k=5$  è standard (ti dà il 20% di test per fold, è comune per cross-validation).  $k=10$  quando hai molti dati e hai bisogno di più fold per significatività statistica.  $k=3$  quando hai pochi dati.

**IN PAROLE SEMPLICI**

Quanti fold nella divisione. 5 è standard e quasi sempre va bene.

**W26 Picker «Test Fold»**

Sidebar sinistra, sotto lo slider k..



Picker di menu. Le opzioni sono dinamicamente  $0..<k$ , etichetta «Fold 1» a «Fold N» (quindi 1-indexed nell'UI, 0-indexed internamente). Se l'indice scelto in precedenza è  $\geq k$  (ad es. perché hai ridotto  $k$  da 10 a 5), viene resettato automaticamente a 0. Il test fold scelto viene rappresentato in rosso nel globo, tutti gli altri in verde.

**IN PAROLE SEMPLICI**

Quale fold è attualmente il test fold. Puoi cliccare attraverso e vedere come appare ogni singolo fold nel globo.

**W27 Pulsante «Export fold-assignment.json»**

Sidebar sinistra, in basso..



Apri una finestra di salvataggio con nome file predefinito `fold-assignment.json`. Dopo la conferma il modulo Holdout codifica la divisione attuale in uno schema JSON (assegnazione fold per frame più meta blocco strategia). Questo file può poi essere passato al prossimo training con `--benchmark`, in modo che lo stesso holdout venga usato per la valutazione finale delle metriche. Gli errori di scrittura vengono mostrati come testo di errore; il successo in testo verde come «Saved to (filename)».

**IN PAROLE SEMPLICI**

Salva la divisione train/test attuale come JSON. Questo file puoi passarlo direttamente al training, in modo che venga usato di nuovo lo stesso set di test.

**W28 SCNView (globo 3D delle fotocamere)**

Pannello centrale nella finestra Holdout..



Vista globo SceneKit. La scena è composta da: una sfera wireframe (raggio 1.0, 36 segmenti, grigio scuro), tre moncherini di assi colorati (rosso/verde/blu per X/Y/Z, ciascuno lunghi 1.2), e per fotocamera una piccola sfera marker (raggio 0.03) nella posizione di direzione di vista corrispondente sulla sfera unitaria (leggermente all'esterno, in modo che non scompaia NELLA sfera wireframe). I marker NON vengono ricostruiti ad ogni cambio di fold — il rebuild è necessario solo se cambia la lista dei frame (cioè quando viene caricato un nuovo JSON). Invece per aggiornamento avviene un aggiornamento in place dei colori del materiale: rosso per indici di test, verde per training, grigio chiaro se né né. Così i tick dello slider rimangono performanti anche con  $N > 1000$  fotocamere.

Il controllo della fotocamera è attivato — puoi ruotare con il mouse il globo, zoomare, panare. L'illuminazione assicura che i marker non appaiano piatti. Lo sfondo è grigio scuro.

**IN PAROLE SEMPLICI**

Il globo 3D con le posizioni delle fotocamere. Verde = training, rosso = test, grigio chiaro = non assegnato (non appare, tutte le fotocamere appartengono da qualche parte). Con il mouse puoi ruotare il globo e zoomare.

**W29 FoldCard (tocca per selezionare il fold)**

Sidebar destra, sezione «Angular Correlation»..



Per fold una vista carta — rettangolo arrotondato con raggio 6 pt, padding 10, layout verticale con due righe (in alto «Fold N» + numero fotocamere, in basso «Mean nearest angle:» + valore in gradi). Colore di sfondo condizionale: fold attivo = colore di accento semitrasparente, inattivo = materiale standard neutro. Toccando seleziona il fold, e il globo si ricolora dal vivo.

Il punteggio «Mean nearest angle» è l'angolo medio più piccolo per fotocamera di test alla fotocamera di training più vicina (calcolato internamente in radianti, visualizzato in gradi nell'UI).

**IN PAROLE SEMPLICI**

Per fold una piccola carta a destra con numero di fotocamere e distanza media dalla fotocamera di training più vicina. Cliccarla seleziona questo fold come test.

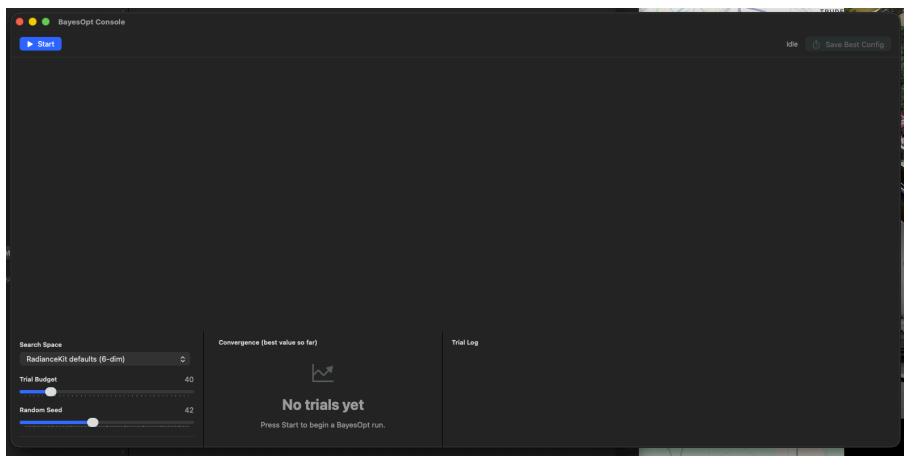
**BayesOpt Console (W30–W39)**

Figura 24: Console BayesOpt — stato vuoto prima dell'avvio dei trial

Stato vuoto con picker Search Space (RadianceKit defaults (6-dim)), slider Trial Budget (default 40), Random Seed (42) e tre pannelli vuoti per Convergence Chart, Trial Log e lista di parametri dello spazio di ricerca.

Stato vuoto (dopo la prima apertura) — Convergence Chart e tabella dei trial si riempiono non appena viene avviato un run, vedi il prossimo scatto.

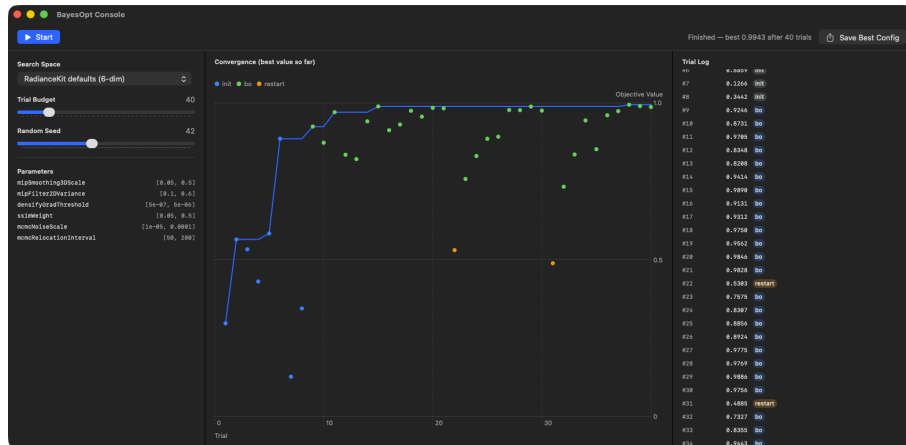


Figura 25: Console BayesOpt dopo 40 trial — Convergence Chart sale ripido fino al trial 15, Best Value 0.9943, Trial Log con tag init/bo/restart

**COSA MOSTRA L'IMMAGINE** Stato in alto a destra «Finished — best 0.9943 after 40 trials». Sidebar sinistra: picker Search Space su RadianceKit defaults (6-dim), Trial Budget 40, Random Seed 42. La lista dei parametri mostra i sei iperparametri da tunare con i loro intervalli di valori: 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]. Al centro: Convergence Chart (X = indice trial 1-40, Y = valore objective 0-1) — punti grigi = sample iniziali (LHS), punti blu = acquisizione BayesOpt, punti arancioni = trial di restart (#22 e #31). La linea del valore migliore sale ripida fino al trial ~7, poi solo miglioramento marginale fino al trial 15, da lì plateau piatto a 0.99+. Sidebar destra: log trial #1-#34 con score + tag (init/bo/restart). Il pulsante Save Best Config in alto a destra scrive `bayesopt-best.json`.


**Cos'è:** Una console di ottimizzazione bayesiana per la ricerca di iperparametri. L'ottimizzazione bayesiana è una procedura automatica che cerca di trovare il punto ottimale di una funzione sconosciuta con il minor numero possibile di esperimenti — tipicamente: «quale combinazione di mcmcMaxGaussians, capMultiplier, ssimWeight e gradThreshold fornisce il miglior PSNR per la mia classe di scena?» Invece di una griglia di  $6^4 = 1296$  trial, BayesOpt prova circa 40-100 trial informati e si avvicina così all'ottimo.

**Importante:** La versione attualmente fornita nell'app non esegue l'ottimizzazione contro esecuzioni di training reali (durerebbe giorni), ma contro un obiettivo demo sintetico — un paesaggio multi-modale con carattere di hill climbing più rumore leggero. È deliberatamente così: la finestra dovrebbe mostrarti il comportamento dell'ottimizzatore (andamento di convergenza, sample point, best so far) e farti capire le definizioni dello spazio di ricerca. Per veri run BayesOpt guidati da training (come eseguiti nella Fase Q7 per i preset Scene-Class) viene usato un workflow CLI offline separato; la finestra è la variante UI dal vivo.

**QUANDO APRIRE** Tre casi d'uso: 1. Vuoi capire come lavora BayesOpt — allora avvia un run demo e osserva il Convergence Chart. 2. Pianifichi una nuova classe di scena (ad esempio «acquari» o «mobili antichi») per la quale i 10 preset integrati non si adattano

perfettamente. Definisci mentalmente uno spazio di ricerca, verificalo qui con la «Bowl demo» o il preset «Densify», esporta poi la best config come JSON e usala come punto di partenza per un vero run di training. 3. Vuoi ispezionare gli spazi di ricerca di default definiti nel pacchetto RKBayesOpt (Mip subset, RadiancKit Defaults) — vengono elencati nel pannello dei parametri della sidebar sinistra.

**COME INTERPRETARE** - **Convergence Chart** (colonna centrale): Y = miglior valore della funzione obiettivo finora raggiunto. X = indice del trial. Inizialmente in salita ripida (BayesOpt prova casualmente i sample iniziali, alcuni sono fortunati), poi sempre più piatto, perché la regione vicino all'ottimo è esaurita. Se la linea rimane piatta per 20+ trial, puoi fermare il run — altri trial non portano più nulla. I singoli punti nel chart sono i valori individuali del trial (quindi non «best so far»), colorati per fase: grigio = sample iniziale, blu = acquisizione bayesopt, arancione = restart. - **Tabella trial** (colonna destra): #1, #2, #3, ... con rispettivamente valore e tag fase. Il miglior trial finora è contrassegnato con una stella gialla. Dalla tabella puoi identificare il best trial e visualizzare in seguito i suoi valori dei parametri all'esportazione. - **Ispettore Search Space** (sidebar sinistra): mostra per il preset selezionato tutti i nomi dei parametri e i loro intervalli di ricerca `[lo, hi]`. Se sei sul preset «RadiancKit defaults (6-dim)», vedi ad es. «densifyGradThreshold [5e-7, 5e-6]» — quindi log uniforme tra questi due valori.

 Seleziona il preset «RadiancKit defaults (6-dim)», Trial Budget 40, seed 42. Clicca «Start». Osserva: i primi 8 trial sono grigi (sample iniziali, LHS Latin Hypercube), i seguenti blu (acquisiti da BayesOpt). Il Convergence Chart sale ripido fino al trial ~15, poi si appiattisce. Al trial ~30–40 si stabilizza il valore migliore. Clicca «Save Best Config» — viene salvato un `bayesopt-best.json` con nome preset, indice trial, valore e valori dei parametri decodificati. Questo JSON puoi poi trasferire manualmente nella tua definizione di preset.

**W30 Pulsante «Start»**

Toolbar a sinistra, nello stato Idle/Finished..



Resetta la lista dei trial, passa allo stato Running, genera un nuovo run ID (per il rilevamento stale di clic Start multipli) e crea un fresco pause gate. Poi avvia un task in background che esegue l'ottimizzatore come stream asincrono. La dimensione dei sample iniziali risulta da  $\min(8, \text{budget} / 4 + 1)$  — quindi tipicamente 8 sample Latin Hypercube con budget  $\geq 28$ , meno con budget piccolo. Gli aggiornamenti dei trial vengono ricevuti in modo incrementale e aggiunti alla lista. Protezione da run stale: se nel frattempo un secondo clic Start imposta di nuovo il run ID, gli aggiornamenti dal vecchio run vengono scartati.

Stile primary action per il look prominente del pulsante.

**IN PAROLE SEMPLICI**

Avvia una fresca esecuzione di ottimizzazione con lo spazio di ricerca, il budget e il seed attuali.

**W31 Pulsante «Pause»**

Toolbar a sinistra, nello stato Running..



Imposta il pause gate attivo e passa allo stato Paused. L'effetto effettivo: il runner attende in un polling loop di 50 ms prima di valutare la prossima funzione obiettivo. Ciò significa che un trial attualmente in esecuzione viene portato a termine (è sintetico e dura solo microsecondi), ma nessun altro trial viene avviato. Non appena Resume gira, prosegue da dove si era fermato.

**IN PAROLE SEMPLICI**

Mette in pausa il run. Il calcolo attuale prosegue fino alla fine, poi va in pausa.

**W32 Pulsante «Stop»**

Toolbar a sinistra, negli stati Running e Paused..



Interrompe il task del runner, azzera il riferimento, rilascia il pause gate (se ancora in pausa) e passa allo stato Finished (se esistono trial) o Idle (se nessuno). I trial già calcolati rimangono visibili nella lista — Stop non li elimina. Il ruolo distruttivo del pulsante mostra il pulsante in rosso, perché interrompe il run.

**IN PAROLE SEMPLICI**

Interrompe definitivamente il run. I trial rimangono visibili, puoi comunque esportare la best config.

**W33 Pulsante «Resume»**

Toolbar a sinistra, nello stato Paused..



Rilascia il pause gate e torna allo stato Running. Il task del runner è già in esecuzione (aspetta nel polling loop); non appena il loop si accorge che la pausa è stata rilasciata, prosegue e avvia il prossimo trial.

**IN PAROLE SEMPLICI**

Riprende un run in pausa.

**W34 Pulsante «Save Best Config»**

Toolbar a destra, sempre visibile (ma disabilitato se non c'è bestTrial)..



Apri una finestra di salvataggio con nome file predefinito `bayesopt-best.json`, limitato a JSON. Dopo la conferma viene costruito un payload dictionary: nome preset, indice trial, valore (objective score), parametri (dictionary dei nomi dei parametri decodificati → valori). La decodifica proietta le coordinate dello spazio di ricerca normalizzato in  $[0,1]^d$  indietro nell'intervallo di valori originale (con scale log-uniform/lineare/intera di conseguenza). L'output JSON è pretty-printed e con chiavi ordinate. In caso di errore di scrittura (nella versione demo attuale) viene silenziosamente ignorato — nessuna UI di errore, perché è un percorso demo. Il pulsante rimane in grigio finché nessun trial è stato eseguito.

**IN PAROLE SEMPLICI**

Salva i valori dei parametri del best trial finora come JSON. Puoi poi trasferire manualmente questi valori nella tua configurazione di preset.

**W35 Picker «Search Space» preset**

Sidebar sinistra, in alto..



Picker di menu con quattro opzioni preset: - «RadiancKit defaults (6-dim)» — lo spazio di ricerca standard completo con tutti gli iperparametri Q7. - «Mip subset (2-dim)» — solo `mipSmoothing3DScale` [0.05, 0.5] log uniforme e `mipFilter2DVariance` [0.1, 0.6] lineare. Utile se vuoi sintonizzare Mip-Splatting per una classe di scena. - «densify-until + ssim-weight + grad-thresh» — tre parametri rilevanti per la densify (`densifyGradThreshold` log uniforme, `ssimWeight` lineare, `densifyUntilIter` intero). - «Bowl demo (1-dim)» — spazio di ricerca pedagogico a parametro singolo per demo «così funziona BayesOpt».

Mentre un run è attivo, lo spazio di ricerca non può essere cambiato (confonderebbe l'ottimizzatore).

**IN PAROLE SEMPLICI**

Quale spazio di ricerca degli iperparametri esamina BayesOpt. Lo standard è «RadiancKit defaults». Per tentativi di tuning Mip mirati «Mip subset». Per capire come lavora BayesOpt «Bowl demo».

**W36 Slider «Trial Budget»**

Sidebar sinistra, sotto il picker Search Space..



Slider da 10 a 200, passo 5. Default 40. Significa: BayesOpt può fare al massimo N trial. Di questi i primi sono sample iniziali (Latin Hypercube), il resto sono veri trial BayesOpt. Regole pratiche per la pratica: uno spazio di ricerca con d dimensioni ha bisogno di circa  $10d$  a  $20d$  trial per un buon ottimo. Con default 6-dim quindi 60–120, con sottoinsieme Mip 2-dim 20–40, con Bowl Demo 1-dim 10–20.

Durante l'esecuzione lo slider è disabilitato.

 IN PAROLE SEMPLICI


Quanti tentativi di ottimizzazione al massimo. Più tentativi = soluzione migliore, ma costa più tempo. 40 è un buon default per l'obiettivo demo.

**W37 Slider «Random Seed»**

Sidebar sinistra, sotto lo slider budget..



Slider da 1 a 100, passo 1. Default 42. Il seed viene passato sia ai sample iniziali Latin Hypercube sia al componente rumore dell'obiettivo demo. Riproducibilità: stesso seed + stesso spazio di ricerca + stesso budget produce esattamente la stessa sequenza di trial. Utile per «tutti i tuoi colleghi ottengono lo stesso run se ricostruiscono la demo?». Durante il run disabilitato.

 IN PAROLE SEMPLICI

Controlla il generatore casuale. Stesso seed = stesso run — per riprodurre.

**W38 Chart (Convergence)**

DOVE

Colonna centrale della finestra..



TECNICO

Grafico Swift Charts con due layer: 1. una linea per «best value so far» per trial — una curva monotona-mente crescente o invariata in colore di accento. 2. un punto per trial con il valore objective individuale, colorato per fase. Dimensione simbolo 40. Tre etichette fase: «init» (grigio), «bo» (blu), «re-start» (arancione).

Una piccola legenda mostra i colori delle fasi in alto a sinistra. Se la lista dei trial è vuota (prima del primo start), viene mostrata invece una visualizzazione empty state con icona chart e nota «Press Start to begin a BayesOpt run.».

**IN PAROLE SEMPLICI**

Il chart di andamento. La linea continua è «miglior soluzione finora trovata»; i punti sono i singoli tentativi. Se la linea rimane piatta per molto tempo, BayesOpt ha trovato l'ottimo.

**W39 Table (Trial Log)**

DOVE

Colonna destra della finestra..



TECNICO

Area di scorrimento con righe di trial impilate lazy. Per riga uno stack orizzontale: numero trial (3 cifre monospaziato, a sinistra), valore (monospaziato, allineato a destra, larghezza 70 pt), tag fase (capsule, riempito con colore fase al 25% di opacità), opzionalmente una stella gialla se questo trial è quello attualmente migliore. Un meccanismo di auto-scroll salta automaticamente alla fine non appena un nuovo trial si aggiunge — in modo che tu possa leggere l'andamento dal vivo in fondo allo schermo, senza dover scrollare tu stesso.

**IN PAROLE SEMPLICI**

La tabella di tutti i tentativi. Valore, fase, stella per il migliore. Scrolla automaticamente, i nuovi trial appaiono in basso.

**Finestra principale: andamento della perdita e Gaussian Count (I39–I41, rimando)**

Tre delle visualizzazioni dell'Inspector nella finestra principale meritano una propria spiegazione, perché sono costantemente visibili durante un training in corso e ci sono regole pratiche importanti su quando l'andamento appare sano. Le visualizzazioni sono

nell'Inspector sotto la sezione «Loss Chart» (vedi Capitolo 2 — Inspector) e completano l'analisi Holdout dalla finestra ausiliaria sopra.

**Quando la curva Loss è sana?** Una curva Loss sana mostra tre fasi: (1) **Warmup** — nelle prime 200–500 iterazioni il loss scende ripido dall'alto (tipicamente 0.15–0.25 per L1+SSIM combinato a seconda della scena) a circa la metà. Se il loss in questa fase NON scende, di solito l'input è sbagliato (immagini danneggiate, pose SfM scadenti, numero di gaussiane iniziali troppo piccolo). (2) **Densification** — tra ~500 e densifyUntilIteration (classicamente 15K, MCMC fino a 20K o 25K) il loss continua a scendere, spesso con piccoli salti in giù quando le operazioni di densify inseriscono nuove gaussiane e l'optimizer le sfrutta. Il gaussian count sale in questa fase. (3) **Refinement** — dopo il loss entra in una coda che si appiattisce. Valori finali tipici: Tanks-&-Temples Truck con P4 Quality si attesta a  $L1 \approx 0.023$ , Horse con Full Classic V546 a  $L1 \approx 0.0230$ , scene outdoor Mip-NeRF360 spesso peggio (0.04–0.07).

**Cosa significa un plateau?** Un plateau (la curva Loss corre orizzontale per diverse migliaia di iterazioni) ha due interpretazioni: (a) il modello è convergente, altro training non porta più nulla — è il caso buono. (b) il modello è bloccato (minimo locale, informazione di gradiente scadente, un cap al limite del buffer) — il caso cattivo. Entrambi appaiono identici nel chart. Distinzione: guarda il gaussian count. Se anche è piatto E vicino al cap MCMC (ad es. 150K su 150K con `.fullMCMC`), sei al limite — o aumenta il cap o accetta il plateau. Se il gaussian count cresce ancora, ma il loss non scende, è bloccato.

**Quando interrompere vs continuare il training?** Regola pratica: 10K iterazioni senza miglioramento del Min Loss → interrompere, ulteriori iterazioni sono sprecate. Prima: puoi tramite Cmd+T (menu Training → Continue Training → +5K iterations) aggiungere ancora un'estensione, se vedi miglioramento borderline. Attenzione: con MCMC il plateau è spesso vero — il cap è il limite naturale.

**Il plateau del Gaussian Count NON è un segnale di «finito».** Significa solo che MCMC ha raggiunto il cap o che la densification Classic è esaurita. La vera domanda «finito» la pone solo l'analisi Holdout — PSNR/SSIM/LPIPS su un set di test indipendenti, valutato nella finestra Holdout (W23–W29) o tramite il flag `--benchmark`.

**PSNR/Holdout è la verità, il Loss solo proxy.** Il loss è una metrica relativa: scende mentre il tuo modello si adatta alle viste di training. Un loss basso non significa però automaticamente buon modello — se il modello ha imparato a memoria le immagini di training (overfitting), il loss sarebbe piccolo, ma il PSNR su viste mai viste (holdout) sarebbe scadente. Quindi: per la valutazione finale della qualità guarda sempre le metriche holdout, non il loss finale da solo.

## Box delle regole pratiche

- User Guide e Keyboard Shortcuts sono aiuto statico — per domande veloci, per la profondità usare il manuale presente.
- Aprire Manage Storage non appena il disco scende sotto il 10% di spazio libero. Log e staging Imports sono i soliti colpevoli.

- Pareto Dashboard sensato solo dopo almeno tre o quattro report di training. Asse X = costo (Time / Gs), asse Y = qualità (PSNR / SSIM). La frontiera di Pareto mostra le combinazioni efficienti.
- Usare Holdout Analysis prima di pubblicare benchmark PSNR con altri — ti assicura che il tuo set di test sia davvero rappresentativo.
- BayesOpt Console è primariamente uno strumento di apprendimento e ispezione per definizioni di spazio di ricerca. Per veri tuning di iperparametri guidati da training usare il workflow CLI offline.
- Loss-plateau e Gaussian-Count-plateau sono da interpretare separatamente. Il limite del cap non è un segnale di «finito». La vera qualità la misura solo il PSNR holdout.
- 10K iterazioni senza miglioramento del Min Loss → fermare il training.

## CAPITOLO

## Capitolo 6 — Configurazione del training

```
preview-preset.json
{
  "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: Preset Preview esportato come JSON + visualizzato in TextEdit — campi id/name/category/version/createdAt/description, trainingConfig con tutti i parametri rilevanti (maxIterations 5000, densifyUntilIteration 3500, ssimWeight 0.20, renderScale 0.50, strategy classic, cameraAlignment applePhotogrammetry, densifyGradThreshold 2.0e-06, opacityResetInterval 3000, minOpacity 0.005, sei toggle bool)

**COSA MOSTRA L'IMMAGINE** Un tipico export JSON di preset. Campi top-level: `id` (UUID), `name`, (`classic` | `mcmc` | `sceneClass` | `custom`), (`version` schema), (`createdAt` timestamp), (`description` testo libero). L'oggetto annidato contiene i parametri critici per la riproducibilità — all'import l'intero blocco viene deserializzato nella struttura `TrainingConfig`, e i default della versione dell'app riempiono i campi mancanti nel JSON (ad es. dopo un aggiornamento dell'app). Chi passa un preset a un altro Mac, invia semplicemente questo file JSON.

La struttura `TrainingConfig` è il cuore di ogni esecuzione di training in RadiancesKit. Raccoglie ogni parametro che influenza il training — dal numero massimo di iterazioni alle otto learning rate fino ai campi speciali per MCMC, Mip-Splatting, curriculum e logica scene-aware cap. La modifichi nella sidebar nell'area sezione Configurazione del training (Vista Esperto), la salvi come preset o la passi come export JSON a un altro

Mac. Al training esattamente questo oggetto viene congelato e passato al backend GPU.

Questo capitolo è materiale di riferimento per power user e autori di script. Elenca tutti gli 81 campi pubblici, i 9 preset statici e l'unico metodo pubblico. Il file sorgente è `TrainingConfig.swift` — in caso di dubbio vale come fonte di verità il commento doc lì memorizzato e il default dell'inizializzatore.

#### NOTA · UI VS. PRESET/CLI

Solo 12 degli 81 campi hanno uno slider, toggle o picker diretto nell'Inspector (build sandboxed App Store): **T1, T2, T17, T20, T22, T38, T56–T58, T60, T61, T73**. I restanti 69 campi vengono impostati tramite il **preset** scelto (Capitolo 7) e possono essere sovrascritti direttamente solo tramite **flag CLI** (vedi Capitolo 5). Questa separazione è intenzionale: i default rimangono stabili e collaudati in produzione, i power user hanno comunque una via di fuga. Se un campo ti interessa particolarmente: prima guarda nel Capitolo 2 (Inspector) e nel Capitolo 5 (CLI) se puoi raggiungerlo senza smanettare con JSON.

#### Indice:

1. Iterazione (T1–T2)
2. Learning Rate (T3–T10)
3. Densification — Classic (T11–T16)
4. Loss (T17–T20)
5. Progressione del grado SH (T21)
6. Prestazioni (T22–T25)
7. Diagnosi e preparazione della nuvola di punti (T26–T30)
8. Regolarizzazione (T31–T37)
9. Refinement (T38–T44)
10. Sky-Dome (T45–T48)
11. Adam + LR-Schedule (T49–T55)
12. Post-processing + Apple AI (T56–T60)
13. Densification MCMC (T61–T73)
14. Mip-Splatting (Q1.5) (T74–T76)
15. Densification adattiva (Q5) (T77–T79)
16. Curriculum (Q6) (T80–T81)
17. Preset statici (TP1–TP9)
18. Metodo:
19. Quale campo per cosa? (cheat sheet)
20. Campi pericolosi

## Iterazione (T1–T2)

### T1 maxiterations

#### DETTAGLI

**Default:** 30 000 (inizializzatore), 35 000 ( `.full` ), 200 000 ( `.fullMCMC` ) **Range:** 1 000 – 500 000 (slider UI), nessun limite superiore rigido nella logica  
**Defined in:**

#### TECNICO

Numero totale di iterazioni di training che il backend esegue. Un'iterazione indica un forward render di una singola fotocamera di training, un backward pass su tutti i componenti di loss (L1 + SSIM + regolarizzazioni opzionali + sky mask) e uno step Adam optimizer. Questo numero agisce direttamente sugli altri schedule: la learning rate di posizione segue una curva di cosine annealing da 0 fino o a `T1` stesso o a `T49 positionLRScheduleEndIteration`; la densification si ferma a `T2 densifyUntilIteration`; il decadimento del rumore MCMC termina a `T69 mcmcNoiseDecayEnd`; gli upgrade del grado SH avvengono ai tre marker definiti in `T21`. Con densification classica il sweet spot determinato empiricamente è 20 000–35 000 iterazioni (sessioni 1–32, test V546), con MCMC 60 000–200 000 (V534). Un aumento drastico oltre i valori memorizzati nel preset raramente porta qualità aggiuntiva — l'Adam momentum satura, e senza fine del LR decay il loss ristagna. Viceversa, scendere sotto ~5 000 porta a geometrie convergenti in modo incompleto (il density control ha troppo poco tempo per clone/split).

#### IN PAROLE SEMPLICI

Quanto a lungo l'app calcola. Più iterazioni = risultato migliore, ma a un certo punto non più visibilmente migliore, ma molto più a lungo. I preset sono scelti in modo che tu abbia un buon valore senza pensarci: Quick 1 000, Preview 5 000, Balanced 20 000, Quality 35 000, MCMC Quality 200 000. Se lo regoli tu stesso: con MCMC puoi salire (100 000–200 000), con Classic non oltre 40 000 — dopo non porta più nulla.

## **T2** densifyUntilIteration

### DETTAGLI

**Default:** 15 000 (inizializzatore), 5 000 ( `.full` ), 160 000 ( `.fullMCMC` ) **Range:** 0 – **Defined in:**

### TECNICO

Iterazione a partire dalla quale la densification si ferma. Fino a qui le gaussiane vengono clonate, divise e pruned secondo le regole parametrizzate in `T11–T16` (Classic) o `T67–T70` (MCMC); dopo il numero di gaussiane rimane costante e solo posizioni, rotazioni, scale, opacità e coefficienti SH vengono ottimizzati (fase di refinement). Nel paper 3DGS originale il valore è al 50% di `T1`, nel preset `.full` di RadiancesKit solo a ~14% (5 000 su 35 000) — conseguenza degli esperimenti V310/V338, che hanno mostrato che dopo 5 000 iterazioni un'ulteriore densificazione peggiora piuttosto il risultato (più floater, più memoria, nessun guadagno di qualità). MCMC invece esegue la rilocalizzazione fino all'80% di `T1` (V504b), perché MCMC non produce floater dannosi. Se `T2` è troppo piccolo ( $< 1\,000$ ), nascono troppe poche gaussiane; troppo grande con Classic ( $> 50\%$  di `T1`) porta a overgrowth e RGB saturation outlier (vedi findings di outdoor overtraining).

### IN PAROLE SEMPLICI

Fino a quando l'app può creare nuove gaussiane. Dopo viene solo raffinato ciò che è già lì. Con training classico di 35 000 iterazioni qui 5 000 è il valore giusto — tutto sopra rende la scena più pastosa. Con MCMC è 80% delle iterazioni totali (quindi 160 000 con run di 200 000). Se modifichi il preset Quality, lascia stare questo campo.

## Learning Rate (T3–T10)

### T3 positionLearningRate

#### DETTAGLI

**Default:** 0.00016 **Range:** 1e-7 – 1e-3 (raccomandato) **Defined in:**

#### TECNICO

Learning rate Adam per la posizione XYZ di ogni gaussiana all'inizio del training (iterazione 0). Segue una curva di cosine annealing e scende durante il training a `T4 positionLearningRateFinal`. Il default 0.00016 proviene dal paper 3DGS originale (Kerbl et al. 2023) e in RadiancKit non va scalato anche all'aumentare della risoluzione dell'immagine — la posizione si muove nel sistema di coordinate mondo, non nello spazio dei pixel. Un aumento significativo ( $> 0.0005$ ) fa sì che le gaussiane saltino su lunghe distanze e il loss diventi instabile; valori significativamente più bassi ( $< 0.00005$ ) portano nuvole di punti inizializzate in modo errato a non trovare mai il loro posto. V414 ha testato il raddoppio del valore di init  $\rightarrow$  16.8% di loss L1 peggiore; i tuning V544a hanno confermato il default del paper come ottimale. Nota: con `.fullMCMC` lasciamo deliberatamente questo valore al default — MCMC ha bisogno di learning rate costanti per la sua logica di rilocalizzazione, quindi il tuning qui non porta nulla.

#### IN PAROLE SEMPLICI

Quanto velocemente i punti splat possono muoversi nello spazio. Il valore standard è molto ben calibrato e non ha realmente bisogno di modifiche. Solo se vedi splat che «sbandano» nell'immagine o un intero angolo manca perché nulla si sposta, il learning rate sarebbe un punto su cui regolare — ma tipicamente prima qualcos'altro non quadra (pose delle fotocamere, nuvola di punti iniziale).

**T4** positionLearningRateFinal DETTAGLI

**Default:** 0.0000016 (inizializzatore + paper), 0.000016 ( `.full` , `.fullMCMC` — 10x più alto) **Range:** 0 — **Defined in:**

 TECNICO

Valore finale della curva di cosine annealing della LR di posizione. Viene raggiunto o a `T1 maxIterations` o, se impostato, a `T49 positionLRScheduleEndIteration`. Il preset `.full` di RadianceKit usa 0.000016 — quindi 10x più alto del default del paper 0.0000016. Gli esperimenti V420 hanno mostrato che 0.5x del valore finale (0.000008) peggiora il loss del 6.4%; V414 ha mostrato che 2x del valore di init lo peggiora del 16.8%. L'alto valore finale non è un trade-off, ma una scelta deliberata: con decay troppo forte le gaussiane perdono nella fase di refinement la capacità di adattarsi ai candidati di densification appena aggiunti. Tramite l'estensione V431/V433 la fase schedule può essere accorciata ( `T49 < T1` ), in modo che `T4` venga raggiunto già prima della fine del training e il resto del training corra con mini LR costante — configurazione tipica: `T49 = 20 000` , `T1 = 35 000` , refinement quindi a 0.000016 per 15 000 iterazioni.

 IN PAROLE SEMPLICI

Quanto lentamente diventa la LR di posizione alla fine del training. L'abbiamo deliberatamente impostato meno aggressivo del paper originale — gli splat possono ancora tremolare un po' fino alla fine, questo li rende più nitidi. Se ci regoli: più alto = splat più irrequieti alla fine, più basso = gli splat non possono più adattarsi quando ne appaiono di nuovi.

**T5** shDCLearningRate **DETTAGLI**

**Default:** 0.0025 (inizializzatore + paper), 0.005 ( `.full` e tutti i preset MCMC — 2x) **Range:** 0.0001 – 0.05 **Defined in:**

 **TECNICO**

Learning rate Adam per la quota DC (grado 0, quindi albedo costante) del colore SH. SH-DC corrisponde al tono base indipendente dalla direzione di una gaussiana, in un certo senso il «colore base». Gli esperimenti V176 e V188 hanno trovato 2x più alto del default del paper come ottimale — convergenza del colore più veloce, soprattutto perché con training breve (5 000 iterazioni) altrimenti SH-DC non si forma. A differenza delle LR geometriche SH-DC non ha decay; la learning rate rimane costante su tutte le iterazioni (o segue solo il decay opzionale extended phase di `T51`). V416 ha testato la quadruplicazione a 0.01 → 6.4% di loss peggiore con Adam beta2=0.99.

 **IN PAROLE SEMPLICI**

Quanto rapidamente il colore base di ogni splat si adatta. Il valore non si cambia quasi mai da soli — i preset hanno il valore giusto. Più alto andrebbe più veloce, ma può portare a colori instabili.

**T6** shRestLearningRate **DETTAGLI**

**Default:** 0.000125 (inizializzatore + paper), 0.00025 ( `.full` e MCMC — 2x) **Range:** 0.000001 – 0.005

**Defined in:**

 **TECNICO**

Learning rate Adam per i coefficienti SH di ordine superiore (grado 1, 2, 3 — quindi le quote di colore dipendenti dalla direzione di vista, che si occupano di luci speculari, riflessi e ombreggiature morbide). 20x più piccolo di `T5` per convenzione del paper, perché questi coefficienti crescono quadraticamente in numero (3 per grado 1, 5 per grado 2, 7 per grado 3 → in totale 15 float per gaussiana) e senza learning rate più piccola sovrasaturerebbero l'immagine. Viene sbloccato in due passaggi — fino al primo marker in `T21` `shDegreeUpgradeIterations` è attivo solo il grado 0 (quindi solo `T5`), dopo 1, poi 2, infine 3. Valori bassi qui sono particolarmente importanti su scene con molta illuminazione diffusa; su superfici molto lucide (vernice auto, acqua) non vale la pena regolare — la rappresentazione SH di per sé è limitata.

 **IN PAROLE SEMPLICI**

Quanto velocemente imparano gli effetti di colore dipendenti dalla direzione (riflessi, lucentezza). Di default molto piccolo, perché altrimenti tutto luccica. Il valore è meglio lasciarlo stare — chi vuole ottenere luci speculari migliori è meglio servito da MCMC e tempi di training più lunghi che da questa LR.

## T7 opacityLearningRate

### DETTAGLI

**Default:** 0.05 (inizializzatore + paper), 0.1 ( `.full`, MCMC — 2x) **Range:** 0.001 – 1.0 **Defined in:**

### TECNICO

Learning rate Adam per l'opacità logit di ogni gaussiana. L'app memorizza l'opacità come valore float non limitato e la trasforma con sigmoid in  $[0, 1]$ ; la LR agisce nello spazio logit. Il default del paper 0.05 è ripristinato dopo i test V50 (best single run L1 0.1664), V71 ha ripristinato 0.025 di V67. Il raddoppio V188 a 0.1 rende il pruning più efficiente — le gaussiane morte cadono più velocemente sotto la T14 `pruneOpacityThreshold`. V418 ha mostrato: 0.05 con Adam `beta2=0.99` è 7.1% peggiore di 0.1 — l'interazione con la configurazione Adam non è banale. Valori bassi ( $< 0.01$ ) portano le gaussiane «morte» a rimanere in giro per sempre e consumare memoria; valori troppo alti ( $> 0.5$ ) possono portare a esplosione dell'opacità, quindi il valore logit nell'optimizer viene limitato a  $[-15, 3]$  (vedi nota «Opacity Explosion Prevention» in CLAUDE.md).

### IN PAROLE SEMPLICI

Quanto velocemente gli splat diventano trasparenti o opachi. Importante per la pulizia — gli splat che non contribuiscono devono sparire velocemente, altrimenti si forma un velo. Il valore standard va bene, solo i professionisti lo modificano.

**T8** `opacityLearningRateFinal` **DETTAGLI**

**Default:** 0.0 (= «nessun decay») **Range:** 0 oppure 0.001 – **Defined in:**

 **TECNICO**

Valore finale opzionale di cosine decay per la LR di opacità (V427). Se 0.0, decay disabilitato e la LR di opacità rimane costante a T7 su tutto il training. V427 ha testato un decay 0.1 → 0.01 — risultato 11.5% di loss peggiore; revertito, da qui il default «off». L'ipotesi dietro il campo: nella fase di refinement una LR di opacità costante potrebbe portare a oscillazione, in modo che splat che hanno già raggiunto il giusto grado di trasparenza vengano spostati di nuovo da fluttuazioni casuali del gradiente. Empiricamente non si conferma — la logica di logit clamping lo intercetta comunque. Il campo rimane disponibile per esperimenti futuri; anche run MCMC molto lunghi (> 500K iterazioni) potrebbero beneficiarne.

 **IN PAROLE SEMPLICI**

Se la learning rate di opacità debba diventare più piccola verso la fine. Standard: no. L'abbiamo provato, era peggio, lo lasciamo disabilitato. Rimani a 0.

**T9** `scaleLearningRate` **DETTAGLI**

**Default:** 0.005 (inizializzatore + paper), 0.01 (`.full`, MCMC — 2×) **Range:** 0.0001 – 0.1 **Defined in:**

 **TECNICO**

Learning rate Adam per le tre componenti di scala di ogni gaussiana nello spazio log (RadianceKit memorizza  $\log(\text{scale})$ , in modo che le scale rimangano positive). Il default del paper 0.005, in RadianceKit raddoppiato a 0.01 per migliore convergenza di scala con le configurazioni di learning rate ottimizzate. Esperimento V423: 0.005 con Adam  $\beta_2=0.99$  → 18.7% di loss peggiore e visibilmente troppo poche gaussiane (il density control non poteva clonare, perché gli update di scala erano troppo lenti). La scala controlla l'estensione di ogni gaussiana — apprendimento troppo veloce porta a gaussiane «ago» (splat estremamente lunghi e sottili, vedi T34 `scaleRatioPruneThreshold`), apprendimento troppo lento fa rimanere gli splat troppo compatti e il density control deve splittare troppo spesso.

 **IN PAROLE SEMPLICI**

Quanto velocemente si adatta la forma degli splat. Lo standard è buono. Se lo aumenti, ottieni volentieri splat «ago» — gocce estremamente lunghe e sottili, che fanno fluttuare l'immagine.

**T10** rotationLearningRate **DETTAGLI**

**Default:** 0.001 (inizializzatore + paper), 0.002 (`.full`, MCMC — 2×) **Range:** 0.0001 – 0.05 **Defined in:**

 **TECNICO**

Learning rate Adam per le quattro componenti quaternioni di ogni gaussiana. Il quaternioni viene rinormalizzato ad ogni step optimizer dopo l'update Adam (norma L2 = 1) — altrimenti la matrice di covarianza diventerebbe degenere. RadianceKit raddoppia il default del paper nei preset Quality, perché la rotazione rispetto a scala/posizione ha magnitudini di gradiente assolute più piccole (sulla sfera unitaria ogni step rimane breve) e senza 2× la rotazione nella finestra di 35 000 iterazioni sarebbe nettamente sotto-convergente. V188 documentato. Su scene NeRF-Blender (Lego, Chair) la rotazione si ripercuote in modo particolare — gli spigoli degli oggetti si allineano correttamente solo dopo 5 000–10 000 iterazioni.

 **IN PAROLE SEMPLICI**

Quanto velocemente gli splat imparano a ruotare — quindi a venire nell'orientamento corretto sulla superficie di un oggetto. Lo standard va bene. In altre parole: se gli splat appaiono come mattoncini messi storti invece di aderire alla superficie, è piuttosto il tempo di training troppo breve, non questa learning rate troppo bassa.

## Densification — Classic (T11–T16)

### T11 densifyGradThreshold

#### DETTAGLI

**Default:** 0.000002 (inizializzatore, calibrato per risoluzione 0.5x), 0.0000011 ( `.full` , calibrato per 1.0x), 0.000004 ( `.quickTest` , calibrato per 0.25x),  $2e-7$  ( `.fullClassicPaper` ) **Range:**  $1e-8 - 1e-3$  (dipendente dalla risoluzione) **Defined in:**

#### TECNICO

Soglia per la norma L2 del gradiente proiettato nello screen space `dMean2D` , sopra la quale una gaussiana viene contrassegnata per clone o split. Il valore assoluto dipende direttamente dalla risoluzione di training — `dMean2D` scala approssimativamente come  $1/\text{risoluzione}^2$  (più pixel = gradienti per pixel più piccoli). Quindi ogni stadio di `T22` `trainingRenderScale` ha bisogno di una soglia calibrata:  $0.25x \rightarrow 4e-6$ ,  $0.5x \rightarrow 2e-6$ ,  $1.0x \rightarrow 5e-8$  ...  $1.1e-6$  ( `.full` ). Il default del paper 0.0002 è NDC-normalizzato e nella pipeline spazio-mondo di RadianceKit non direttamente confrontabile. Con il flag `T52` `adaptiveDensifyThreshold` aggiunto in V440 il valore può essere calcolato in runtime dal p98 della distribuzione attuale del gradiente — ma V440 l'ha testato su scene reali e ha prodotto 63 K gaussiane (perdita catastrofica di pruning); il flag rimane off. Q5 (T77–T79) fornisce una logica adattiva alternativa via rolling median. **Questo campo non è privo di pericoli** — dimezzandolo si producono 2–4x più gaussiane (pressione di memoria, rischio OOM); raddoppiandolo si può sotto-densificare la scena.

#### IN PAROLE SEMPLICI

Quanto è sensibile l'app quando deve decidere se uno splat è sotto-rappresentato e deve essere moltiplicato. Valore basso = più sensibile = più splat. Più alto = meno splat. È uno dei valori più pericolosi in assoluto: troppo basso e il Mac si riempie di memoria con milioni di splat e potrebbe crashare. Lascia stare il campo, o modificalo solo in step del 10%.

**T12** densifyFromIteration **DETTAGLI****Default:** 500 **Range:** 100 – 5 000 **Defined in:** **TECNICO**

Prima iterazione a partire dalla quale la densification diventa attiva. Prima avviene solo apprendimento «nudo» sulla nuvola di punti SfM iniziale, senza che vengano create nuove gaussiane. Il default 500 proviene dal paper 3DGS e dà tempo all'inizializzazione di stabilizzarsi — se si densifica già dall'iterazione 0, i punti SfM posizionati in modo errato si clonano in molte copie, prima che trovino il loro posto corretto. V349 ha testato 1000 → loss leggermente peggiore; il default è ottimale.

 **IN PAROLE SEMPLICI**

Quando l'app inizia per la prima volta a clonare splat. Prima impara solo i punti già presenti. 500 è il valore standard — dà all'app abbastanza tempo per orientarsi prima di moltiplicare.

**T13** densifyInterval **DETTAGLI****Default:** 100 (inizializzatore, MCMC), 200 ( `.full` )  
**Range:** 50 – 1 000 **Defined in:** **TECNICO**

Quante iterazioni si trovano tra due step di densification. Nel default del paper 100 — ogni 100 iterazioni viene valutata la lista dei candidati di densify, clonata/divisa e contemporaneamente la lista dei candidati di prune ( $\text{sigmoid}(\text{opacity}) < T_{14} \text{pruneOpacityThreshold}$ ) rimossa. I test V112 hanno trovato 200 come ottimale per `.full` — ciò alleggerisce la GPU, perché vengono eseguiti meno pass di riorganizzazione, e dà a ogni gaussiana più tempo per stabilizzarsi dopo un'azione di clone. V417 ha testato 100 con  $\text{beta2}=0.99 \rightarrow 5.8\%$  peggiore (957 K gaussiane, sovra-densificazione). Con MCMC lo stesso campo viene interpretato come intervallo di rilocazione; vedi T67 `mcmcRelocationInterval` per la logica specifica MCMC.

 **IN PAROLE SEMPLICI**

Con quale frequenza l'app cerca nuovi splat. 100 = spesso, 200 = medio. Più alto significa: ogni splat ha più tempo per sistemarsi prima che venga moltiplicato di nuovo. È bene. Abbassarlo a 50 può tenere la GPU costantemente occupata senza diventare significativamente migliore.

**T14** `pruneOpacityThreshold` **DETTAGLI**

**Default:** 0.005 (inizializzatore, paper, MCMC), 0.001 (`.full`) **Range:** 0.0001 – 0.1 **Defined in:**

 **TECNICO**

Soglia di opacità sigmoid sotto la quale una gaussiana viene eliminata al prossimo step di densificazione. Agisce insieme a `T7 opacityLearningRate` e alla logica di logit clamp nell'optimizer. V393 ha abbassato il default da 0.005 a 0.001 in `.full` — conseguenza: gli splat che giocano un ruolo solo da angoli di visualizzazione esotici rimangono più a lungo e contribuiscono al dettaglio SH. V394 ha testato 0.0001 → leggermente peggio (troppo poco pruning, memoria sprecata). Importante: il density control deve SEMPRE eseguire il pruning, anche se la capacità del buffer è già piena tramite altre misure (vedi «Density Control Must Always Prune» in CLAUDE.md) — altrimenti le gaussiane morte si accumulano e il count si congela.

 **IN PAROLE SEMPLICI**

Quando uno splat è considerato «abbastanza trasparente» per essere eliminato. 0.005 è lo standard del paper, in Quality abbiamo 0.001 — quindi diamo agli splat più chance. Questo rende meglio rappresentabili luce morbida e ombre deboli. Impostare più alto (sopra 0.01) fa scendere rapidamente il numero di splat — può essere sensato in caso di scarsità di memoria, ma costa dettaglio.

**T15** opacityResetInterval **DETTAGLI**

**Default:** 3 000 (inizializzatore + paper), 100 000 (`.full` = effettivamente disabilitato), 200 000 (`.fullMCMC` = disabilitato) **Range:** 1 000 – 100 000+ **Defined in:**

 **TECNICO**

Ogni quante iterazioni l'opacità di tutte le gaussiane viene resettata a un valore basso (~0.01) — una misura del paper 3DGS per rivalutare splat «congelati». V194 ha mostrato che con il setup warmup + stochastic training + 2× learning rate di RadianceKit il reset di opacità costa il 5.5% di qualità e il logit clamp copre già la funzione di reset. Quindi in `.full` praticamente disabilitato (100 000 > 35 000 = mai attivato). V421 ha testato reset ogni 3 000 con  $\beta_2=0.99 \rightarrow 4.9\%$  peggiore; revertito. Con `.fullClassicPaper` (Q1.5-A, test fedele al paper) è deliberatamente riportato a 3 000 — è una delle leve con cui si dovevano raggiungere i budget di gaussiane con magnitudine paper.

 **IN PAROLE SEMPLICI**

Ogni quante iterazioni l'app resetta la visibilità di tutti gli splat a «quasi invisibile» — una sorta di pulsante reset per l'opacità. Da noi disabilitato (valore così alto che non succede mai), perché altri meccanismi lo rendono superfluo. Attivarlo solo per esperimenti fedeli al paper.

**T16** maxScreenSize **DETTAGLI**

**Default:** 0.0 (= disabilitato) **Range:** 0 (off) o > 0 **Defined in:**

 **TECNICO**

Dimensione massima screen space (in pixel proiettati) che una gaussiana può raggiungere prima di essere splittata forzatamente. Il valore è impostato a 0 (V48 ha testato e revertito) — il density control di RadianceKit usa invece la soglia di scala spazio mondo della logica `dMean2D`. Rimane nel catalogo dei campi, perché futuri esperimenti con Mip-Splatting (T74–T76) o strategie di splatting specifiche per scena potrebbero beneficiarne. L'attivazione (valore > 0, ad es. 20) costringerebbe gli splat che sono diventati molto grandi nello screen a dividersi — rilevante con grandi superfici lisce di pareti, dove un singolo splat gigante offre troppo poco dettaglio.

 **IN PAROLE SEMPLICI**

Limite di quanto può diventare grande un singolo splat sullo schermo. Da noi off. Acceso farebbe dividere forzatamente splat piatti enormi (ad es. su una parete) in più piccoli. Lascialo off, salvo esperimenti espliciti.

## Loss (T17–T20)

### T17 ssimWeight

#### DETTAGLI

**Default:** 0.2 (inizializzatore + paper + `.full`), 0.05 (tutti i preset MCMC) **Range:** 0.0 – 1.0 **Defined in:**

#### TECNICO

Peso della quota D-SSIM nella funzione di loss combinata  $loss = (1 - \lambda) * L1 + \lambda * D\text{-SSIM}$ , dove  $\lambda = T17$ . Il default del paper 3DGS 0.2 è ottimale per la densification Classic — V383 ha testato 0.3 → 28.9% peggiore, V373b ha confermato 0.2 come sweet spot. Per MCMC in V521b/V534 è stato indipendentemente stabilito: 0.05 è ottimale, perché MCMC tramite la sua esplorazione stocastica ha bisogno di una quota di segnale L1 più forte — pesi SSIM più alti annacquerebbero le decisioni di rilocalizzazione. SSIM è significativamente più costoso da calcolare di L1 (finestre locali 11×11 sull'intera immagine); RadianceKit usa un'implementazione accelerata MPS che rimane sotto 1 ms per immagine 1080p. Gli sweep Q7-BayesOpt hanno trovato ottimi specifici per scena tra 0.05 (`.outdoorPreset` : 0.082) e 0.171 (`.indoorPreset` ).

#### IN PAROLE SEMPLICI

Quanto è importante per l'app, oltre a «ogni pixel corretto», anche «le strutture sono simili». 0.2 è lo standard e fornisce una buona immagine. Più basso = più preciso al pixel, ma può ottenere transizioni più morbide. Più alto = più simile strutturalmente, ma i dettagli diventano più morbidi. Lascia decidere ai preset.

**T18** **ssimWeightRefinement** **DETTAGLI**

**Default:** 0.0 (= «nessun cambio, mantieni ssim-Weight») **Range:** 0 oppure 0 – 1.0 **Defined in:**

 **TECNICO**

Valore SSIM opzionale per la fase di refinement dopo T2 `densifyUntilIteration`. V428 ha testato 0.2 → 0.3 in refinement → 16% di loss peggiore (sia L1 sia SSIM sono peggiorati); revertito, da qui default 0.0. L'ipotesi dietro il campo era che dopo la densification — quando non nascono più nuove gaussiane — una quota SSIM più forte massimizzerebbe la nitidezza strutturale. Empiricamente errato: aumentare il peso SSIM significa indirettamente abbassare il peso L1, e L1 è il segnale nettamente più espressivo nella fase di refinement finale. Il campo rimane disponibile per esperimenti futuri con loss percettiva (T60) o edge loss (T19), dove potrebbe essere sensata una composizione di loss specifica per refinement.

 **IN PAROLE SEMPLICI**

Impostazione speciale per la seconda fase di training (refinement dopo la moltiplicazione degli splat). A 0.0: stessa ponderazione SSIM di prima. Regolarlo empiricamente non porta nulla, quindi off.

**T19** edgeLossWeight **DETTAGLI**

**Default:** 0.0 (= disabilitato) **Range:** 0 oppure 0.001 – 1.0 **Defined in:**

 **TECNICO**

Loss sperimentale V437: peso di un loss L1 sul gradient domain Sobel, che confronta direttamente i bordi dell'immagine (Sobel ground truth vs Sobel render) in aggiunta a L1+SSIM. Ipotesi: l'informazione sui bordi è un pilastro percettivo della qualità dell'immagine e un termine esplicito dovrebbe incoraggiare le gaussiane a colpire meglio i bordi. Risultati dei test: peso 0.1 → 11% di loss peggiore, 0.01 → neutrale in qualità ma 10% più lento. Il pass Sobel costa un forward MPS aggiuntivo su ground truth e render. Quindi disabilitato in modo permanente. Caso d'uso futuro: scene con bordi artificiali duri (architettura, mobili, rendering) potrebbero beneficiare — i preset Q7-Scene-Class però non l'hanno preso, ma hanno invece scalato il peso SSIM.

 **IN PAROLE SEMPLICI**

Aggiunta sperimentale che prende i bordi come extra importanti. Empiricamente non porta nulla. Rimane off.

**T20** skyMaskingEnabled **DETTAGLI**

**Default:** false (inizializzatore e tutti i preset) **Range:** boolean **Defined in:**

 **TECNICO**

Attiva il Sky Masking. In ogni immagine la regione del cielo viene mascherata via Apple Vision Framework (VNGenerateForegroundInstanceMaskRequest), e il loss in quest'area viene impostato a zero. Senso: le scene outdoor soffrono spesso del fatto che pixel di cielo blu/grigi/bianchi portano l'app a posizionare gaussiane esattamente lì — ciò che viene percepito come «floater». Senza sky mask il loss in quest'area non sarebbe mai zero, perché il cielo nell'immagine varia leggermente e l'app cerca all'infinito di ricostruirlo con splat. La maschera Vision viene calcolata una volta per fotocamera prima del training e mantenuta in RAM. Tipicamente attivata insieme a T45 skyDomeEnabled (logica UI nella view Impostazioni). Con scene interne o rendering sintetici lasciare disabilitato — la maschera riconoscerebbe erroneamente soffitti o pareti come «cielo».

 **IN PAROLE SEMPLICI**

Attiva una modalità speciale per riprese esterne: il cielo viene ignorato durante il training, in modo da non provare a ricostruirlo con splat. Raccomandato per ogni scena esterna. Con interni o con rendering 3D da Blender lasciare off.

## Progressione del grado SH (T21)

### T21 shDegreeUpgradeliterations

#### DETTAGLI

**Default:** [1\_000, 2\_000, 3\_000] (inizializzatore), [2\_000, 5\_000, 8\_000] (.full, MCMC), [1\_000, 2\_000] (.preview — grado 3 saltato) **Range:** [Int], ogni valore in [0, maxIterations], monotonamente crescente

#### Defined in:

#### TECNICO

Iterazioni a cui il grado SH attivo viene aumentato da 0→1, 1→2, 2→3. Prima del primo marker sono attive solo le componenti DC (quindi T5 shDCLearningRate), dopo il primo marker DC + 3 coefficienti di grado 1, dopo il secondo marker + 5 coefficienti di grado 2, dopo il terzo marker tutti i 15 coefficienti. Il fabbisogno di memoria per gaussiana cresce a stadi — 4 float → 16 float → 36 float → 64 float. I preset Quality ritardano gli upgrade rispetto ai default dell'inizializzatore (V228), perché la geometria dovrebbe prima stabilizzarsi, prima che i dettagli di colore con la loro frequenza più alta vengano sopra. V384 ha testato [1K, 2K, 3K] per .full → 9.3% peggiore — conferma il ritardo. .preview si ferma al grado 2, perché il grado 3 in 5 000 iterazioni non converge e consuma solo capacità optimizer. Q6 (T80–T81) offre una logica curriculum alternativa che sovrascrive dinamicamente questa lista.

#### IN PAROLE SEMPLICI

In quali punti del training l'app impara che i colori da diversi angoli di visualizzazione possono apparire diversi (luci, riflessi). Solo tardi — in modo che prima la forma sia corretta, poi il colore. I valori nei preset sono impostati in modo che funzioni bene. Non modificare nulla, salvo che tu sappia esattamente perché.

## Prestazioni (T22–T25)

### T22 trainingRenderScale

#### DETTAGLI

**Default:** 1.0 (inizializzatore, `.full`, MCMC, Scene-Class), 0.5 (`.preview`), 0.25 (`.quickTest`) **Range:** 0.05 – 2.0 (tipico 0.25, 0.5, 1.0) **Defined in:**

#### TECNICO

Risoluzione di rendering al training relativa alla risoluzione originale delle immagini di training. A 0.5 ogni immagine viene ridotta al 50% larghezza × 50% altezza (quindi 25% dei pixel) e il rendering delle gaussiane avviene in questa risoluzione più piccola. Riduce quadraticamente sia il fabbisogno di memoria che quello di calcolo. Importante: `T11 densifyGradThreshold` deve corrispondere alla risoluzione scelta — le magnitudini di gradiente scalano con  $1/\text{risoluzione}^2$ , quindi `.quickTest` (0.25×) ha una soglia molto più alta ( $4e-6$ ) di `.full` (1.0×,  $1.1e-6$ ). RadianceKit avverte con immagini molto grandi e adatta automaticamente — risoluzione target 3 MP. Con immagini di input estreme 4K 0.5 o anche 0.25 sarebbe sensato, altrimenti ogni Mac va in compaction CPU.

#### IN PAROLE SEMPLICI

Quanto sono grandi le immagini al training. 1.0 = originale, 0.5 = metà. Metà dimensione = quattro volte più veloce, ma i dettagli più fini mancano. I preset scelgono il valore giusto; con immagini di input estremamente grandi (oltre 12 megapixel) l'app passa automaticamente verso il basso.

**T23** resolutionWarmupScale **DETTAGLI**

**Default:** 0.0 (= disabilitato) **Range:** 0 oppure 0.1 –  
**Defined in:**

 **TECNICO**

Ottimizzazione V133: addestra la fase di densificazione (iter 0 a  $T_2$ ) in una risoluzione più bassa rispetto alla fase di refinement. V308 l'ha spenta di nuovo per `.full`, perché con  $T_{22} = 1.0$  e cosine annealing il guadagno di tempo era marginale e la qualità soffriva minimamente. Rimane nel catalogo dei campi, perché potrebbe diventare di nuovo sensata con input 4K e lunghi run di training — Q6 Curriculum (T80) ha ripreso una logica simile, lì è però accoppiata allo LR schedule. Se attivato e `T80 curriculumResolutionRamp` anche true, vince Q6 e sovrascrive questo valore.

 **IN PAROLE SEMPLICI**

Funzione speciale: nella prima metà del training imparare con immagini più piccole, nella seconda con quelle grandi. Risparmia tempo. Off, perché la variante più nuova Q6 lo risolve meglio.

**T24** tileSize **DETTAGLI**

**Default:** 16 **Range:** 8, 16, 32 **Defined in:**

 **TECNICO**

Dimensione delle tile di rasterizzazione in pixel. Il rendering Gaussian Splatting è basato su tile: l'immagine viene divisa in tile 16×16 pixel, ogni tile raccoglie le gaussiane rilevanti per essa, le ordina per profondità e le compone. 16 è lo standard usato praticamente da tutte le implementazioni 3DGS ed è hard-coded nei kernel Metal di RadianceKit; una modifica di questo valore richiederebbe la ricompilazione degli shader e non è effettiva nello stato attuale. Rimane come campo, nel caso una futura versione del motore supporti dinamicamente la tile size.

 **IN PAROLE SEMPLICI**

Parametro di rendering interno. Standard 16, non modificare.

**T25** throttleDelayMs **DETTAGLI**

**Default:** 0 (inizializzatore, `.full`, MCMC, Scene-Class), 0 (`.preview`) **Range:** 0 – 100 **Defined in:**

 **TECNICO**

Ritardo artificiale tra iterazioni di training in millisecondi. 0 = piena velocità (standard). Valori più alti rendono il Mac più «usabile» durante il training, dando a GPU/CPU regolari pause di respiro — l'usabilità di altre app aumenta, il tempo di training però linearmente con il ritardo. Valori tipici: 1–2 ms (throttling «leggero», +5% di tempo training, il Mac si sente più reattivo), 5 ms («medio», +15% di tempo training), 10+ ms («Eco», potenzialmente tempo training doppio). Viene offerto nell'Inspector sotto «Performance», ma non è nella vista standard — vedi backlog `dev_ux-backlog.md`, che propone di rimuoverlo dalla vista esperto, perché frainteso prolunga drammaticamente il tempo di training.

 **IN PAROLE SEMPLICI**

Quanti millisecondi di pausa l'app fa tra step di training. 0 = nessuna pausa, il più veloce possibile. Valori più alti rendono il Mac più usabile durante il training — ma il training dura anche più a lungo. Su un M3 Ultra o Mac Studio puoi lasciare a 0; su un MacBook Air 2 o 5 sarebbe un buon valore.

## Diagnosi e preparazione della nuvola di punti (T26–T30)

### T26 depthDistortionWeight

#### DETTAGLI

**Default:** 0.0 (= disabilitato) **Range:** 0 oppure 0.0001 – 0.05 **Defined in:**

#### TECNICO

Sperimentale V366: peso di un loss di regolarizzazione di depth distortion. Penalizza le gaussiane che sono stratificate in profondità lungo un raggio di render ma concettualmente appartengono alla stessa superficie — ciò incoraggia distribuzioni di profondità concentrate e riduce i floater. Test: 0.01 → 4.5% peggiore, 0.001 → 8.1% peggiore. Il vantaggio teorico — migliorare la consistenza multi-view — non si riflette nel loss L1, perché l'ipotesi assume implicitamente che la geometria SfM sia corretta e le gaussiane debbano solo essere «impilate». In pratica la nuvola di punti SfM è di solito il componente più debole, non l'impilamento. Rimane disponibile per dataset multi-view con pose particolarmente pulite (sintetici, Mip-NeRF 360 con ground truth).

#### IN PAROLE SEMPLICI

Funzione sperimentale per evitare più splat uno dietro l'altro nello stesso punto. Non attivato, perché i test non hanno portato nulla.

### T27 singleViewOverfit

#### DETTAGLI

**Default:** false **Range:** boolean **Defined in:**

#### TECNICO

Flag diagnostico: se true, in ogni iterazione di training viene forzatamente usato l'indice di fotocamera 0 invece di uno a caso dal pool. Senso: se il modello non riesce a fare overfitting nemmeno su una singola view (cioè il loss su view 0 anche dopo 10 000 iterazioni non va a zero), nel forward/backward pass c'è un bug fondamentale. Questo switch è stato usato intensamente durante lo sviluppo degli shader Metal e dei kernel del rasterizzatore differenziabile — fase V42–V47. Oggi disponibile solo come sanity check, se qualcuno ha modificato il codice backend e vuole fare un regression test. Via CLI con `--single-view`.

#### IN PAROLE SEMPLICI

Modalità di test per sviluppatori. Possono verificare con essa se l'app può in generale imparare da UNA sola immagine. Per utenti normali irrilevante, sempre lasciare off.

**T28 maxCameras** **DETTAGLI**

**Default:** 0 (= «usare tutte le fotocamere») **Range:** 0 oppure 1 – N **Defined in:**

 **TECNICO**

Limite diagnostico da V43: addestra solo con le prime N fotocamere, ignora tutte le altre. Senso originale: testare l'ipotesi che troppe fotocamere generino conflitti di gradiente (troppi segnali di loss contraddittori per la stessa gaussiana). Risultato del test: nessun vantaggio sistematico con limitazione artificiale — più frame portano praticamente sempre più qualità. Rimane come flag CLI (`--max-cameras N`) per esperimenti mirati, ad es. «il training funziona sulle prime 100 immagini di un volo di drone di 1 500 immagini?» Non esposto nell'UI.

 **IN PAROLE SEMPLICI**

Campo diagnostico per sviluppatori — usare solo le prime N immagini, ignorare il resto. Utente normale non ne ha bisogno, valore a 0 = tutte le immagini. Più immagini = risultato migliore (vedi `feedback_more-frames-better.md`).

**T29 maxInitialPoints** **DETTAGLI**

**Default:** 0 (= «usare tutti i punti SfM») **Range:** 0 oppure 1 000 – 200 000+ **Defined in:**

 **TECNICO**

Salvaguardia V54: limita il numero di punti SfM iniziali con cui inizia il training. Le ricostruzioni COLMAP dense possono produrre > 60 000 punti, il che con grandi scale iniziali porta a 200–300 gaussiane per overlap di pixel — ciò crea un «campo di nebbia» in cui il training non converge. Il sottocampionamento a ~16 000 punti (logica di hard cap nel motore di training) porta la densità iniziale al livello che il 3DGS di riferimento usa, e riduce drammaticamente l'overlap. Viene impostato automaticamente con SfM molto densi; via CLI con `--max-points N`.

 **IN PAROLE SEMPLICI**

Quanti punti iniziali dalla ricostruzione delle fotocamere vengono usati. Con ricostruzioni molto dense (oltre 60 000) l'app limita automaticamente a 16 000 — altrimenti c'è troppa nebbia all'inizio. Non hai bisogno di impostarlo; l'app lo regola.

**T30** cameraClusterOutlierMultiplier **DETTAGLI**

**Default:** 10.0 (tutti i preset — mai sovrascritto) **Range:** 1.0 – 100.0 **Defined in:**

 **TECNICO**

Moltiplicatore per il camera cluster outlier filter, introdotto in fase 3.10 A.1. Prima del training, il motore di training calcola il centroide di tutte le posizioni delle fotocamere e la distanza massima di una fotocamera dal centroide. I punti SfM la cui distanza dal centroide supera  $\text{multiplier} \times \text{maxCameraDistance}$  vengono scartati come outlier. Default 10x preserva il comportamento prima della fase 3.10. Un bug sottile: SfM tighter (fotocamere più strette insieme) → maxCameraDistance più piccolo → soglia più piccola → più punti scartati come outlier. SfM looser → soglia più grande → meno punti scartati. Questa è una delle cause dell'anti-correlazione funnel-vs-training della fase 3.9: un SfM migliore può portare downstream a un training peggiore, perché vengono uccisi troppi punti iniziali. Il campo è disponibile come override CLI (`--camera-cluster-outlier-multiplier`) per gli sweep A.3; non esposto nell'UI. Valori sotto 5 sono di solito troppo restrittivi, sopra 20 inefficaci.

 **IN PAROLE SEMPLICI**

Filtro speciale che scarta punti dalla ricostruzione lontani dalla nuvola della fotocamera. 10 = l'app è generosa, tiene quasi tutto. Aumentare può essere sensato se punti lontani (montagne sullo sfondo) appaiono nell'immagine come grumi fluttuanti. Abbassare solo in caso di emergenza — perdi così dettaglio in lontananza.

## Regolarizzazione (T31–T37)

### T31 coarseToFineBlurRadius

#### DETTAGLI

**Default:** 0 (= disabilitato) **Range:** 0 oppure 1 – 10  
**Defined in:**

#### TECNICO

Sperimentale V369: raggio di box blur che viene applicato all'immagine ground truth all'inizio della fase di densification e ridotto linearmente fino alla fine della densification (  $T_2$  ) a 0. Ipotesi: training coarse-to-fine — prima imparare strutture grossolane, poi dettagli — dovrebbe fornire geometria più stabile. Test:  $r=3 \rightarrow 9.6\%$  peggiore,  $r=1 \rightarrow 5.1\%$  peggiore. Il motivo del fallimento: la densification decide basandosi sui gradienti nel dominio dell'immagine, e il blur riduce esattamente i segnali importanti per «qui deve essere clonato». Rimane nel catalogo per futuri test con schema di density control diverso.

#### IN PAROLE SEMPLICI

Modalità sperimentale «prima grossolano, poi dettagliato». Non ha portato nulla, rimane off.

### T32 scaleRegWeight

#### DETTAGLI

**Default:** 0.0 (= disabilitato) **Range:** 0 oppure 0.0001 – 0.05 **Defined in:**

#### TECNICO

Sperimentale V370: regolarizzazione L1 su scala spazio mondo. Penalizza gaussiane che diventano troppo grandi — impedisce «mega splat» che coprono intere superfici di parete con una singola gaussiana. Test: 0.01  $\rightarrow$  200% di loss peggiore (2 M gaussiane, esplosione totale), 0.001  $\rightarrow$  214% peggiore. Il motivo: la regolarizzazione di scala entra in conflitto con il density control — scale più piccole significano che servono più gaussiane, quindi il density control splitta più spesso, il che significa più sforzo di gradiente. Disabilitato, ma documentato per esperimenti Mip-Splatting (T74): in questo contesto un limite inferiore di scala potrebbe essere sensato.

#### IN PAROLE SEMPLICI

Regolarizzazione che costringe gli splat a rimanere piccoli. Nei test ha provocato esplosioni di splat (milioni di splat). Non attivare.

**T33 anisotropyRegWeight** **DETTAGLI**

**Default:** 0.0 (= disabilitato) **Range:** 0 oppure 0.0001 – 0.05 **Defined in:**

 **TECNICO**

Sperimentale V445: penalità sul rapporto  $\max(\text{scale})/\min(\text{scale})$ , dovrebbe impedire gaussiane «ago» estreme allungate, che vengono percepite come floater. Test: 0.01 → 69% peggiore, 0.001 → 15% peggiore. Il motivo: la regolarizzazione costringe gli splat verso forma «rotonda», il che su una superficie piatta (parete, tavolo, pavimento) è esattamente sbagliato — lì una gaussiana piatta e larga è più efficiente di una sferica. Disabilitato. V549f ha offerto con T34 `scaleRatioPruneThreshold` un approccio alternativo più mirato, anch'esso revertito.

 **IN PAROLE SEMPLICI**

Regolarizzazione che punisce gli splat troppo lunghi e sottili. Suona sensato, ma nei test era peggio. Off.

**T34 scaleRatioPruneThreshold** **DETTAGLI**

**Default:** 0.0 (= disabilitato) **Range:** 0 oppure 5.0 – 100.0 (tipico 10.0 – 30.0) **Defined in:**

 **TECNICO**

Pruning post-training sperimentale che elimina ogni gaussiana il cui rapporto  $\max(\text{scale})/\min(\text{scale})$  supera la soglia lineare qui impostata. Mira a floater «ago/disco» estremamente allungati che non possono essere eliminati con la sola regolarizzazione. Nel test il pruning rimuoveva i floater come sperato, ma contemporaneamente anche splat piatti sensati su pareti e pavimenti — l'immagine è diventata più bucata. Quindi di default off, il flag CLI (`--scale-ratio-prune N`) rimane disponibile per esperimenti mirati. Valori raccomandati se si vuole comunque testare: 30 (molto conservativo, rimuove solo outlier estremi), 10 (aggressivo, costa dettaglio).

 **IN PAROLE SEMPLICI**

Tentativo di filtrare gli splat molto allungati dopo il training. Nei test era netto-negativo — floater via, ma anche dettaglio via. Off.

**T35** `opacityRegWeight` **DETTAGLI**

**Default:** 0.0 (= disabilitato) **Range:** 0 oppure 0.0001 – 0.05 **Defined in:**

 **TECNICO**

Sperimentale V446: penalità di binary cross entropy che tira l'opacità verso 0 o 1 (quindi via da «semi trasparente»). Ipotesi: distribuzione di opacità più nitida migliorerebbe la chiarezza dell'immagine. Test combinato con T33 → la regolarizzazione costa qualità, entrambi disabilitati. Disabilitato. Attenzione: nella 1.4.3 beta è emerso un bug che aveva esattamente questo campo in una modifica del valore di default (inizializzatore = 0.01), il che ha portato a mass extinction del gaussian count (460 K → 5 in un'iterazione). Dalla 1.4.4 fissato a 0.0 come default.

 **IN PAROLE SEMPLICI**

Regolarizzazione che rende gli splat o completamente trasparenti o completamente solidi. Non porta nulla, può anche diventare pericoloso (bug mass extinction 1.4.3). Lascia a 0.

**T36** `opacityDecayFactor` **DETTAGLI**

**Default:** 0.0 (inizializzatore = disabilitato), 0.9995 (`.full`, `.classicBalanced` — standard HTGS) **Range:** 0 (off) oppure 0.95 – 1.0 **Defined in:**

 **TECNICO**

Implementazione V546 dello schema HTGS (Hierarchical Time-Gating, Eurographics 2025): ogni `T37 opacityDecayInterval` iterazioni l'opacità sigmoid di ogni gaussiana viene moltiplicata con questo fattore.  $0.9995 \times 100$  applicazioni dà ~95% di rimanenza per fase di densification — una leggera ma costante pressione verso il basso su tutte le opacità, che fa scendere in modo affidabile le gaussiane che contribuiscono poco contro la `T14 pruneOpacityThreshold`. Il risultato: 14% di loss L1 migliore su Horse Full (3-trial-avg V546) rispetto a V438 senza decay. Attivo solo durante la fase di densification (fino a `T2`), dopo il training prosegue senza decay, in modo che le opacità stabilite nel refinement rimangano stabili. Con MCMC non usato (MCMC ha propri meccanismi via `T67 mcmcRelocationInterval`).

- `T68 mcmcDeadOpacityThreshold`).

 **IN PAROLE SEMPLICI**

«Sbiadimento dolce» di tutti gli splat sul tempo di training. Rende gli splat diventati inattivi più velocemente trasparenti, in modo che spariscano alla pulizia. È stata la leva di qualità più importante dell'update V546: 14% migliore. Integrata nel preset Quality. Non raccomandata per la regolazione da soli, perché esattamente bilanciata.

**T37** opacityDecayInterval DETTAGLI**Default:** 50 **Range:** 10 – 500 **Defined in:** TECNICO

Intervallo di iterazione in cui viene applicato T36 opacityDecayFactor. Default paper HTGS 50, lasciato in `.full`. Intervalli lunghi (>200) annullano parzialmente l'effetto, perché tra due applicazioni avvengono abbastanza update di gradiente che l'opacità risale. Intervalli più brevi (<20) rendono il decay troppo aggressivo. Attivo solo in fase di densification.

 IN PAROLE SEMPLICI

Con quale frequenza viene applicato lo «sbiadimento». 50 = ogni 50 iterazioni un piccolo step di sbiadimento. Va bene.

**Refinement (T38–T44)****T38** gradientAccumulationSteps DETTAGLI**Default:** 1 (= «una view per step Adam») **Range:** 1 – 8 **Defined in:** TECNICO

Funzione V424: numero di view i cui gradienti vengono accumulati prima che venga eseguito un update Adam. Con `> 1` l'app gira su un percorso backward project separato e «unfused» che somma i gradienti in un buffer separato; l'applicazione finale scala con  $1/N$  per mantenere costante la magnitudine. V424 ha testato 2-view → neutro in qualità, ma 10% più lento (perché il percorso unfused è più costoso del fused). Revertito per `.full`, ma per MCMC deliberatamente usato — `.fullMCMC` gira con, ma i test V544a hanno mostrato che con esso il gap di qualità verso Classic si riduce al 5% (invece dell'11%). Nel default dell'inizializzatore 1, nel preset attuale 1, rimane flag CLI (`--accum-steps N`).

 IN PAROLE SEMPLICI

Quante immagini l'app guarda prima di adattare gli splat. 1 = ogni immagine singolarmente. Più alto = guardare più immagini contemporaneamente e poi applicare una media. Nel caso standard non porta nulla; con MCMC 2 può aiutare un po'.

**T39** testViewIndices **DETTAGLI**

**Default:** `[]` (= vuoto, tutte le view vengono usate per il training) **Range:** `Set<Int>`, qualsiasi sottoinsieme degli indici di fotocamera **Defined in:**

 **TECNICO**

Funzione V546: set di indici di fotocamera che NON vengono usati per il training, ma riservati come holdout per la valutazione PSNR/SSIM/LPIPS. Viene impostato automaticamente quando il flag CLI `--benchmark` è attivo: allora ogni ottava view, iniziando dall'indice 0 (standard LLFF, identico alle convenzioni Mip-NeRF-360 e paper 3DGS). Senza benchmark vuoto — il training usa tutte le view. **Attenzione:** l'impostazione manuale di questo campo senza comprensione degli indici può rendere inutilizzabile il benchmark (ad es. se tutti gli indici sono impostati sopra N, mentre ci sono solo N-50 view → nessun holdout → nessuna valutazione). Al proprio export preset testViewIndices non viene persistito, perché è dipendente dalla scena e altrimenti lascerebbe valori senza senso tra diversi dataset.

 **IN PAROLE SEMPLICI**

Quali immagini vengono «salte» al training, per usarle in seguito per la misurazione della qualità. Non lo imposti tu stesso; il flag `--benchmark` lo fa automaticamente (ogni ottava immagine è test). Se imposti indici tuoi: pericoloso, può falsare il benchmark.

**T40** refinementPruneInterval **DETTAGLI**

**Default:** 0 (= disabilitato) **Range:** 0 oppure 100 – 5 000 **Defined in:**

 **TECNICO**

Funzione V425: ogni N iterazioni durante la fase di refinement (dopo `T2`) viene eseguito un pass di prune aggiuntivo, che rimuove le gaussiane con `sigmoid(opacity) < T41 refinementPruneOpacityThreshold`. Senso: durante la densification ci sono regolari chiamate di density control, dopo non più — le gaussiane la cui opacità continua a scendere rimangono però nel buffer. V425 ha testato e revertito: il pruning aggiuntivo correlava con V426 (Two-Phase Densification, anch'esso interrotto in 0-gaussian cascade failure). Disabilitato. Flag CLI disponibile per esperimenti; se attivato, 1 000 o 2 000 sono valori sensati.

 **IN PAROLE SEMPLICI**

Pulizia aggiuntiva durante la fase di refinement. Non porta nulla, rimane off.

**T41 refinementPruneOpacityThreshold** **DETTAGLI**

**Default:** 0.0 (= «usa T14») **Range:** 0 oppure 0.001 – 0.1 **Defined in:**

 **TECNICO**

V425b: soglia di opacità separata per refinement prune. Dopo la densification la maggior parte delle gaussiane ha raggiunto un'opacità decisamente più alta (> 0.001), quindi lo standard T14 pruneOpacityThreshold sarebbe troppo lasco. Se T40 attivo, questo campo determina la propria soglia. A 0.0 viene continuato l'uso di T14. Rilevante solo se T40 > 0.

 **IN PAROLE SEMPLICI**

Soglia per la pulizia aggiuntiva di refinement (vedi T40). Entrambi i campi non attivi, quindi irrilevanti.

**T42 midTrainingCompactificationIterations** **DETTAGLI**

**Default:** [] (= disabilitato) **Range:** [Int], valori in (densifyUntilIteration, maxIterations) **Defined in:**

 **TECNICO**

Funzione V549: punti di iterazione espliciti durante la fase di refinement, in cui viene eseguito un pass di compactification (rimuove sigmoid(opacity) < 0.01 + gaussiane di scala outlier, stessa logica di T56 postTrainingCompactification). Senso: lunghe fasi di refinement possono mostrare accumulo di confetti/floater, il cui SH poi va in overfitting su artefatti specifici della view. Configurazione tipica se attivata: [10000, 20000, 30000] per 40K Classic. **MA:** i test V549 A/B sul dataset Family hanno mostrato in tutte le configurazioni un L1 peggiore: [10K, 20K, 30K]@0.01 → -48% count ma +36% L1; [20K, 30K]@0.005 → -44% count ma +45% L1; [20K, 30K]@0.001 → -17% count ma +87% L1. Quindi disabilitato. Flag CLI --mid-compact "10000,20000" disponibile, se si preferisce il tradeoff floater visivo (meno confetti nel viewport) rispetto alla regressione di loss.

 **IN PAROLE SEMPLICI**

Azioni di pulizia intermedie durante il training. Nei test la pulizia ha reso il risultato finale peggiore (sì meno floater, ma anche meno dettaglio). Off, può essere accesa via CLI se i floater ti danno più fastidio di un'immagine leggermente più pastosa.

**T43 frustumCullEnabled** **DETTAGLI****Default:** false **Range:** boolean **Defined in:** **TECNICO**

Funzione V549b: dopo il training vengono rimosse tutte le gaussiane che si trovano al di fuori dell'unione di tutti i frustum delle fotocamere di training. Tali gaussiane non sono mai state limitate dal segnale di loss e sono sempre floater. Particolarmente efficace per scene in cui la novel view si trova dietro o accanto al percorso della fotocamera (ad es. retro di un volo di drone lineare) — i floater lì non sono mai visibili nella fase di training, ma sì nello spostamento successivo nel viewer 3D. V549b A/B su voli di drone risultati positivi, quindi disponibile come opt-in. Default false, perché con object capture con piena copertura orbit l'unione dei frustum comprende l'intera scena e la funzione non rimuove nulla — viene offerta nelle Settings sotto «Floater Reduction» e testata anche in Q9 Outdoor preset implicitamente tramite T44

`frustumCullExpansion` (Q7-BayesOpt però non l'ha attivata, perché Outdoor Sky Dome risolve meglio lo stesso problema).

 **IN PAROLE SEMPLICI**

Filtro speciale per voli di drone o riprese lineari: dopo il training vengono eliminati gli splat che non sono stati «visti» in nessuna fotocamera. Opzionalmente attivabile nelle Settings. Per semplici riprese di oggetto inutile.

**T44 frustumCullExpansion** **DETTAGLI****Default:** 1.1 **Range:** 1.0 – 2.0 **Defined in:** **TECNICO**

Margine NDC per T43 `frustumCullEnabled`. 1.0 taglierebbe esattamente al bordo dell'immagine, il che taglierebbe troppo splat traballanti al bordo dell'immagine. 1.1 = 10% di padding oltre l'esatto framing della fotocamera — dà un po' di tolleranza per pixel di bordo che in una novel view leggermente spostata potrebbero diventare visibili. Valori > 1.2 rendono il cull praticamente inefficace, perché il frustum espanso comprende molto più spazio.

 **IN PAROLE SEMPLICI**

Quanto rigorosamente il filtro descritto sopra taglia. 1.1 = un po' di distanza di sicurezza dal bordo dell'immagine. Lascia il valore.

## Sky-Dome (T45–T48)

### T45 skyDomeEnabled

#### DETTAGLI

**Default:** false (inizializzatore + tutti i preset eccetto P9 Outdoor) **Range:** boolean **Defined in:**

#### TECNICO

Funzione V549e: prima dell'avvio del training viene generata una nuvola di punti sferica (sfera di Fibonacci con T46 sample point), posizionata in un raggio di T47 `skyDomeRadiusMultiplier` × `scene_extent` attorno al centro della scena e inizializzata con i colori dei pixel mascherati come cielo di tutte le fotocamere di training (vedi T20 `skyMaskingEnabled`). Queste gaussiane sky dome vengono inserite all'inizio del buffer gaussiano e «congelate» durante il training (gradienti posizione/scala/rotazione = 0, solo SH e opacità rimangono ottimizzabili). Effetto: invece di aree «confetti» nere in lontananza, l'utente vede in novel view un cielo reale. MVP V549e funziona molto bene su scene di drone e paesaggio; in P9 Outdoor preset default on. Con scene interne lasciare off — la sfera penderebbe inutile fuori dalla stanza.

#### IN PAROLE SEMPLICI

Accende una «cupola del cielo» artificiale attorno alla scena. Rende le riprese esterne molto più belle: invece di grumi neri al bordo dell'immagine, l'app mostra il cielo reale. Obbligatorio per voli di drone e paesaggi, inutile per interni.

**T46** skyDomeSampleCount **DETTAGLI**

**Default:** 5 000 **Range:** 1 000 – 50 000 (tipico 2 000 – 10 000) **Defined in:**

 **TECNICO**

Numero di sample point della sfera di Fibonacci sulla sfera sky dome. Valori più alti → sky dome più denso (migliore con grandi risoluzioni e molto cielo visibile), ma più fabbisogno di memoria. 5 000 è sweet spot per rendering 4K; con risoluzioni più basse 2 000–3 000 basta. I punti vengono inizializzati per cosine distance a ogni vettore di vista delle fotocamere di training con i corrispondenti pixel mascherati come cielo — i sample point il cui view cone non vede alcuna fotocamera rimangono indietro con un basso valore iniziale di opacità, ma non vengono modificati durante il training (congelati).

 **IN PAROLE SEMPLICI**

Quanto denso è il cielo artificiale. 5 000 punti bastano normalmente. Di più = transizione migliore in lontananza, ma costa un po' di memoria.

**T47** skyDomeRadiusMultiplier **DETTAGLI**

**Default:** 30.0 (inizializzatore + la maggior parte dei preset), 59.0 (P9 Outdoor, ottimo Q7-BayesOpt) **Range:** 5.0 – 200.0 **Defined in:**

 **TECNICO**

Raggio della sfera sky dome relativo all'estensione della scena (= distanza media tra posizioni delle fotocamere). 30 = la sfera ha 30 volte il diametro della nuvola della fotocamera. Troppo piccolo (< 5) → lo sky dome interferisce con la scena stessa (ad es. uno splat sky dome finisce in primo piano); troppo grande (> 100) → perdita di precisione float32 sulle posizioni sky dome, il che provoca glitch di rendering in lontananza. Q7-BayesOpt su Bicycle (Mip-NeRF 360) ha trovato 59.0 come ottimo specifico per scena outdoor — ciò indica che lo standard 30.0 è troppo piccolo per paesaggi profondi e i pixel sky dome nelle aree di bordo immagine renderizzano visibilmente come «parete».

 **IN PAROLE SEMPLICI**

Quanto lontana deve essere la cupola del cielo artificiale. 30 = abbastanza lontano. Con grandi paesaggi 50–60 è meglio (Outdoor preset lo fa automaticamente). Troppo piccolo sarebbe come avere grumi direttamente davanti all'obiettivo.

**T48 frozenGaussianCount** **DETTAGLI**

**Default:** 0 (= nessuna gaussiana congelata) **Range:** 0 oppure 1 – **T46** **Defined in:**

 **TECNICO**

Numero di gaussiane all'inizio del buffer i cui gradienti posizione/scala/rotazione vengono impostati a zero nell'optimizer — rimangono spazialmente rigide su tutto il training. Il density control non può clonarle, splittarle o prunarle. Usato per l'iniezione sky dome (vedi **T45**): se sky dome è on, questo campo viene impostato automaticamente su **T46** skyDomeSampleCount. L'impostazione manuale è possibile (ad es. per congelare una nuvola di punti pre-posizionata da una scansione LiDAR), ma non direttamente accessibile nell'UI. Importante: le prime N gaussiane nel buffer sono sempre le frozen — l'ordine nel buffer decide, non un indice esplicito.

 **IN PAROLE SEMPLICI**

Quanti splat all'inizio sono fissi e non possono muoversi. Viene impostato automaticamente sul numero sky dome, se sky dome è on. Non hai bisogno di regolarlo da solo.

**Adam + LR-Schedule (T49–T55)****T49 adamResetIteration** **DETTAGLI**

**Default:** 0 (= disabilitato) **Range:** 0 oppure 100 – **Defined in:**

 **TECNICO**

Funzione V430: iterazione a cui gli accumulatori di momentum Adam (m1, m2) vengono resettati a zero. La correzione del bias successivamente gira con (iter - adamResetIteration) invece di iter. V430 ha testato reset a 5 000 (dopo fine densification) → 12.8% di loss peggiore. Motivo: il momentum Adam che si è costruito durante la densification porta informazioni sulle magnitudini tipiche di gradiente e accelera la fase di refinement. Gettarlo via costa le prime ~500 iterazioni di refinement in convergenza. Disabilitato. Rimane flag CLI per esperimenti di ricerca.

 **IN PAROLE SEMPLICI**

Pulsante reset per la «memoria» interna dell'ottimizzatore Adam. Nei test ha danneggiato, rimane off.

**T50 positionLRScheduleEndIteration** **DETTAGLI**

**Default:** 0 (inizializzatore = «usa maxIterations»), 20 000 ( `.full` — cosine termina a 20K nonostante `maxIter=35K`), 30 000 ( `.fullClassicPaper` ) **Range:** 0 oppure 1 000 – **Defined in:**

 **TECNICO**

Funzione V431: iterazione a cui la curva di cosine annealing per la LR di posizione raggiunge il suo minimo. Se 0, è identico a `T1 maxIterations`. Se  $> 0$ , lo schedule corre fino a questo valore e dopo rimane a `T4`

`positionLearningRateFinal` costante. Ciò permette una «extended refinement phase» con LR minima ma costante — raffina lentamente le posizioni senza nuovo decay. `.full` lo fa (schedule end a 20K, il training corre fino a 35K), V434c/V434d hanno confermato: 15K e 25K entrambi più o meno uguali, 20K minimamente ottimale. Continua ad essere usato in combinazione con `T51`, per modificare nella extended phase anche le LR non di posizione.

 **IN PAROLE SEMPLICI**

Quando l'app smette di abbassare ulteriormente la LR di posizione. Se più basso dell'iterazione massima, gira dopo con mini rate costante — ciò raffina molto lentamente ma molto stabile. Integrato in Quality preset, non hai bisogno di regolarlo.

**T51 extendedPhaseLRDecay** **DETTAGLI**

**Default:** 0.0 (= disabilitato, LR costanti) **Range:** 0 oppure 0.01 – 1.0 **Defined in:**

 **TECNICO**

Funzione V433: moltiplicatore minimo per le LR non di posizione (scala, rotazione, opacità, SH) nella «extended phase» — cioè dopo che `T50` è raggiunto e la LR di posizione è già a `T4`. Se 0.1, scala/rotazione/opacità/SH vengono a loro volta cosine-decayed da 1.0 (= la loro LR standard) a 0.1× del loro standard. Se 0.0 (default), rimangono costanti. V457 ha testato decay completo (0.0 = decay-fino-a-zero) contro nessun decay e ha trovato: avg 0.0400 (2 run) = stesso loss di V438 senza decay. Comportamento più pulito con decay, ma non misurabilmente migliore. Quindi disabilitato. Rimane nel CLI come `--nonpos-lr-scale F`.

 **IN PAROLE SEMPLICI**

Nella fase tardiva di refinement anche le learning rate di colore e forma più piccole. Rende il training più «stabile», ma empiricamente non migliore. Off.

**T52** adaptiveDensifyThreshold **DETTAGLI****Default:** false **Range:** boolean **Defined in:** **TECNICO**

Sperimentale V440: se true, l'app calcola in ogni step di densification il p98 dell'attuale distribuzione del gradiente e lo usa come soglia dinamica (limitata ad almeno 0.5x del valore configurato da `T11`, in modo che non sbordi troppo). Ipotesi: l'adattamento automatico alla fase di scena attuale renderebbe il density control più robusto — ad es. all'inizio pruning più severo, dopo più lasco, o viceversa. V440 ha testato e revertito: drop catastrofico a 63 K gaussiane (mass pruning, perché il p98 nelle prime iterazioni è estremamente alto e poi quasi nulla supera la soglia). La soglia fissa è già ben calibrata, l'adattamento dinamico danneggia più di quanto giovi. Q5 (T77) offre una logica adattiva alternativa via rolling median che aggira il problema.

 **IN PAROLE SEMPLICI**

Versione adattiva della soglia di densify. Nei test catastrofica (numero di splat crollato a 63K). Off. Q5 ne ha una variante migliore.

**T53** mergeAfterDensification **DETTAGLI****Default:** false (inizializzatore), true ( `.full` , `.classicBalanced` , `.fullClassicPaper` ) **Range:** boolean **Defined in:** **TECNICO**

Funzione V438: alla fine della fase di densification (iter `T2` ) viene eseguito un pass di merge una tantum, che unisce gaussiane vicine tra loro con scala e colore simili. Riduce il numero di gaussiane tipicamente del 5–15% senza perdita di qualità visibile. Senso: dopo un intenso clonaggio nascono cluster di gaussiane quasi identiche, che non contribuiscono nulla di nuovo — il merging libera capacità optimizer per altre aree. Standard nei preset Classic Quality. Con MCMC non usato, perché MCMC tramite la sua logica di rilocazione non lascia nemmeno nascere tali cluster.

 **IN PAROLE SEMPLICI**

Alla fine della fase di moltiplicazione degli splat unire i cloni che sono quasi identici. Riduce la quantità di dati senza effetto visibile. Standardmente on nel preset Quality.

**T54** densifyPhase2FromIteration **DETTAGLI**

**Default:** 0 (= disabilitato) **Range:** 0 oppure T2 – T1 **Defined in:**

 **TECNICO**

Sperimentale V426: abilita una seconda fase di densification che inizia dopo la pausa di refinement a questa iterazione e corre fino a T55. Ipotesi: dopo una fase di refinement gli accumulatori di gradiente hanno magnitudini più stabili e possono dire più precisamente quali aree hanno bisogno di gaussiane aggiuntive. V426 ha testato e revertito: densification a due fasi è caduta in 0-gaussian-cascade-failure (combinata con V425 refinement pruning ha distrutto il buffer). Disabilitata. Flag CLI disponibile per esperimenti.

 **IN PAROLE SEMPLICI**

Seconda fase di moltiplicazione dopo pausa. Nei test ha distrutto la popolazione di splat. Off.

**T55** densifyPhase2Untilliteration **DETTAGLI**

**Default:** 0 **Range:** 0 oppure T54 – T1 **Defined in:**

 **TECNICO**

Fine della Two-Phase Densification V426. Rilevante solo se T54 > 0. Entrambi i campi insieme disabilitati.

 **IN PAROLE SEMPLICI**

Fine della seconda fase di moltiplicazione (vedi T54). Entrambi off.

## Post-processing + Apple AI (T56–T60)

### T56 postTrainingCompactification

#### DETTAGLI

**Default:** true (in tutti i preset di produzione), false ( `.quickTest`, `.preview` ) **Range:** boolean **Defined in:**

#### TECNICO

Funzione V443: dopo la fine del training le gaussiane con  $\text{sigmoid}(\text{opacity}) < 0.01$  vengono rimosse duramente (non contribuiscono praticamente più all'immagine). Riduce il gaussian count tipicamente del 58% e la dimensione del file di export del 55% senza perdita di qualità visibile. Attiva di default nei preset di produzione — il risultato finale dovrebbe poter essere consegnato il più compatto possibile. In `.quickTest` off, perché un run diagnostico non viene comunque esportato. A differenza di T42 `midTrainingCompactificationIterations` (V549) la compactification avviene solo alla fine — il refinement può usare fino ad allora tutte le gaussiane.

#### IN PAROLE SEMPLICI

Pulizia dopo il training: gli splat quasi invisibili vengono rimossi. Rende il file di export circa la metà più piccolo senza perdita di qualità. Funzione obbligatoria, off solo nei run diagnostici.

### T57 metalFXUpscaling

#### DETTAGLI

**Default:** false **Range:** boolean **Defined in:**

#### TECNICO

Funzione V444: attiva l'upscaler spaziale MetalFX di Apple invece dell'interpolazione bilineare nell'output del viewer 3D. Se la risoluzione di training  $<$  dimensione viewport (ad es. training a 0.5x, visualizzazione viewport in piena risoluzione), MetalFX può fornire un'immagine decisamente più nitida. Cambia live nel viewport, nessun re-training necessario. Si esclude con T58 `mpsLanczosScaling` — MetalFX ha la precedenza. Raccomandazione: accendere se l'immagine nel viewer appare «sbiadita» rispetto al dettaglio atteso.

#### IN PAROLE SEMPLICI

Affinamento dell'immagine basato su ML Apple nel viewer 3D. Aiuta se hai addestrato a una risoluzione più bassa e mostri il risultato a schermo intero. Toggle dal vivo, prova.

**T58** mpsLanczosScaling **DETTAGLI**

**Default:** false **Range:** boolean **Defined in:**

 **TECNICO**

Funzione V444: MPSImageLanczosScale per lo scaling viewport invece dell'interpolazione bilineare. Lanczos è un metodo di ricampionamento classico basato su sinc che fornisce risultati decisamente più nitidi del bilineare con overhead minimo. Toggle dal vivo. Viene sovrascritto da `T57` se entrambi attivi.

 **IN PAROLE SEMPLICI**

Metodo di affinamento classico per il viewer 3D (Lanczos). MetalFX (T57) è basato su ML e di solito migliore; Lanczos è un'alternativa meno aggressiva.

**T59** livePreviewInterval **DETTAGLI**

**Default:** 50 (inizializzatore e la maggior parte dei preset) **Range:** 0 (off) oppure 10 – 5 000 **Defined in:**

 **TECNICO**

Quanto spesso durante il training il viewer 3D viene aggiornato con le gaussiane attuali. 50 = ogni 50 iterazioni un nuovo render nel viewer — abbastanza per osservare il progresso senza rallentare il training. 0 = il viewer non viene aggiornato affatto (training in background, max velocità). Adattamento tipico: con `.quickTest` giù a 10 (si vuole vedere ogni step), con lunghi run MCMC su a 500–2000 (overhead di update in totale percettibile).

 **IN PAROLE SEMPLICI**

Con quale frequenza l'anteprima 3D viene aggiornata durante il training. 50 = ogni 50 iterazioni. Più alto = meno spesso = un po' più veloce, ma vedi meno spesso il progresso. 0 = nessuna anteprima (per la massima velocità).

**T60** perceptualLossWeight **DETTAGLI**

**Default:** 0.0 (= disabilitato) **Range:** 0 oppure 0.001 – 0.5 **Defined in:**

 **TECNICO**

Funzione futura V444: peso di un termine di loss percettivo via MPSGraph (piccola rete simil-VGG). Catturerebbe similarità strutturale e di texture a un livello semantico superiore a L1+SSIM — tipico in pipeline di ricerca, dove «pixel-perfect» è meno importante di «sembra realistico». Implementazione ancora in attesa (stub di codice presente, ma forward pass non implementato). Default 0.0. Rimane nel catalogo dei campi per attivazione futura; flag CLI `--percep-weight F` riservato.

 **IN PAROLE SEMPLICI**

Funzione pianificata che con aiuto AI mira ad «aspetto naturale» invece di «preciso al pixel». Non ancora completamente implementata.

**Densification MCMC (T61–T73)****T61** densificationStrategy **DETTAGLI**

**Default:** `.classic` (inizializzatore + preset Classic), `.mcmc` (tutti i preset MCMC + Scene-Class)  
**Range:** `.classic` o `.mcmc` **Defined in:**

 **TECNICO**

Sceglie tra densification classica (clone/split/prune, Kerbl et al. 2023) e densification MCMC (Stochastic Gradient Langevin Dynamics con rilocalazione, Kheradmand et al. NeurIPS 2024). Con `.classic` vengono valutati T11–T16, con `.mcmc` i T62–T73. Attenzione al cambio: i default Classic e i default MCMC sono calibrati in modo completamente diverso — chi flippa il picker nella vista esperto senza caricare un preset appropriato, rischia mass extinction in stile 1.4.3 bug (460 K → 5 in un'iterazione, perché MCMC OpacityReg a 0.01 uccide le opacità Classic). Quindi i default init MCMC deliberatamente «addolciti» (tutti i valori reg 0.0).

 **IN PAROLE SEMPLICI**

Quale algoritmo viene usato per moltiplicare gli splat. Classic = metodo originale (veloce, molti splat). MCMC = metodo più nuovo (più lento, molti meno splat, in compenso più compatto). I preset scelgono quello giusto. Cambialo da solo solo se carichi anche il preset corrispondente (P5–P7 o P8–P10).

**T62** mcmcMaxGaussians **DETTAGLI**

**Default:** 150 000 (inizializzatore + `.fullMCMC` + `.mcmcBalanced`), 100 000 (`.mcmcPreview`), 1 500 000 (`.fullMCMCMip` — variante Mip-Splatting con budget 10×), 1.19 M (`.renderPreset`), 1.25 M (`.outdoorPreset`), 670 K (`.indoorPreset`) **Range:** 0 (= «usa capacità buffer») oppure 10 000 – 5 000 000 **Defined in:**

 **TECNICO**

Limite superiore rigido per il numero di gaussiane con strategia MCMC. Il numero cresce gradualmente di `mcmcGrowthRate` (tipico 5%) per step di rilocalizzazione fino a questo cap. V473/V531 hanno trovato 150 K come sweet spot — oltre 200 K si dilata la qualità degli splat (troppe gaussiane piccole, ridondanti), sotto 100 K la scena rimane sotto-densificata. Con scene molto grandi (ad es. volo di drone da 1 545 foto con 158 K SfM-init) 150 K è troppo basso — da qui l'estensione 1.4.5 `mcmcCapMultiplier` + `mcmcAutoScaleByScene`. Q7-BayesOpt ha trovato ottimi specifici per scena tra 670 K (Indoor) e 1.25 M (Outdoor). Con valore 0 il motore usa la piena capacità buffer come cap.

 **IN PAROLE SEMPLICI**

Numero massimo di splat con MCMC. 150 000 è lo standard e basta per la maggior parte delle scene. Outdoor e Render preset (P8, P9) salgono a 1+ milione per scene più ricche di dettagli. Aumentare può portare dettaglio, costa memoria; abbassare è piuttosto un freno di emergenza.

**T63** **mcmcNoiseScale** **DETTAGLI**

**Default:** 0.00005 (5e-5 = default paper) **Range:** 1e-6 – 1e-3 **Defined in:**

 **TECNICO**

Moltiplicatore per il rumore gaussiano che in ogni iterazione MCMC viene aggiunto alla posizione di ogni gaussiana (logica SGLD). Più alto = più esplorazione (le gaussiane vagano di più, trovano potenzialmente posti migliori), più basso = più exploitation (le gaussiane rimangono dove sono già buone). V467 e V536 hanno confermato 5e-5 come ottimale — 1e-5/2e-5 troppo poca esplorazione, 1e-4 troppo (gli splat si dissipano). Viene cosine-decayed sul tempo di training fino a T69 `mcmcNoiseDecayEnd` — alla fine della zona di decay il rumore è effettivamente 0 e le gaussiane convergono.

 **IN PAROLE SEMPLICI**

Quanto «tremolio» casuale l'app permette agli splat, in modo che trovino da soli il posto migliore. Il valore standard è ottimalmente testato. Se lo aumenti, gli splat diventano irrequieti.

**T64** **mcmcOpacityRegWeight** **DETTAGLI**

**Default:** 0.0 (= disabilitato nei default RadiancKit, paper: 0.01) **Range:** 0 oppure 0.001 – 0.05 **Defined in:**

 **TECNICO**

Penalità L1 specifica MCMC sull'opacità. Default paper 0.01 (spinge le gaussiane non usate verso zero, le rende disponibili per rilocalizzazione). V464b ha mostrato però: senza reg in RadiancKit è misurabilmente migliore (sessione 28 conferma). Motivo: il criterio di pruning definito con T68 `mcmcDeadOpacityThreshold` basta da solo — una penalità L1 aggiuntiva costringe a morire anche gaussiane preziose a bassa opacità. Quindi default 0. **Attenzione:** nella build 1.4.3 beta il default dell'inizializzatore era erroneamente 0.01, il che è risultato nel mass extinction bug (vedi spiegazione T61); dalla 1.4.4 fissato a 0.0.

 **IN PAROLE SEMPLICI**

Regolarizzazione speciale MCMC. Off, perché l'altro meccanismo MCMC (soglia in T68) lo copre già. Lascia a 0.

**T65** **mcmcScaleRegWeight** **DETTAGLI**

**Default:** 0.0 (= disabilitato, paper: 0.01) **Range:** 0 oppure 0.001 – 0.05 **Defined in:**

 **TECNICO**

Penalità L1 specifica MCMC sugli autovalori di scala. Default paper 0.01. V464b: senza reg migliore, stesso ragionamento di T64. Disabilitato in tutti i preset MCMC di RadianceKit. Attenzione come per T64: bug 1.4.3.

 **IN PAROLE SEMPLICI**

Come T64, ma per dimensione splat. Off.

**T66** **mcmcRelocationInterval** **DETTAGLI**

**Default:** 100 (inizializzatore + tutti i preset MCMC, standard paper), 155 (P9 Outdoor — ottimo Q7-BayesOpt) **Range:** 50 – 500 **Defined in:**

 **TECNICO**

Intervallo di iterazione in cui MCMC riloca le gaussiane morte ( $\text{sigmoid}(\text{opacity}) < T68$  `mcmcDeadOpacityThreshold`) in nuove posizioni. V537 ha testato 50 (troppo dirompente, loss oscilla) e 200 (marginalmente peggio, MCMC perde reattività). 100 è ottimale. Q7-BayesOpt su Bicycle ha trovato 155 come ottimo specifico per scena outdoor — gli intervalli un po' più lunghi danno ad Adam più tempo per integrare le gaussiane nuove posizionate, prima che il prossimo evento di reloc le metta sotto pressione.

 **IN PAROLE SEMPLICI**

Ogni quante iterazioni MCMC sposta gli splat morti da qualche altra parte. 100 è standard. Non hai bisogno di regolarlo da solo — Outdoor preset ha già il valore ottimale.

**T67** **mcmcWarmupIterations** **DETTAGLI****Default:** 500 **Range:** 100 – 5 000 **Defined in:** **TECNICO**

Numero di iterazioni iniziali in cui non avviene ancora rilocalizzazione MCMC. Solo dopo questo warmup inizia la logica di reloc. Senso: nelle prime iterazioni i valori di opacità non sono ancora stabilizzati — se si iniziasse subito con reloc, le gaussiane verrebbero posizionate in punti sbagliati e dovrebbero essere subito spostate, il che distrugge il momentum Adam. Default paper 500. RadianceKit assume questo valore, perché V464b ha mostrato che è robusto.

 **IN PAROLE SEMPLICI**

Quante iterazioni MCMC ha prima di «arrivare», prima di iniziare a riposizionare gli splat. 500 è standard e va bene.

**T68** **mcmcDeadOpacityThreshold** **DETTAGLI****Default:** 0.005 (inizializzatore, standard paper), 0.01 (`.fullMCMC` e tutti i preset MCMC — ottimo V535) **Range:** 0.001 – 0.05 **Defined in:** **TECNICO**

Soglia sigmoid(opacity) sotto la quale una gaussiana è considerata «morta» ed entra in considerazione per rilocalizzazione. V535 ha trovato 0.01 come ottimale (0.005 marginale, 0.02 peggiore). Più alto = reloc più aggressivo (più gaussiane vengono spostate), più basso = più cauto. 0.01 corrisponde a circa «0.5% di visibilità visiva». P10 Indoor usa via Q7-BayesOpt 0.0142 come ottimo.

 **IN PAROLE SEMPLICI**

A partire da quale trasparenza uno splat è considerato «morto», così che MCMC lo sposti altrove. 0.01 è ottimale nei nostri test. Non hai bisogno di regolarlo da solo.

**T69** `mcmcNoiseDecayEnd` **DETTAGLI**

**Default:** 0 (inizializzatore = «nessun decay»), 160 000 ( `.fullMCMC` = 80% di 200K), 96 000 ( `.mcmcBalanced` = 80% di 120K), 40 000 ( `.mcmcPreview` ) **Range:** 0 oppure 1 000 – **Defined in:**

 **TECNICO**

Iterazione a cui il rumore `T63 mcmcNoiseScale` viene smorzato completamente a zero (cosine decay da iter 0 a qui). V497c/V502 hanno trovato 80% del `maxIterations` ottimale — dà a MCMC abbastanza tempo di esplorazione, ma lascia l'ultimo 20% alla convergenza senza rumore. 0 = rumore costante su tutte le iterazioni (raramente sensato, MCMC non può poi convergere).

 **IN PAROLE SEMPLICI**

Quando il «tremolio» casuale degli splat termina. Nei preset MCMC all'80% delle iterazioni totali — prima esplorazione, poi convergenza. Lascia il valore.

**T70** `mcmcGrowthRate` **DETTAGLI**

**Default:** 0.05 (standard paper = 5%) **Range:** 0.01 – 0.2 **Defined in:**

 **TECNICO**

Tasso di crescita del target di popolazione MCMC per step di rilocazione. La logica: ad ogni evento reloc la dimensione target di popolazione viene aumentata di  $(1 + \text{growthRate})$ , fino a quando `T62 mcmcMaxGaussians` (o la variante scalata via `T72/T73`) viene raggiunta. V512/V522 hanno trovato 0.05 come ottimale — valori più alti portano a crescita troppo veloce (le gaussiane vengono inserite prima che l'Adam momentum possa integrarle), più bassi a scene sotto-densificate alla fine.

 **IN PAROLE SEMPLICI**

Quanto velocemente cresce il numero di splat con MCMC. 5% per step è ottimale. Lascia il valore.

**T71** **mcmcSigmoidK** **DETTAGLI****Default:** 100.0 **Range:** 10.0 – 500.0 **Defined in:** **TECNICO**

Parametro di sharpness sigmoid per l'attenuazione del rumore MCMC. Nello step SGLD il rumore per gaussiana viene smorzato attraverso — gaussiane ad alta opacità (il cui logit è positivo) ricevono esponenzialmente meno rumore di quelle a bassa opacità.  $K = 100$  è nitido, cioè la transizione da «pieno-rumore» a «nessun-rumore» avviene molto velocemente attorno a opacità 0.5. V484–V487 hanno trovato  $K = 100$  ottimale — valori più piccoli (10–50) lasciano tremare anche gaussiane ad alta opacità (distrugge gaussiane convergenti), più grandi ( $> 500$ ) rendono la transizione artificialmente dura e le gaussiane morte non vengono più spostate.

 **IN PAROLE SEMPLICI**

Parametro speciale che determina quanto nettamente MCMC distingue tra «abbastanza trasparente per spostare» e «solido, non toccare». Il valore standard è ottimale. Non regolare.

**T72** **mcmcCapMultiplier** **DETTAGLI****Default:** 3.0 (inizializzatore + `.fullMCMC`), 2.0 (`.mcmcPreview`), 2.5 (`.mcmcBalanced`), 2.98 (P8 Render), 5.32 (P9 Outdoor), 1.76 (P10 Indoor) **Range:** 0 (= disabilitato) oppure 1.0 – 10.0 **Defined in:** **TECNICO**

Funzione 1.4.5: scaling cap adattivo alla scena. Se T73 `mcmcAutoScaleByScene` è true, il cap effettivo viene calcolato come (limitato a capacità buffer). Sfondo: con scene grandi (ad es. volo di drone da 1545 foto → 158 K SfM-init) T62 = 150 000 è troppo basso — il density control non potrebbe affatto crescere. Con multiplier 3.0 il cap in questo esempio viene scalato a 474 K (158 K × 3.0). Q7-BayesOpt ha trovato ottimi specifici per scena: Outdoor beneficia di multiplier alto (5.32 → ~830 K cap con 156 K bicycle init), Indoor si accontenta di 1.76 (le pareti saturano più velocemente). Risoluzione completa del cap vedi metodo.

 **IN PAROLE SEMPLICI**

Moltiplicatore che adatta automaticamente il cap di splat alla dimensione della scena. Scena grande = più punti iniziali = cap più alto. Standard 3x va bene per la maggior parte delle scene; Outdoor preset sale a 5x (grandi range di profondità), Indoor a 1.76x (le pareti limitano comunque).

**T73** **mcmcAutoScaleByScene** **DETTAGLI**

**Default:** true (inizializzatore + tutti i preset MCMC)

**Range:** boolean **Defined in:**

 **TECNICO**

Funzione 1.4.5: master switch per la logica cap scene-aware (vedi T72 +). Se false, viene usato esclusivamente T62 `mcmcMaxGaussians` come cap (ritorno al comportamento 1.4.4). Standardmente on, perché altrimenti i problemi di mass extinction con scene grandi della 1.4.3 ritornano. Disattivare manualmente solo se vuoi esplicitamente impostare un cap rigido — ad es. per addestrare una variante 150 K la cui dimensione finale è pianificabile.

 **IN PAROLE SEMPLICI**

Accende l'adattamento automatico del cap splat alla dimensione della scena. Standardmente on. Lascia off solo se vuoi tu stesso esattamente un determinato numero di splat.

## Mip-Splatting (Q1.5) (T74–T76)

**Stato:** Q1.5 il 2026-05-25 dopo 14 iterazioni autonome + overnight 1.5M confidence check è stato scartato come «closed no-win» (max  $\Delta@2\times = +0.27$  dB, il gate originale richiedeva  $\geq +1.5$  dB di media su  $0.5\times/2\times$ , FALLISCE su 0/11 pair scenes). I campi rimangono **opt-in** per esperimenti di ricerca; tutti i preset di produzione hanno. Vedi verdetto: `docs/plans/2026-05-25-phase-q1.5-final-verdict.md`.

**T74 useMipSplatting** **DETTAGLI**

**Default:** false (tutti i preset di produzione), true (`.fullMCMCMip` — sibling di ricerca) **Range:** boolean **Defined in:**

 **TECNICO**

Attiva Mip-Splatting (Yu et al. CVPR 2024): filtro di smoothing 3D + filtro 2D + compensazione  $\alpha$ , che limita la frequenza per gaussiana al limite di Nyquist della densità di campionamento più alta della fotocamera di training. Obiettivo teorico: eliminazione dell'aliasing al rendering in scale fuori training (0.5x o 2x della risoluzione di training). Attivato negli shader di preprocess e backward projection, funzionalmente verificato corretto nel test Q1.5-D. Ma: il gate di accettazione originale ( $\Delta@1x \geq +0.3$  dB E  $\text{avg}(\Delta@0.5x, \Delta@2x) \geq +1.5$  dB) non è stato raggiunto su nessuna delle 11 pair scenes. Massimo osservato: family 750K classic  $\Delta@2x = +0.270$  dB. Le scene outdoor (Truck, Flowers) hanno mostrato addirittura peggioramento 1x e 0.5x. Ipotesi: lo smoothing 3D compete con la rilocazione MCMC con high-Gs. Il campo rimane per future re-eval multi-scala con metodologia Mip-NeRF-360 corretta (vedi O3-backlog nel benchmark path).

 **IN PAROLE SEMPLICI**

Filtro antialiasing da un paper del 2024. Teoricamente fantastico, praticamente nei nostri test non ha portato nulla e talvolta addirittura danneggiato. Rimane disponibile per sperimentatori, ma non lo raccomandiamo. Lascia off.

**T75 mipSmoothing3DScale** **DETTAGLI**

**Default:** 0.2 (default paper) **Range:** 0.05 – 1.0 **Defined in:**

 **TECNICO**

Parametro di scala di smoothing 3D (Yu et al. §3.3, default paper 0.2). Più grande = più smoothing spazio mondo per gaussiana (= più antialiasing, ma anche più blur nella scala di default), più piccolo = più nitido ma più suscettibile all'aliasing. Viene consultato solo se `T74 useMipSplatting = true`. Nei test Q1.5 non ulteriormente ottimizzato — il gate A/B ha già perso con default paper 0.2, ulteriori sweep sarebbero inutili.

 **IN PAROLE SEMPLICI**

Parametro Mip-Splatting. Se non hai acceso Mip, irrilevante.

**T76 mipFilter2DVariance** **DETTAGLI**

**Default:** 0.3 (= esattamente il comportamento legacy V242) **Range:** 0.1 – 1.0 **Defined in:**

 **TECNICO**

Varianza filtro 2D Mip che viene aggiunta alla diagonale  $\Sigma_{2D}$  (varianza direttamente, non al quadrato). 0.3 è esattamente il valore legacy V242, che prima di Mip-Splatting era hardcoded nel kernel. Se `T74 useMipSplatting = false`, il kernel ignora completamente questo valore e scrive lo hard-coded 0.3 — in modo che la baseline non possa regredire (garanzia Codex-Round-1-S3-1). Se, viene usato il valore qui impostato. Rimane nel catalogo dei campi per sweep Mip.

 **IN PAROLE SEMPLICI**

Altro parametro Mip-Splatting. Con Mip off: irrilevante.

**Densification adattiva (Q5) (T77–T79)****T77 adaptiveDensification** **DETTAGLI**

**Default:** false **Range:** boolean **Defined in:**

 **TECNICO**

Funzione Q5: rolling median tracker come alternativa al fisso `T11 densifyGradThreshold`. Se `true`, in ogni step di `densify` la soglia attuale viene sovrascritta con `median(ultimi N avgGrad samples) × T79 adaptiveDensifyMultiplier`. `N = T78 adaptiveWindow`. Più rigoroso di `V440 p98` (la trappola catastrofica di 63 K pruning), `median + 2×` si trova circa al `p70–p80` della distribuzione del gradiente in steady state. Test Q5: da solo FAIL 0/3 scene, ma insieme a Q6 (vedi `T80/T81`) PASS 1/3 scene — il bundle Q5+Q6 è stato passato come opt-in il 2026-05-25 ed è attivabile via CLI `--adaptive-densify`. Q6 è così il «portatore» del guadagno di qualità, Q5 contribuisce piuttosto alla stabilità.

 **IN PAROLE SEMPLICI**

Soglia di `densify` auto-apprendente. Invece di una sensibilità impostata fissa, l'app si adatta alla scena. Da sola testata non migliore, ma insieme al curriculum Q6 sì. Accendere entrambi insieme o entrambi off.

**T78** adaptiveWindow DETTAGLI**Default:** 1 000 **Range:** 100 – 10 000 **Defined in:** TECNICO

Rolling median window in eventi di densification (NON iterazioni — ogni step `T13 densifyInterval` fornisce un sample). Default 1 000 — con significa che le ultime 100 000 iterazioni di training contribuiscono al median, quindi tipicamente l'intera storia del training fino a qui. Fase iniziale (prima dei sample `T78`): il tracker restituisce nil → fallback alla soglia fissa `T11`. Rilevante solo se.

 IN PAROLE SEMPLICI

Quanti vecchi step di densify confluiscono nel median per T77. Standard 1000 va bene. Rilevante solo se Q5 Adaptive è on.

**T79** adaptiveDensifyMultiplier DETTAGLI**Default:** 2.0 **Range:** 1.0 – 4.0 **Defined in:** TECNICO

Moltiplicatore sul rolling median per la soglia adattiva. Default 2.0 corrisponde approssimativamente a p70–p80 della distribuzione tipica del gradiente. Più basso = crescita più aggressiva (più cloni), più alto = più rigoroso (meno cloni). Test Q5 in range 1.5–3.0 — 2.0 miglior default. Rilevante solo se.

 IN PAROLE SEMPLICI

Fattore per T77/T78. Standard 2.0 = più rigoroso del median tipico. Non regolare.

## Curriculum (Q6) (T80–T81)

### T80 curriculumResolutionRamp

#### DETTAGLI

**Default:** false **Range:** boolean **Defined in:**

#### TECNICO

Funzione Q6: la risoluzione di training inizia a  $0.5 \times$  e cambia a  $T50 \text{ positionLRScheduleEndIteration} / 2$  (o  $T1 \text{ maxIterations} / 2$ , se T50 non è impostato) a  $T22 \text{ trainingRenderScale}$ . Usa l'infrastruttura `resize/restoreImageBuffers` sviluppata in Q1.5.1. Sovrascrive T23 `resolutionWarmupScale` se attivato. Q6 è passato come «portatore del guadagno di qualità» nel bundle Q5+Q6 (vedi T77) — l'aumento graduale di risoluzione dà all'app tempo di trovare geometria grossolana sulla risoluzione più bassa, prima di passare al lavoro di dettaglio fine. Via CLI: `--curriculum-resolution`.

#### IN PAROLE SEMPLICI

«Prima grossolano, poi fine» per la risoluzione di training. Metà risoluzione nella prima metà, poi piena risoluzione. Aiuta in determinate scene, in altre no — meglio accendere insieme a T81.

### T81 curriculumSHProgression

#### DETTAGLI

**Default:** false **Range:** boolean **Defined in:**

#### TECNICO

Funzione Q6: sovrascrive T21 `shDegreeUpgradeIterations` con `[maxIter/4, maxIter/2, maxIter*3/4]`, distribuisce quindi gli upgrade SH uniformemente sul tempo di training invece di caricarli sul fronte. Ipotesi: la geometria stabile viene stabilita prima dell'esplosione del color detail, il che posiziona più precisamente gli effetti di luce dipendenti dalla direzione di vista. Q5+Q6 insieme PASS 1/3 scene, Q6 come portatore del guadagno (Q5 alone FAIL). Via CLI: `--curriculum-sh`.

#### IN PAROLE SEMPLICI

«Prima forma, poi colore» — gli effetti di lucentezza vengono sbloccati solo tardi nel training, in modo che gli splat trovino prima la loro posizione e dimensione. Con T80 insieme accendibile; da solo non porta tanto.

## Preset statici (TP1–TP9)

Qui solo le differenze strutturali rispetto al default dell'inizializzatore. La descrizione marketing completa degli undici preset UI P1–P11 la trovi nel Capitolo 7.

**TP1** `.preview`

### DETTAGLI

Preset diagnostico/anteprima per sistemi  $\geq 10$  GB RAM. Override rispetto all'inizializzatore: - 30 000  $\rightarrow$  5 000 - 15 000  $\rightarrow$  3 500 (70% di maxlter) -  $1.6e-6 \rightarrow 1.6e-5$  (10x piú alto, decay meno aggressivo) - ,, ,,, , ciascuno 2x (V176) - 3 000  $\rightarrow$  100 000 (effettivamente off, V172: il reset distrugge i training brevi)

- [1K, 2K, 3K]  $\rightarrow$  [1K, 2K] (V182: il grado 3 non converge

in 2K iter) - 1.0  $\rightarrow$  0.5

### IN PAROLE SEMPLICI

qualsiasi valutazione iniziale di una serie di immagini appena importate — 2–3 min di attesa, dopo il risultato basta per una domanda binaria «vale la pena un run Quality?».

**TP2** `.full`

### DETTAGLI

Quality di produzione Classic. Override: - 30 000  $\rightarrow$  35 000 (V550: i test 40K Truck overtraining +10.7% Gs con -1.3% L1) - 15 000  $\rightarrow$  5 000 (V310 sweet spot, V338 7K worse) - Tutte le LR 2x (V188) -  $1.6e-6 \rightarrow 1.6e-5$  (V45 10x) -  $2e-6 \rightarrow 1.1e-6$  (V335)

- 100  $\rightarrow$  200 (V112) - 0.005  $\rightarrow$  0.001 (V393) - 3 000  $\rightarrow$  100 000

(V194 disabilitato, V421 confermato) -

[1K, 2K, 3K]  $\rightarrow$  [2K, 5K, 8K] (V228 ritardato) - 0.0  $\rightarrow$  0.9995 (V546 HTGS, 14% di miglioramento) - 50 (invariato, V546) - false  $\rightarrow$  true (V438)

- 0  $\rightarrow$  20 000 (V431) - true (V443, già default

dell'inizializzatore per `.full` )

### IN PAROLE SEMPLICI

ogni ripresa fotografica standard (oggetto, piccola stanza, scultura) con  $< 500$  immagini. Il miglioramento del 14% del loss annunciato in V546 contro V438 è stato confermato su Horse Full come media di 3 trial.

**TP3** `.fullClassicPaper`

### DETTAGLI

Sibling di test Q1.5-A di TP2, Classic fedele al paper. Override rispetto a TP2: - 35 000  $\rightarrow$  30 000 (standard paper) - 5 000  $\rightarrow$  15 000 (paper: 50% di maxlter) -  $1.6e-5 \rightarrow 1.6e-6$  (default paper) - ,, indietro a default paper (0.05, 0.005, 0.001) -  $1.1e-6 \rightarrow 2e-7$  (calibrato per ~1-2M Gs su Bicycle) - 200  $\rightarrow$  100 (paper) - 0.001  $\rightarrow$  0.005 (default paper) - 100 000  $\rightarrow$  3 000 (paper §5.2, rischioso — può scatenare regressione V194) - 0.9995  $\rightarrow$  0.0 (paper non ha decay) - 20 000  $\rightarrow$  30 000 (cosine gira al 100% di maxlter)

### IN PAROLE SEMPLICI

esperimenti di ricerca Q1.5 che hanno bisogno di budget di gaussiane di magnitudine paper (1–2 M) per test Mip-Splatting. Dopo il verdetto Q1.5 «closed no-win» il preset rimane accessibile per utenti avanzati, ma non è raccomandato per produzione.

**TP4** `.fullMCMC` **DETTAGLI**

Quality di produzione MCMC. Override rispetto all'inizializzatore: - 30 000 → 200 000 (V534, MCMC ha bisogno di 5x più iter di Classic) - 15 000 → 160 000 (V504b 80% di maxIter) - 1.6e-6 → 1.6e-5 - LR schedule come TP2 (tutte 2x) - 0.2 → 0.05 (V521b/V534: MCMC ha bisogno di segnale L1 più forte) - [1K, 2K, 3K] → [2K, 5K, 8K] - `.classic` → `.mcmc` - 150 000 (già nell'inizializzatore, nel preset confermato) - 5e-5 (V467/V536 ottimale) - 0.005 → 0.01 (V535 ottimale) - 0 → 160 000 (80% di maxIter, V497c/V502) - 3.0 (già nell'inizializzatore) - true (già nell'inizializzatore) - 3 000 → 200 000 (effettivamente off, MCMC usa reloc invece di reset)

 **IN PAROLE SEMPLICI**

consegna web, object capture con aspirazione al dettaglio, voli di drone (anche se P9 Outdoor è ancora migliore). 71% in meno di gaussiane di Classic con L1 paragonabile.

**TP5** `.fullMCMCMip` **DETTAGLI**

Sibling di test Q1.5-D di TP4, con Mip-Splatting + budget MCMC di magnitudine paper. Override rispetto a TP4:

- `mcmcMaxGaussians` 150 000 → 1 500 000 (10x, magnitudine paper)
- `useMipSplatting` false → true (Mip on)

 **IN PAROLE SEMPLICI**

Tutti gli altri campi identici a TP4. Q1.5 D-PASS su Bicycle 2026-05-24 (interrompe la striscia di 12-iter multi-scale-FAIL). Verdetto finale Q1.5 2026-05-25 comunque closed-no-win — il guadagno Mip-Splatting non riproducibile su 11 pair scenes. Il preset rimane opt-in.

**TP6** `.classicBalanced` **DETTAGLI**

Mid-tier Classic. Override rispetto a TP2: - 35 000 → 20 000 (V149: 20K = 30K con 33% in meno di tempo) - 20 000 → 0 (cosine corre su maxIter = 20K, nessuna extended phase)

 **IN PAROLE SEMPLICI**

casi standard con tempo di attesa più breve. V149 identificato come sweet spot.

**TP7** `.mcmcPreview`**DETTAGLI**

Diagnostica MCMC. Override rispetto a 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 = scaling più leggero)

**IN PAROLE SEMPLICI**

vedere rapidamente un risultato MCMC, per valutare se TP4 o un preset Scene-Class vale la pena.

**TP8** `.mcmcBalanced`**DETTAGLI**

Mid-tier MCMC. Override rispetto a TP4: - 200 000 → 120 000 (V518) - 160 000 → 96 000 (80%) - 160 000 → 96 000 (80%) - 3.0 → 2.5 (tra Preview 2.0 e Full 3.0)

**IN PAROLE SEMPLICI**

MCMC senza il pieno run 200K. ~120 K iterazioni sono un buon compromesso tra qualità e tempo di attesa.

**TP9** `.quickTest`**DETTAGLI**

Puro test funzionale. Override rispetto all'inizializzatore: - 30 000 → 1 000 - 15 000 → 500 -  $2e-6$  →  $4e-6$  (calibrato per risoluzione 0.25x) - 100 → 50 - 3 000 → 100 000 (off, perché troppo breve) - 1.0 → 0.25

**IN PAROLE SEMPLICI**

sanity check «il training parte in generale sensatamente?». Durata < 30 s su M3 Ultra. Sicuramente sembra pastoso.

**Metodo:**

**Signature:** `public func resolveMcmcMaxGaussians(initialPointCount: Int, bufferCapacity: Int) -> Int` **Defined in:**

**TECNICO** Unica fonte di verità per la domanda «quante gaussiane MCMC può far crescere al massimo?». Si calcola da tre input: il T62 `mcmcMaxGaussians` configurato (con floor mass extinction 150 000, se 0), il (numero di punti SfM init) e la (dimensione del buffer gaussiano pre-allocato). Logica:

1. `base = T62 > 0 ? T62: 150_000` (il floor mass extinction protegge contro bug di default dell'inizializzatore come l'incidente mass extinction 1.4.3)
2. Se `T73 mcmcAutoScaleByScene && initialPointCount > 0 && T72 mcmcCapMultiplier > 0`:
  - `scaled = max(base, ceil(initialPointCount × T72))` altrimenti
3. Se `bufferCapacity > 0`: `return min(scaled, bufferCapacity)`
4. Altrimenti `return scaled`

Esempio: Bicycle (Mip-NeRF 360, 194 foto-frame) → SfM-init ~156 K punti, `T62 = 150 000`, `T72 = 5.32`, capacità buffer 8 M. Resolved cap =  $\min(8M, \max(150K, \text{ceil}(156K \times 5.32))) = \min(8M, 830K) = 830 K$ . È il cap di crescita effettivo a cui si attiene la logica di rilocalizzazione MCMC.

**IN PAROLE SEMPLICI** Calcola il vero numero massimo di splat con MCMC. Prende la tua impostazione, guarda quanti punti ha la tua scena all'inizio, e scala con il `Multiplier` se l'adattamento automatico è on. Così il cap si adatta alla scena, invece di forzare lo stesso valore per una scena piccola e una enorme. Non devi chiamare il metodo tu stesso — il training lo usa internamente.

## Quale campo per cosa? (cheat sheet)

Obiettivo	Campi da regolare
Più dettaglio in lontananza	<code>T62 mcmcMaxGaussians</code> alto, <code>T72 mcmcCapMultiplier</code> 5+
Più dettaglio in generale (Classic)	<code>T1 maxIterations</code> alto ( $\leq 40K$ ), <code>T2 densifyUntilIteration</code> $\leq 14\%$ di T1
Ridurre floater nei voli di drone	<code>T43 frustumCullEnabled</code> on, <code>T20 skyMaskingEnabled</code> on, <code>T45 skyDomeEnabled</code> on
Bel cielo in scene esterne	<code>T45 skyDomeEnabled</code> on, <code>T47 skyDomeRadiusMultiplier</code> 30–60
File di export più piccolo	Strategia <code>.mcmc</code> (T61), <code>T56 postTrainingCompactification</code> on, <code>T62 mcmcMaxGaussians</code> $\leq 200K$
Training più veloce	<code>T22 trainingRenderScale</code> 0.5, <code>T1 maxIterations</code> dimezzare — ma non entrambi!
Migliori luci speculari	<code>T21 shDegreeUpgradeIterations</code> con [2K, 5K, 8K] (nessun early front load), MCMC + 200K iter
Mantenere il Mac reattivo	<code>T25 throttleDelayMs</code> 5–10 (costa ~15% di tempo training)
Live preview più frequente	<code>T59 livePreviewInterval</code> giù a 10–20
Transizioni più morbide alle ombre	<code>T17 ssimWeight</code> un po' alto (0.15–0.25), ma non oltre 0.3
Mantenere interni compatti	Preset P10 Indoor (, <code>T72 = 1.76</code> )

## Campi pericolosi

Questi campi con configurazione errata possono portare a OOM, crash dell'app, mass extinction delle gaussiane o dati benchmark inutilizzabili. Da trattare con cautela:

- T11 `densifyGradThreshold` — un dimezzamento può generare 2–4× più gaussiane, il che fa rapidamente esplodere la memoria GPU. Da considerare anche: deve corrispondere a T22 `trainingRenderScale` (1.0× → 1e-6, 0.5× → 2e-6, 0.25× → 4e-6).
- T72 `mcmcCapMultiplier` — con scene grandi con > 200 K punti SfM-init e multiplier > 5 nasce un cap risolto di milioni di gaussiane. Su Mac da 36 GB di RAM OOM possibile. Il 5.32 del preset Outdoor funziona solo perché Mip-NeRF-360 Bicycle ha 156 K punti init → 830 K cap.
- T39 `testViewIndices` — l'impostazione manuale può rendere il benchmark inutilizzabile (tutti gli indici > N → nessun holdout). Lascia che il flag `--benchmark` lo imposti.
- T64 `mcmcOpacityRegWeight` e T65 `mcmcScaleRegWeight` — nella 1.4.3 beta impostati a 0.01, il che ha portato a mass extinction (460 K → 5 gaussiane in un'iterazione). Dalla 1.4.4 fissato a 0.0, ma l'aumento manuale può riprodurre il problema.
- T15 `opacityResetInterval` — se non 100 000+ (effettivamente off) e il training è più breve di 10 000 iterazioni, il reset distrugge la convergenza. `.preview` ce l'ha quindi su 100 000 nonostante `maxIterations = 5 000`.
- T54/T55 `densifyPhase2*` — la densification a due fasi è stata interrotta nei test in 0-gaussian cascade. Lascia entrambi a 0.
- T74 `useMipSplatting` — Q1.5 closed-no-win 2026-05-25, può addirittura peggiorare il PSNR su alcune scene outdoor. Default off, opt-in solo per ricerca.

Se un campo è in questa lista e vuoi modificarlo, fai prima un backup del tuo preset attuale (export come JSON) e considera se puoi misurare in modo riproducibile il risultato — altrimenti dopo non sai se hai prodotto un miglioramento o un peggioramento.

## CAPITOLO

## Capitolo 7 — Preset di qualità integrati

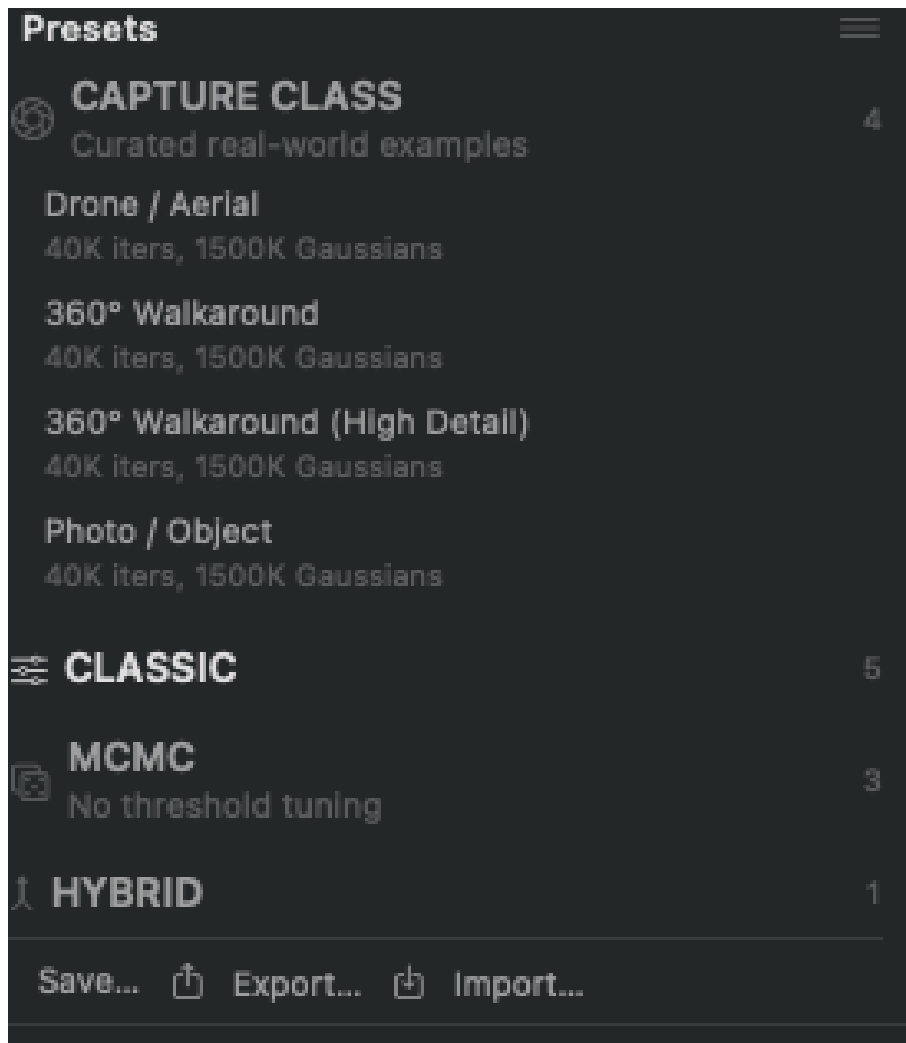


Figura 27: Sezione Preset con tutti e quattro i gruppi espansi — CAPTURE CLASS (4 preset: Drone/Aerial, 360° Walkaround, 360° Walkaround (High Detail), Photo/Object), CLASSIC (5 preset: Quick/Preview/Balanced/Quality/Ultra Detail), MCMC (3 preset, nota «No threshold tuning»), HYBRID (1 preset: Balanced (Hybrid))

**COSA MOSTRA L'IMMAGINE** Sezione Preset nell'Inspector, tutti e quattro i gruppi espansi. CAPTURE CLASS con i quattro preset curati per il mondo reale (Drone / Aerial, 360° Walkaround, 360° Walkaround (High Detail), Photo / Object) — è il gruppo primario e nella Modalità principiante l'unico visibile. CLASSIC con Quick (1K iter), Preview (5K iter, selezione attiva con segno di spunta blu), Balanced (20K iter), Quality (35K iter) e Ultra Detail (35K iter). MCMC con sottotitolo «No threshold tuning» — MCMC non ha bisogno

di una soglia densify-until: Preview (60K iter, 100K gaussiane), Balanced (120K, 150K), Quality (200K, 150K). HYBRID con il Balanced (Hybrid) (20K iter, 150K gaussiane). Riga azioni a piè di pagina: Save..., Export..., Import...

Un preset è una configurazione preparata per il training. RadianceKit fornisce tredici preset integrati in quattro gruppi: quattro preset **Capture-Class** (P9–P12) — ricette curate, validate a occhio su materiale reale della community per tipi di ripresa reali (drone, percorso 360°, foto-oggetto) e l'asse primario dalla v1.6 —, cinque preset Classic (P1–P5: Quick/Preview/Balanced/Quality/Ultra Detail), tre preset MCMC (P6–P8) e un preset Ibrido (P13) che combina le strategie Classic e MCMC. I precedenti preset «Scene-Class» (Render/3D, Outdoor, Indoor, ottimizzati nella Fase Q7 in modo accademico contro scene Mip-NeRF-360 e NeRF-Blender) sono stati ritirati come gruppo visibile nella v1.6 — la Capture-Class validata a occhio su materiale reale è ora l'asse primario; le configurazioni ottimizzate in Q7 rimangono solo internamente. Scegli i preset nella sidebar nell'area **Preset** o nella Modalità principiante durante l'importazione. I pulsanti **+** aprono finestre di dialogo per creare preset propri accanto — i tredici integrati non possono essere eliminati, ma duplicati.

Nella vista Esperto i preset appaiono raggruppati per tipo di ripresa e strategia (Capture Class / Classic / MCMC / Ibrido). Un clic su una voce scrive la configurazione del training memorizzata nello stato attuale. Non è uno snapshot — se in seguito ruoti i cursori, lo stato cambia, ma il preset stesso rimane invariato; una nota colorata mostra poi «modified».

Quale preset sia quello giusto quando dipende soprattutto dal tipo di scena e dall'hardware. I tre prospetti tabellari alla fine del capitolo riassumono il tutto.

## I P1 — Quick



Inspector → sezione Preset → gruppo «Classic» → voce «Quick». Suffisso UUID `...001`.



Preset diagnostico con 1 000 iterazioni, strategia di densification classica (adattiva) e una scala di risoluzione di training di 0.25× (l'immagine di input viene ridotta al 25 % prima del training). Non destinato a fornire una scena finale, ma a determinare rapidamente se il setup (pose della fotocamera, nuvola di punti, serie di immagini) mostra in generale un movimento sensato nei valori di loss. Su un M3 Ultra tipicamente meno di 30 secondi su 50–200 immagini. La piccola risoluzione nasconde la qualità reale dell'immagine, ma mantiene molto basso il fabbisogno di memoria e lo sforzo di rendering. Viene anche scelto automaticamente come default al primo avvio se il sistema ha meno di 10 GB di RAM.

### IN PAROLE SEMPLICI

Test di funzionamento veloce. Inserisci le immagini, aspetta una scarsa mezza minuta, controlla se appare il contorno grezzo della scena. Se l'immagine nel viewer sembra una macchia sfocata — bene, deve essere così. Se invece vedi solo punti scuri o una forma totalmente distorta, probabilmente le pose delle fotocamere sono sbagliate (vedi Capitolo 9). Per un risultato mostrabile, in seguito hai bisogno almeno di P2 o P3.

## | P2 — Preview (Classic)



DOVE

Inspector → sezione Preset → gruppo «Classic» → voce «Preview». Suffisso UUID ...002 .



TECNICO

5 000 iterazioni di densification classica, scala di risoluzione 0.5x, learning rate doppi rispetto allo standard. La densification (clone + split) è attiva nelle prime 2 500 iterazioni, dopo solo pruning. Preset predefinito per sistemi con  $\geq 10$  GB di RAM. Su un M3 Ultra tipicamente da 90 secondi a 3 minuti per una scena di 200 immagini. Fornisce un'impressione utilizzabile della geometria e della posa della fotocamera, ma le texture sono visibilmente sfocate — la risoluzione di rendering 0.5x non può poi essere aggirata direttamente con un nuovo training con P3 o P4, perché i learning rate sono calibrati appropriatamente per la metà della risoluzione.

### IN PAROLE SEMPLICI

Lo standard per «diamo solo un'occhiata veloce». Se hai appena importato nuove immagini e vuoi vedere se la scena è in generale ricostruibile, è il livello giusto. Circa 2–3 minuti di attesa, dopo puoi ruotare nel viewer 3D e valutare se vale la pena investire ulteriori passaggi di training. Solo quando il risultato dell'anteprima sembra già buono, vale la pena passare a Balanced o Quality.

## | P3 — Balanced (Classic)



DOVE

Inspector → sezione Preset → gruppo «Classic» → voce «Balanced». Suffisso UUID ...005 .



TECNICO

20 000 iterazioni di densification classica a piena risoluzione d'immagine. La densification corre per le prime 15 000 iterazioni, da iter 3 000 con un intervallo di densify di 100. Empiricamente lo «sweet spot» dalle sessioni di training documentate: con densification classica il loss L1 si stabilizza tra iter 18 000 e 22 000, un training più lungo non porta più miglioramenti significativi al di sotto di Quality (P4). Su un M3 Ultra tipicamente 30–60 secondi su 200 immagini, 5–8 minuti su 1 000+ immagini.

### IN PAROLE SEMPLICI

Il «buon compromesso». La maggior parte delle scene già sembra buona con questo, senza che tu debba aspettare un'ora. Se vuoi mostrare il risultato finale da qualche parte (social media, sito web, una demo per un cliente), spesso basta. Solo quando vuoi zoomare nel modello splat o hai bisogno di dettagli della texture superficiale, vale la pena il salto a P4 Quality o P7 MCMC.

## I P4 — Quality (Classic)



DOVE

Inspector → sezione Preset → gruppo «Classic» → voce «Quality». Suffisso UUID `...003`.



TECNICO

35 000 iterazioni di densification classica con V546 «Opacity Decay» (HTGS, Eurographics 2025): dopo ogni ciclo di densify l'opacità di tutte le gaussiane esistenti si moltiplica per un fattore  $< 1.0$ , il che rimuove in modo affidabile le gaussiane diventate inattive durante il pruning e ottiene così, a parità di numero di iter, un loss L1 migliore del 14 % rispetto al classico passaggio a 35 000. Il loss SSIM è attivato ( `ssimWeight=0.05` ). Su un M3 Ultra tipicamente 2–4 minuti su 200 immagini. Fornisce su NeRF-Blender (Lego, Chair, Drums) un L1 finale  $\approx 0.023$  — migliore variante Classic nei 560+ esperimenti documentati. Nota: richiede ~3–5 GB di memoria GPU; sui sistemi a 8 GB P3 è la scelta sicura.

### IN PAROLE SEMPLICI

La migliore variante classica. Fornisce texture nitide e geometria fine, soprattutto su riprese di oggetti (una scultura, una sedia, un vaso). Su grandi scene outdoor o stanze, invece, noti appena una differenza rispetto a Balanced — lì vale più la pena passare a un preset MCMC (P6–P8) o a un preset Capture-Class (P9–P12) che il salto da P3 a P4. Chi vuole il massimo assoluto della famiglia Classic prende P5 Ultra Detail.

## I P5 — Ultra Detail (Classic)



DOVE

Inspector → sezione Preset → gruppo «Classic» → voce «Ultra Detail». Suffisso UUID `...008`.



TECNICO

Circa 35 000 iterazioni di densification classica — il vincitore del passaggio held-out della Quality-Matrix (2026-06-10). Su tutte e tre le scene Mip-NeRF-360 testate, Ultra Detail batte il preset MCMC «Quality» integrato (P8) a parità di tempo wall-clock di circa +0.94 dB PSNR. È così il più forte preset Quality del gruppo Classic e la variante Classic più nitida che RadianceKit fornisce. Su un M3 Ultra tipicamente nello stesso intervallo di tempo di P4 Quality (2–5 minuti su 200 immagini), ma richiede un po' più di memoria GPU; sui sistemi a 8 GB P3 resta la scelta sicura.

### IN PAROLE SEMPLICI

Il livello Classic più nitido e il vincitore held-out dei nostri test di qualità: su scene reali circa un decibel migliore della variante MCMC «Quality» — a parità di tempo di attesa. Se vuoi la massima fedeltà di dettaglio con la collaudata densification classica e hai abbastanza memoria GPU, è la prima scelta. Se la memoria non basta o ti serve un file di esportazione più piccolo possibile, resta su P4 Quality o un preset MCMC.

## I P6 — Preview (MCMC)



DOVE

Inspector → sezione Preset → gruppo «MCMC» → voce «Preview». Suffisso UUID ...006 .



TECNICO

60 000 iterazioni di densification MCMC (3DGS-MCMC, NeurIPS 2024) con un cap di 100 000 gaussiane. MCMC sostituisce la logica euristica clone/split con la rilocalizzazione Markov-Chain-Monte-Carlo: le gaussiane morte vengono ricollocate tramite profondità di campionamento pesate sigmoid, il che dà un numero controllato e riproducibile di gaussiane. Il cap copre rigidamente il numero massimo a 100K — questo risparmia memoria e tempo di rendering, ma costa in dettaglio. Su un M3 Ultra tipicamente 5–8 minuti su 200 immagini. Adatto come «test di funzionamento MCMC» — aiuta a valutare se un passaggio da Classic a MCMC sarebbe sensato, prima di investire più tempo in P7 o P8.

### IN PAROLE SEMPLICI

Come P2 Preview, ma con il più recente metodo MCMC. Fornisce spesso nuvole splat un po' più compatte e distribuite più uniformemente rispetto alla variante Classic. Per una prima valutazione di una scena i 5–8 minuti di solito bastano. Se ti piace il risultato dell'anteprima, il prossimo passo è P7 (Balanced) o direttamente P8 (Quality MCMC).

## I P7 — Balanced (MCMC)



DOVE

Inspector → sezione Preset → gruppo «MCMC» → voce «Balanced». Suffisso UUID ...007 .



TECNICO

120 000 iterazioni MCMC con un cap di 150 000 gaussiane. Il livello MCMC intermedio — quasi il numero finale di gaussiane di P8 Quality, ma solo il 60 % delle iterazioni. Empiricamente il loss L1 nelle sessioni di training documentate è tra 0.026–0.028 su Horse Full, contro P8 con 0.0246 — circa il 7 % più alto, in cambio metà del tempo di attesa. Su un M3 Ultra tipicamente 8–15 minuti su 200 immagini. Utilizza una procedura che scala il cap effettivo delle gaussiane alla densità dei punti della nuvola SfM di ingresso (vedi T75 nel Capitolo 6).

### IN PAROLE SEMPLICI

MCMC con una decente profondità di dettaglio, ma senza il lungo passaggio completo di P8. Per la maggior parte delle scene basta, soprattutto se vuoi infilare un passaggio MCMC nella finestra oraria della pausa pranzo. Se la memoria scarseggia (ad es. su processori M con solo 16 GB), resta qui — P8 ha bisogno di più memoria GPU.

## I P8 — Quality (MCMC)



Inspector → sezione Preset → gruppo «MCMC» → voce «Quality». Suffisso UUID ...004 .



200 000 iterazioni MCMC con un cap di 150 000 gaussiane, loss SSIM 0.05, decadimento del rumore MCMC sull'80 % delle iterazioni. Best-single-run L1 nei 560+ esperimenti: 0.0238 su Horse Full, mediato su 3 trial 0.0246 (contro P4 0.0230 sulla stessa scena). MCMC fornisce inoltre il 71 % di gaussiane in meno (150K vs ~524K) — decisivo se vuoi distribuire il risultato sul web, perché la nuvola più piccola produce file di esportazione nettamente più piccoli. Tempo di training su un M3 Ultra tipicamente 20–35 minuti su 200 immagini; su set di 1 000+ immagini piuttosto 1–2 ore. Migliore scelta quando si desidera la massima qualità d'immagine con dimensione finale minima.

### IN PAROLE SEMPLICI

La migliore variante MCMC. Fornisce nuvole splat molto pulite e compatte — ideale se vuoi incorporare il risultato in seguito come web viewer 3D o inviarlo come file (il file è più piccolo che con P4 Quality a parità di qualità ottica). Hai bisogno però di pazienza — su grandi riprese più di un'ora di attesa. Pianificalo piuttosto come un passaggio «da tagliare durante la notte».

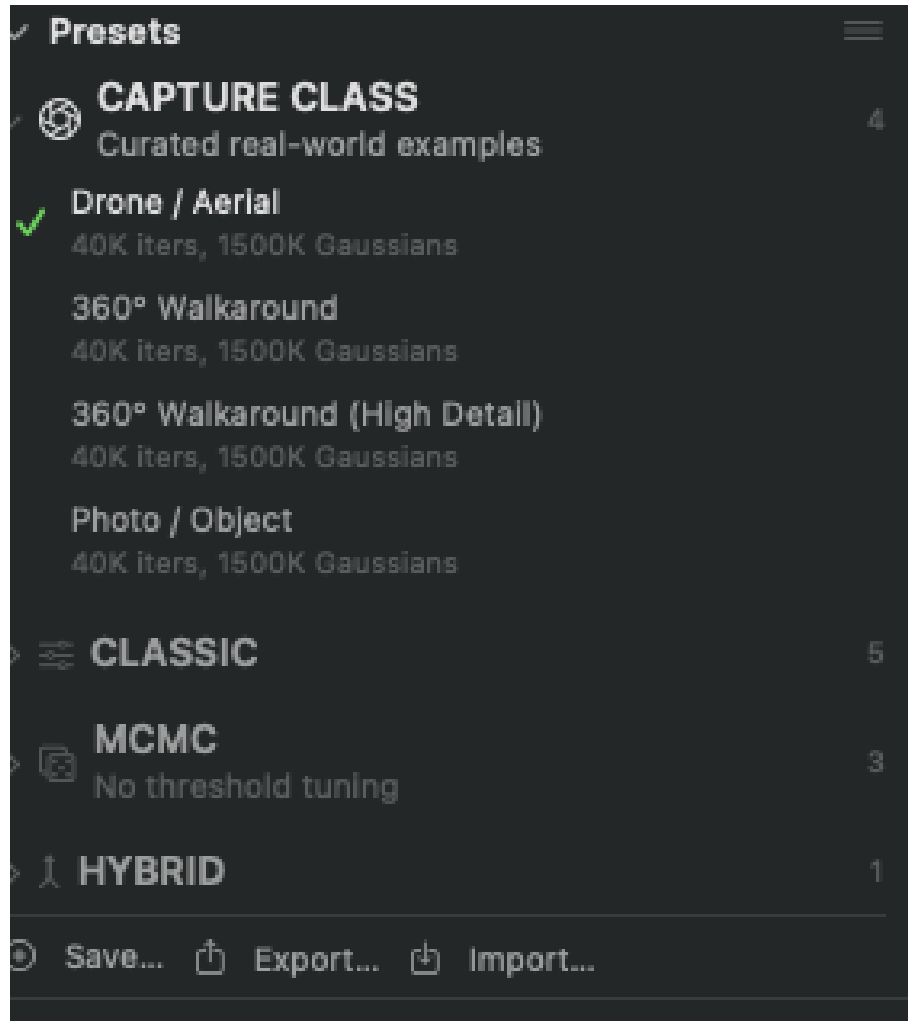


Figura 28: Gruppo CAPTURE CLASS espanso con tutti e quattro i preset curati per il mondo reale — Drone / Aerial (MCMC, 40K iter, Cap 1,5 M), 360° Walkaround (MCMC, 40K, Cap 1,5 M), 360° Walkaround (High Detail) (Ibrido, 40K, Cap 1,5 M, opt-in) e Photo / Object (Ibrido, 40K, Cap 1,5 M). Questo gruppo è in cima ed è l'unico visibile nella Modalità principiante.

**COSA MOSTRA L'IMMAGINE** Inspector con il gruppo CAPTURE CLASS espanso — il gruppo di preset primario dalla v1.6, l'unico mostrato nella Modalità principiante. Ogni voce è una ricetta validata a occhio su materiale reale della community per uno specifico tipo di ripresa (drone, percorso 360°, foto-oggetto), non un valore ottimizzato contro un set di test accademico. La selezione con clic scrive la configurazione del training memorizzata nello stato attuale.

## | P9 — Drone / Aerial



DOVE

Inspector → sezione Preset → gruppo «Capture Class» → voce «Drone / Aerial». Suffisso UUID ...010 .



TECNICO

Preset Capture-Class per orbite aeree e di drone attorno a edifici e paesaggi. Densifier MCMC, 40 000 iterazioni, cap 1,5 mln di gaussiane, loss SSIM 0.5 più termine edge-aware 0.1. Decisiva è la penalità di anisotropia con peso 0.003 a una soglia di rapporto di 6 — lo «spaghetti-killer» contro i tipici artefatti ad ago che il footage di drone produce. Validato su un volo di drone DJI 4K reale sopra il viadotto di Pensford (verificato a occhio, non solo metricamente).



IN PAROLE SEMPLICI

Per riprese dall'alto — voli di drone attorno a un edificio, sopra un paesaggio, lungo una facciata. La forte penalità di anisotropia spazza via gli artefatti ad ago o a spaghetti che il materiale di drone tende a produrre. Se il tuo materiale è ripreso da terra, è più adatto Photo / Object o un preset Classic.

## | P10 — 360° Walkaround



DOVE

Inspector → sezione Preset → gruppo «Capture Class» → voce «360° Walkaround». Suffisso UUID ...011 .



TECNICO

Preset Capture-Class per video di percorso 360°. Densifier MCMC, 40 000 iterazioni, cap 1,5 mln di gaussiane, loss SSIM 0.5 più termine edge-aware 0.1, lieve penalità di anisotropia (peso 0.001 a soglia di rapporto 15). Maschera per persone e cielo attiva. Il preset si aspetta un video equirettangolare 360°, che viene riproiettato internamente in crop prospettici larghi circa 90° prima dell'avvio del training. Validato su percorsi 360° 8K con selfie-stick (scena Monument, verificato a occhio).



IN PAROLE SEMPLICI

Per video di percorso 360° — cammini con una fotocamera 360° o un selfie-stick attraverso una stanza o attorno a un oggetto. RadianceKit scompone da solo il panorama sferico in angoli di visuale normali e maschera via passanti e cielo. Per la massima nitidezza sullo stesso materiale prova in aggiunta la variante High Detail (P11).

## | P11 — 360° Walkaround (High Detail)



DOVE

Inspector → sezione Preset → gruppo «Capture Class» → voce «360° Walkaround (High Detail)». Suffisso UUID ...013 (Opt-in).



TECNICO

Preset Capture-Class opt-in per video di percorso 360° con massimo dettaglio. Densifier Ibrido (clone/split classico a gradiente assoluto + rumore MCMC + rilocazione), 40 000 iterazioni, cap 1,5 mln di gaussiane, penalità di anisotropia 0.0015 a soglia di rapporto 15, loss SSIM 0.2 e termine edge-aware 0 — la ricetta screen-split «r50» bloccata. Su materiale 360° batte il preset MCMC standard «360° Walkaround» (P10) in PSNR, LPIPS e confetti visibili, e questo con circa un terzo del numero di splat. Sta deliberatamente opt-in *accanto* al preset 360 standard, finché non è validato su più scene.

### IN PAROLE SEMPLICI

L'alternativa più nitida al preset 360 standard (P10): più dettaglio, meno confetti, file nettamente più piccolo. Sta deliberatamente accanto invece di sostituirlo — finora confermato su una manciata di scene. Se il tuo percorso 360° è ripreso in modo pulito, prova prima questo preset e confronta il risultato con P10.

## | P12 — Photo / Object



DOVE

Inspector → sezione Preset → gruppo «Capture Class» → voce «Photo / Object». Suffisso UUID ...012 .



TECNICO

Preset Capture-Class per orbite di oggetti da foto singole nitide (nessun video). Densifier Ibrido-t1 (con rilocazione), 40 000 iterazioni, cap 1,5 mln di gaussiane, loss SSIM 0.5 più termine edge-aware 0.1, lieve penalità di anisotropia (peso 0.001 a soglia di rapporto 15), Opacity Decay 0.9995 ogni 50 iterazioni, **nessun** masking. Validato su 163 foto ad alta risoluzione da 41 MP di uno scheletro (verificato a occhio). Poche viste (fino a circa 600) rimangono sotto la soglia di collasso dell'Ibrido.

### IN PAROLE SEMPLICI

Per riprese di oggetti da foto singole nitide — giri attorno a una scultura, un modello, un prodotto con la fotocamera e fai foto invece di video. Nessun masking, perché le foto nitide hanno di solito uno sfondo pulito. Per sorgenti video prendi invece un preset 360° o Drone.

## | P13 — Bilanciato (Ibrido)

### DOVE

Inspector → sezione Preset → gruppo «Ibrido» → voce «Bilanciato (Ibrido)». Suffisso UUID `...009`.

### TECNICO

20 000 iterazioni con la strategia di densification ibrida: il clone/split classico guidato dal gradiente colloca la capacità dove il loss ne ha bisogno, il rumore SGLD di MCMC continua a esplorare e le gaussiane morte vengono ricollocate invece di andare perse nel pruning. L'Opacity Decay (V546) sostituisce i reset dell'opacità; una penalità di anisotropia (peso 0.001, soglia di rapporto 15) tiene sotto controllo gli splat a forma di ago. Il cap delle gaussiane scala con la scena (base 150K, scene-aware  $\times 3.0$ ). Validato su cinque scene contro MCMC puro a parità di budget: in media +0.45 dB PSNR con il 20–30 % di gaussiane in meno (stonehenge +1.23, family +0.82, garden +0.47 dB). Su un M3 Ultra tipicamente 5–10 minuti su 200 immagini.

### IN PAROLE SEMPLICI

Una forte prima scelta per un risultato finale: dettagli più nitidi rispetto ai preset MCMC con un file altrettanto compatto, in una frazione del tempo di training di P8. Se hai tempo per un solo passaggio di qualità e nessuna delle Capture-Class corrisponde chiaramente, inizia qui. I preset Classic restano migliori per i test veloci, e i preset Capture-Class (P9–P12) sono la prima scelta quando la tua scena corrisponde chiaramente a uno di quei tipi di ripresa.

## Quando quale preset?

Scenario	Primo test	Passaggio principale
Test funzionale nuove immagini, < 30 s	<b>P1 Quick</b>	—
Orbita oggetto da foto singole nitide	P2 Preview	<b>P12 Photo / Object</b>
Scansione oggetto singolo (video), < 500 foto	P2 Preview	<b>P4 Quality o P8 Quality MCMC</b>
Video percorso 360°	P6 Preview MCMC	<b>P10 360° Walkaround</b> (nitido: <b>P11 High Detail</b> )
Orbita aerea / di drone, paesaggio	P6 Preview MCMC	<b>P9 Drone / Aerial</b>
Consegna web (piccolo, compatto)	P2	<b>P8 Quality MCMC</b> (file più piccolo a piena qualità)
Dettagli nitidi in poco tempo, esportazione compatta	P2 o P6	<b>P13 Bilanciato (Ibrido)</b>
Massima fedeltà di dettaglio, strategia Classic	P3 o P6	<b>P5 Ultra Detail</b>
Stampa, marketing, pieno dettaglio	P3 o P6	<b>P4 Quality (Classic) o P5 Ultra Detail</b>

## Confronto rapido

Pre-set	Strategia	IterGs max		Scala rendering	Tempo tipico (200 immagini, Q-Sweep M3 Ultra)	
P1 Quick	Classic	1 000	$\infty$	0.25x	~30 s	—
P2 Preview	Classic	5 000	$\infty$	0.5x	2–3 min	—
P3 Balanced	Classic	20 000	$\infty$	1.0x	30–60 s	—
P4 Quality	Classic	35 000	$\infty$	1.0x	2–4 min	V546 HTGS
P5 Ultra Detail	Classic	~35 000	$\infty$	1.0x	2–5 min	Matrix $\Delta+0.94$ 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	Occhio / Viadotto
P10 360° Walkaround	MCMC	40 000	1.5 M	1.0x	10–25 min	Occhio / Monument
P11 360° Walkaround (High Detail)	Ibrido	40 000	1.5 M	1.0x	10–25 min	Occhio (opt-in)
P12 Photo / Object	Ibrido	40 000	1.5 M	1.0x	10–25 min	Occhio / Scheletro
P13 Balanced (Ibrido)	Ibrido	20 000	150 K	1.0x	5–10 min	Matrix $\Delta+0.45$ dB

## Preset propri

Tramite il pulsante **Save...** nella sezione Preset (I1 nel Capitolo 2) salvi la configurazione di training attuale come preset proprio. I preset propri non sono «integrati» e possono essere rinominati, esportati (come JSON), condivisi tramite trascinamento, duplicati ed eliminati. I tredici preset integrati P1–P13 rimangono intoccati dal pulsante di eliminazione.

**Regola pratica:** Se modifichi qualcosa in un preset che vorrai usare ancora più volte — Sky-Dome attivo, peso SSIM più alto per una certa classe di scena, numero di iter diverso — allora salva la variante come preset proprio. Così alla prossima esecuzione sai subito che è una configurazione diversa dallo standard.

## CAPITOLO

# Capitolo 8 — Formati di esportazione

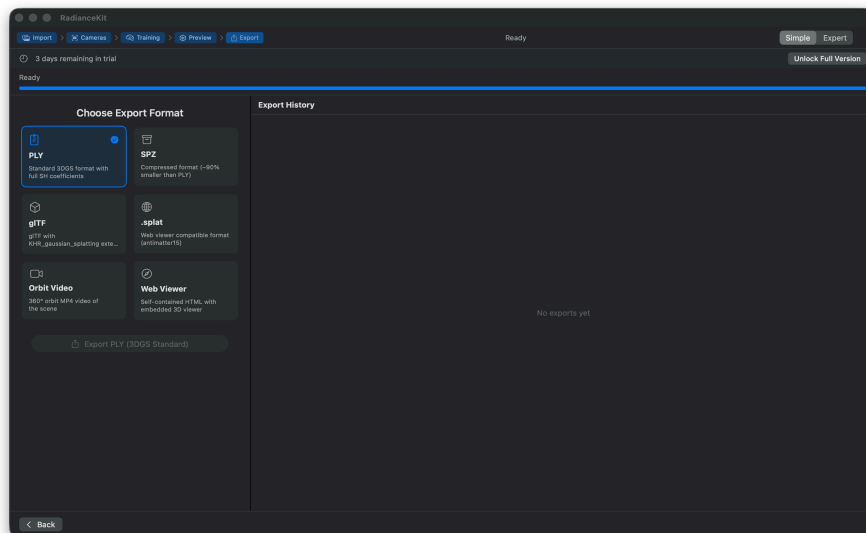


Figura 29: Selezione del formato di esportazione in Modalità principiante — sei schede di formato

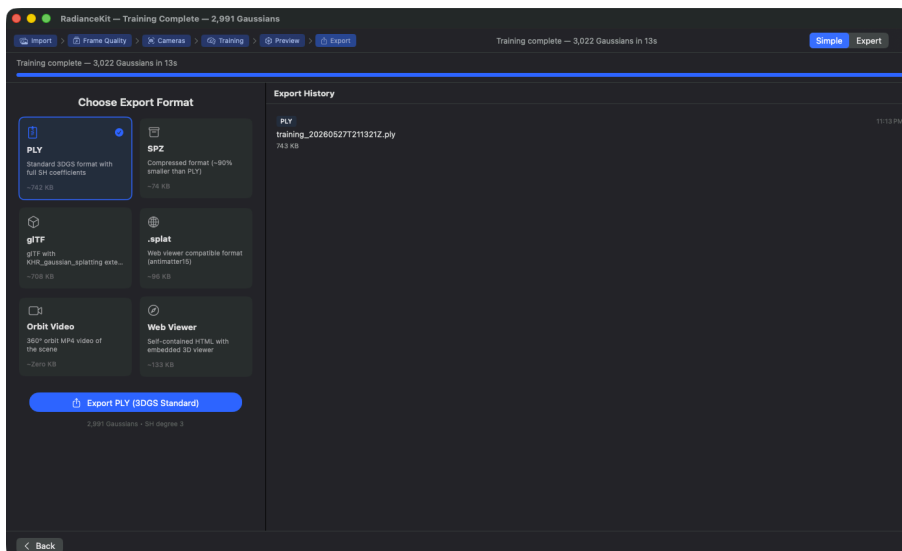


Figura 30: Griglia formati di esportazione live dopo training 5K iter su un bouquet di fiori — tutte e sei le schede con calcolo dinamico della dimensione (PLY 742 KB selezionato, SPZ 74 KB, glTF 708 KB, .splat 96 KB, Orbit Video ~Zero KB, Web Viewer 133 KB), cronologia esportazioni a destra con PLY già salvato

**Cosa mostra l'immagine (2 991 gaussiane, SH grado 3, bouquet sintetico Blender come set di test IP-clean):** Le indicazioni di dimensione sotto ogni scheda di formato vengono calcolate in tempo reale dal conteggio attuale delle gaussiane e dall'overhead del formato — non sono hard-coded. Da 2 991 gaussiane (SH grado 3) nascono così 742 KB di PLY, 74 KB di SPZ (fattore ~10× più piccolo grazie alla quantizzazione), 708 KB di glTF (con l'estensione `KHR_gaussian_splatting`, quindi quasi equivalente al PLY), 96 KB di `.splat` (formato compresso a 24 byte per gaussiana). Orbit Video mostra «~Zero KB», perché la dimensione è nota solo dopo la codifica MP4. Web Viewer (133 KB) raggruppa un file HTML autonomo con viewer WebGL incorporato e dati `splat` compressi — più grande del puro `.splat` per l'overhead del viewer. La cronologia delle esportazioni a destra elenca l'esportazione PLY già completata («`training_20260527T211321Z.ply`, 743 KB, 23:13») con pillola di formato e azione «Mostra nel Finder».

Un training completato fornisce una nuvola gaussiana — una collezione di alcune centinaia di migliaia fino a milioni di distribuzioni gaussiane 3D, che insieme ricostruiscono la scena. RadianceKit conosce dieci modi per scrivere questa nuvola su disco. Sei di essi sono formati di dati 3D puri (PLY, Compressed PLY, SPZ, SOG, glTF, `.splat`), uno raggruppa la nuvola insieme a un viewer HTML pronto (Web Viewer), uno renderizza un file MP4 da una traiettoria di fotocamera in orbita (Orbit Video), e due non esportano contenuti gaussiani ma soltanto il risultato SfM (pose della fotocamera e nuvola di punti grossolana) per il riutilizzo in altre pipeline di training (`transforms.json` + COLMAP Workspace).

Quale formato sia quello giusto quando dipende dall'obiettivo. Per l'archiviazione dei dati completi senza perdita di qualità si prende PLY. Per viewer web sulla propria pagina di solito bastano `.splat` o il Web Viewer integrato. Quando il file deve essere minimo, vale la pena SPZ o SOG. Per il riutilizzo del risultato SfM in Nerfstudio, Postshot o Brush, `transforms.json` e il COLMAP Workspace sono i percorsi giusti.

Tutte le funzioni di esportazione si trovano nel menu «Export» così come nella Modalità principiante all'ultimo passaggio del wizard. La maggior parte dei formati è pienamente conforme al sandbox e funziona nella versione App Store. Solo SOG richiede un binario esterno ( `cwebp` ), che non è necessariamente presente nella build App Store — per i dettagli vedi E4.

## I E1 — PLY (.ply)

### DOVE

Barra dei menu → Export → 3D Formats → Export PLY... (⌘E). Modalità principiante: passaggio wizard Export → scheda formato «PLY». **Dimensione:** tipicamente 100 % (valore di riferimento). **Compatibile con:** SuperSplat, PolyCam, tutti i viewer 3DGS.

### TECNICO

PLY è il formato di archiviazione canonico per 3D Gaussian Splatting. RadianceKit scrive un file binario little-endian con il layout di proprietà 3DGS standardizzato: per gaussiana posizione a tre componenti, tre normali sempre impostate a zero, tre coefficienti SH DC ( `f_dc_0..2` ) per il colore RGB di base, seguiti da fino a 45 ulteriori coefficienti SH ( `f_rest_0..44` ) nell'arrangiamento trasposto channel-major definito dal paper Kerbl 2023 (prima tutti i coefficienti del canale R, poi tutti i G, poi tutti i B), seguiti da opacità logit (valori grezzi pre-sigmoid), tre scale in log-space e una rotazione quaternionica wxyz. Il grado SH massimo esportato viene limitato al minimo tra il desiderio dell'utente e il grado effettivamente appreso; il default è 3 (45 coefficienti rest). Prima della scrittura, la dimensione del payload viene calcolata in interi a 64 bit per intercettare overflow su nuvole estremamente grandi. Il file viene scritto in modo atomico, il che su grandi nuvole occupa brevemente il doppio spazio su disco.

### IN PAROLE SEMPLICI

Questo è il «file originale». File più grande, massima compatibilità, nessuna perdita. Se non sai quale formato prendere, prendi PLY — si apre in quasi ogni strumento 3DGS. Per 1 milione di gaussiane sono, a seconda del grado SH, tra 200 e 800 MB. Se il file diventa troppo grande, dai un'occhiata a E2 (PLY compresso) o E3 (SPZ).

## I E2 — Compressed PLY (.ply)

### DOVE

Barra dei menu → Export → 3D Formats → Export Compressed PLY... Modalità principiante: scheda formato «Compressed PLY». **Dimensione:** circa 10–20 % rispetto a PLY (compressione 5–10 volte).

**Compatibile con:** SuperSplat, motore PlayCanvas, viewer basati su web.

### TECNICO

La variante PlayCanvas del formato PLY con quantizzazione chunked. Le gaussiane sono raggruppate in chunk da 256. Per chunk vengono memorizzati separatamente nell'header i bound min/max per posizione, scala e colore; le singole gaussiane riferiscono i loro valori relativi a questi bound e vengono compresse a 32 bit ciascuna: posizione e scala con packing 11-10-11 bit, rotazione come quaternione «smallest-three» 2-10-10-10 bit, colore come RGBA 8-8-8-8. I coefficienti SH superiori vengono quantizzati con soli 8 bit per componente (`shCoeffCount * 3 uchar` per gaussiana). Il formato stesso è ancora PLY con header ASCII e quindi in linea di principio validabile con strumenti PLY, ma le proprietà dei vertici sono dichiarate come campi `uint`. Il grado SH è di default 0 (nessun coefficiente rest), per massimizzare la compressione — gradi SH più alti possono essere selezionati esplicitamente.

### IN PAROLE SEMPLICI

La variante PLY salva-spazio. Compatibilità con i motori identica al PLY normale, ma 5–10 volte più piccolo. SuperSplat e PlayCanvas lo leggono nativamente. Per il deployment web quasi sempre meglio del PLY normale. La perdita di qualità dovuta alla quantizzazione di solito non è visivamente percepibile, finché la scena non contiene dettagli a frequenza estremamente alta.

## | E3 — SPZ (.spz)

### DOVE

Barra dei menu → Export → 3D Formats → Export SPZ.... Modalità principiante: scheda formato «SPZ». **Dimensione:** circa 10 % rispetto a PLY (90 % più piccolo). **Compatibile con:** Niantic Scaniverse, Niantic Spatial Fields, MetalSplatter.

### TECNICO

Il formato SPZ v2 di Niantic. Le posizioni vengono impacchettate come fixed-point a 24 bit (questo dà una risoluzione di circa 0,25 mm), le scale come quantizzazione a 8 bit nel log-space, le rotazioni come smallest-three a 8 bit (in v2 vengono memorizzati solo xyz, w viene derivato dal decoder dalla norma del quaternion), le opacità come valori sigmoidizzati a 8 bit. Il SH DC viene memorizzato con una formula di packing specifica SPZ ( $dc\_raw * 0.15 * 255 + 0.5 * 255$ ), le bande SH superiori con 5 bit (banda 1) rispettivamente 4 bit (banda 2-3) per coefficiente. L'intero blob binario impacchettato viene poi compresso con gzip standard (RFC 1952), il che dà un formato container gzipped con i magic byte `1f 8b`. RadianceKit invoca per questo il `gzip` di sistema, perché l'API zlib integrata di Apple genera un framing Apple proprietario che non sarebbe compatibile con i reader SPZ in Spatial Fields o MetalSplatter. Il `gzip` di sistema rimane invocabile anche all'interno del sandbox macOS.

### IN PAROLE SEMPLICI

Il file standard più piccolo. Se conosci Scaniverse di Niantic — è il formato che l'app usa. Molto piccolo, molto favorevole al caricamento per le app mobili. Direttamente utilizzabile nel viewer cloud proprio di Niantic (Spatial Fields). Circa il 90 % più piccolo di un PLY con gli stessi dati, mentre per la maggior parte delle scene otticamente quasi indistinguibile.

## I E4 — SOG (.sog)

### DOVE

Barra dei menu → Export → 3D Formats → Export SOG.... Modalità principiante: scheda formato «SOG». **Dimensione:** circa 5–6 % rispetto a PLY (compressione 15–20 volte — l'opzione più piccola). **Compatibile con:** motore PlayCanvas, editor SuperSplat.

### TECNICO

«Spatially Ordered Gaussians» — un formato PlayCanvas che memorizza la nuvola GPU-ready in più immagini WebP lossless. Prima tutte le gaussiane vengono ordinate spazialmente tramite codice Morton 3D (Z-order a 30 bit, 10 bit per asse), il che dà alle immagini una successiva cache locality nel renderer. Poi le posizioni vengono quantizzate con trasformazione logaritmica simmetrica (per un migliore range dinamico) a valori a 16 bit e divise in due immagini RGBA ( `means_l.webp` per gli 8 bit inferiori, `means_u.webp` per i superiori). Le rotazioni vengono codificate come `smallest-three` con 3×8 bit più modalità a 2 bit in un'immagine RGBA (la modalità finisce in alpha come `252 + largest` ). Le scale e il SH DC vengono quantizzati ciascuno con un codebook a 256 voci (distribuito basato su percentile su tutti i valori), gli indici finiscono in `scales.webp` e `sh0.webp` . Le cinque immagini più un `meta.json` con codebook e bound vengono impacchettate in un file ZIP (encoder personalizzato, perché il sandbox blocca lo `zip` di sistema) e salvate con l'estensione `.sog` .

**Attenzione sandbox:** SOG è l'unica opzione di formato che richiede un binario esterno. Lo stadio di encoder WebP invoca `cwebp` da `/usr/local/bin/cwebp` o `/opt/homebrew/bin/cwebp`. Se non viene trovato alcun binario `cwebp`, il codice ricade su una codifica PNG grezza — ma: **il fallback PNG non funziona in SuperSplat**. Nella versione App Store valuta la disponibilità in base alla variante di build; nella variante sviluppatore `cwebp` deve essere installato tramite Homebrew ( `brew install webp` ).

### IN PAROLE SEMPLICI

Il formato 3DGS più piccolo in assoluto, decisamente più piccolo di SPZ. Ma: ha bisogno dello strumento `cwebp` sul tuo Mac, perché RadianceKit stesso non può generare tutti i formati di immagine. Installalo una volta con Homebrew, poi tutto funziona. Nella versione App Store eventualmente non pienamente funzionante — se durante l'esportazione esce PNG invece di WebP, non puoi aprire il file direttamente in SuperSplat. Chi lavora senza Homebrew prende invece SPZ (E3).

## I E5 — glTF (.glb)

### DOVE

Barra dei menu → Export → 3D Formats → Export glTF... Modalità principiante: scheda formato «glTF». **Dimensione:** paragonabile a PLY. **Compatibile con:** viewer glTF con l'estensione `KHR_gaussian_splatting` (bozza standard Khronos).

### TECNICO

Scrive un file binario `.glb` autosufficiente (nessun allegato file bin separato) secondo la specifica dell'estensione `KHR_gaussian_splatting`. Le posizioni vengono memorizzate come dati vertice glTF `POSITION` regolari (float3), tutti gli altri attributi (rotazione come float4, scala come float3, opacità come float, coefficienti SH come float3 × `shCoeffCount`) si trovano in attributi vertice aggiuntivi e vengono referenziati tramite l'estensione. Importante: glTF utilizza un sistema di coordinate Y-up destrorso, COLMAP/3DGS lavora Y-down/Z-forward. Quindi l'esportatore applica una rotazione di 180 gradi attorno all'asse X — le posizioni vengono riscritte con `(x, -y, -z)`, i quaternioni vengono adattati a `(w, x, -y, -z)`. Questo dà una rappresentazione geometricamente corretta e congrua (non specchiata) nei viewer glTF. I chunk JSON e binari vengono allineati a 4 byte come richiesto dallo standard GLB.

### IN PAROLE SEMPLICI

Il formato standard ufficiale Khronos per dati 3D, nella nuova estensione per gli splat gaussiani. Vantaggio: glTF è diffuso in tutti i grandi motori 3D (Babylon.js, Three.js, Unity, Unreal). Svantaggio: l'estensione nel 2026 è ancora allo stadio di bozza, molti viewer non possono ancora leggerla. Sensato soprattutto quando integri dati splat in una pipeline glTF esistente o scrivi un viewer che è già abilitato glTF.

## E6 — Splat (.splat)

### DOVE

Barra dei menu → Export → 3D Formats → Export .splat... Modalità principiante: scheda formato «.splat». **Dimensione:** esattamente 32 byte per gaussiana. **Compatibile con:** gsplat.js, viewer basati su web (riferimento antimatter15), la maggior parte delle demo 3DGS nel browser.

### TECNICO

Il formato `.splat` di antimatter15 — 32 byte per gaussiana, nessun header, nessuna indirezione. Layout per voce: 3 × float32 posizione (coordinate mondo), 3 × float32 scala (trasformata exp dal log-space del buffer interno), 4 × uint8 colore RGBA (coefficiente SH DC scalato con `SH_C0 = 0.282...` e limitato a `[0,255]`), 4 × uint8 quaternion (`w,x,y,z`, normalizzato e codificato come `128 + 128*q` nell'intervallo byte). Viene memorizzato solo SH DC — le bande SH superiori vengono scartate. Questo rende il formato estremamente compatto, ma costa le variazioni di colore dipendenti dalla vista che si verificano in riflessi o highlight speculari. L'ordine di scrittura è esattamente l'ordine di indice della nuvola (nessun ordinamento spaziale), i viewer web come `gsplat.js` renderizzano partendo da questo presupposto.

### IN PAROLE SEMPLICI

Il formato di scelta se vuoi mostrare lo splat in un viewer web proprio con `gsplat.js`. Molto compatto (32 byte per gaussiana), ma nessun grado SH superiore — quindi nessun riflesso lucido o sottili variazioni di colore a seconda dell'angolazione. Per la maggior parte delle applicazioni web non è un problema, perché il colore DC basta completamente e la mancanza di dipendenza dalla vista si nota appena.

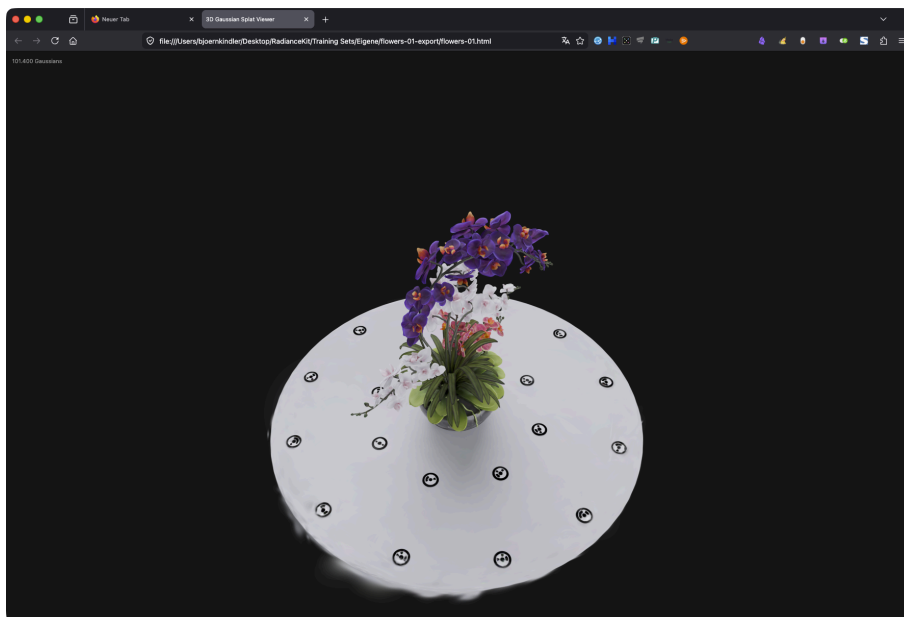


Figura 31: Web Viewer aperto in Firefox — splat del bouquet renderizzato con sfere marker delle fotocamere circostanti, barra dei tab del browser visibile in alto, nessun setup CDN/ server necessario. File `flowers-01.html` autonomo aperto direttamente dal Finder con doppio clic nel browser predefinito — il programma WebGL2 incorporato renderizza la nuvola gaussiana immediatamente, senza rete o server. I marker neri attorno al bouquet sono le fotocamere di training, opzionalmente visualizzabili. Il drag del mouse ruota, lo scroll fa lo zoom.

## | E7 — Web Viewer (.html)

### DOVE

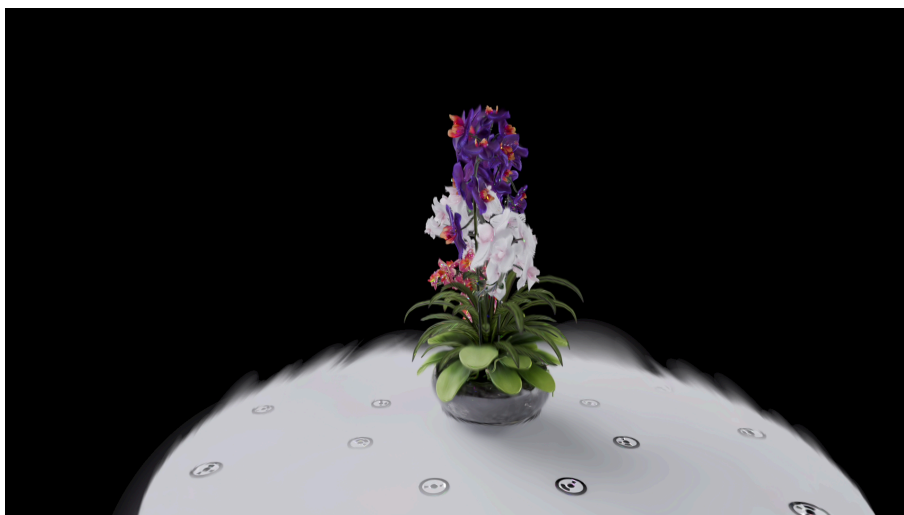
Barra dei menu → Export → Media → Export Web Viewer... Modalità principiante: scheda formato «Web Viewer». **Dimensione:** dati splat codificati base64 ( $\approx 4/3$  di overhead) + circa 5 KB di shell HTML/JS. **Compatibile con:** qualsiasi browser moderno con WebGL2 (tutti i desktop, iOS 15+, Android 5+).

### TECNICO

Raggruppa la nuvola gaussiana insieme a un renderer WebGL2 scritto completamente inline in un singolo file `.html`. Non ci sono dipendenze CDN, nessun WASM, nessun secondo file. La nuvola viene prima codificata internamente come binario `.splat` (stessa logica a 32 byte di E6), poi incorporata in base64, poi decodificata nel browser con `atob`. Il renderer integrato esegue il proprio ordinamento WebGL2, il controllo orbit del mouse e l'ordinamento CPU per frame; tutto il codice JS (shader, matematica, loop) è visibile nell'HTML di output. La convenzione degli assi al confine memoria-renderer è esattamente la stessa di E5: posizione  $(x, -y, -z)$ , quaternion  $(w, x, -y, -z)$ . Opzionalmente può essere visualizzato un overlay di branding (switch del tier gratuito). Poiché tutto è inline, il file funziona anche direttamente dal protocollo `file://` — nessun server web locale necessario per i test.

### IN PAROLE SEMPLICI

Un singolo file HTML che puoi inviare a qualcuno per mail o incorporare in una pagina web. Doppio clic nel Finder, e il browser mostra la tua scena con la rotazione del mouse. Nessun upload nel cloud necessario, nessun secondo file, nessun server. Ideale per presentazioni clienti, portfolio, allegati email. Svantaggio: il file diventa di circa un terzo più grande di un puro `.splat` per la codifica base64 — per scene molto grandi vale quindi la pena ospitare il file `.splat` separatamente insieme a un viewer standard.



*Figura 32: Frame singolo estratto da flowers-01.mp4 — bouquet in render di profilo, piattaforma bianca con marker fotocamera visibile, sfondo nero (sfondo viewport predefinito, modificabile nelle Impostazioni). La fotocamera circonda la scena su una traiettoria parametrica (elevazione + distanza fisse, yaw ruota), durata tipica 6–10 secondi a 30 o 60 fps. Risoluzione frame scalabile da 480p a 8K tramite VideoPreset.*

## | E8 — Orbit Video (.mp4/.mov)

### DOVE

Barra dei menu → Viewport → Record Turntable Video OPPURE Barra dei menu → Export → Media → Export Orbit Video.... Modalità principiante: scheda formato «Orbit Video» con cursore di durata 3–30 s. **Dimensione:** dipende da durata, risoluzione, bitrate. **Compatibile con:** tutte le piattaforme (H.264 e HEVC sono standard Apple).

### TECNICO

Renderizza la nuvola gaussiana lungo una traiettoria di fotocamera in orbita parametrica e codifica ogni frame tramite AVAssetWriter in un file MP4 o MOV. La configurazione dell'orbita controlla velocità di rotazione (giri), distanza, elevazione, FOV, durata e fattore ease-in/out. L'esportazione del video in orbita passa attraverso il ForwardPass PROPRIO di RadianceKit con valutazione SH completa — identica pixel per pixel al viewport in-app (WYSIWYG). Per frame, la matrice di allineamento mondo (calcolata dal renderer per ruotare le coordinate interne nel mondo orbit Y-up) viene moltiplicata con la fotocamera, poi viene applicato un capovolgimento di convenzione fotocamera (camFlip: orbit Y-up → COLMAP Y-down). Il target di rendering offscreen viene tirato tramite IOSurface a un CVPixelBuffer per l'encoder. L'encoder supporta H.264 e HEVC, bitrate configurabile e risoluzione da 480p a 8K. Prima del primo frame, il renderer attende 200 ms affinché l'ordinamento iniziale degli splat sia completato. Questa esportazione è GPU-bound — a 8K e milioni di gaussiane il tempo di rendering per frame è di diversi secondi, quindi sono possibili tempi totali di rendering di 10–30 minuti per 6 s di video.

### IN PAROLE SEMPLICI

Un file MP4 pronto con una rotazione attorno alla tua scena. Perfetto per i social media, marketing, presentazioni. Puoi impostare durata (3–30 secondi), direzione di rotazione e velocità. Il file può essere incorporato direttamente in YouTube, Instagram, in PowerPoint e ovunque altrove. Va in parte lentamente, perché l'app deve renderizzare completamente ogni frame — per un video 8K puoi pianificare da cinque a trenta minuti, a seconda del numero di gaussiane.

## I E9 — SfM Transforms (transforms.json)

### DOVE

Barra dei menu → Export → Photogrammetry → Export SfM (transforms.json)... **Dimensione:** tipicamente 1–10 KB (solo pose + intrinseci, nessuna immagine, nessuna gaussiana). **Compatibile con:** nerfstudio, Brush, gsplat, OpenSplat, Meshroom, tutti i trainer 3DGS feed-forward moderni.

### TECNICO

Scrivi il formato `transforms.json` di nerfstudio con una lista di pose della fotocamera più intrinseci condivisi. Per fotocamera, la view matrix (RadiancesKit interno: world-to-camera in convenzione COLMAP) viene invertita, poi i vettori base Y e Z locali alla fotocamera vengono specchiati per convertire alla convenzione nerfstudio (stile OpenGL, fotocamera guarda lungo  $-Z$ ,  $+Y$  è in alto). La matrice 4x4 finale finisce come nested array row-major di double nel campo `transform_matrix` di ogni frame. Gli intrinseci vengono memorizzati al livello superiore (lunghezza focale x/y, punto principale x/y, larghezza/altezza dell'immagine, `camera_model = "OPENCV"`, più i coefficienti di distorsione `k1, k2, p1, p2`) — a meno che l'esportatore non riconosca più set di intrinseci diversi, allora vengono scritti per frame. I percorsi delle immagini vengono scritti come `images/<filename>` relativi al file JSON; l'utente deve creare una cartella `images/ sibling` con le foto di training.

### IN PAROLE SEMPLICI

Questo file JSON descrive per ogni foto dove si trovava la fotocamera e dove guardava. Il file da solo è piccolo e inutile — viene utilizzato insieme alle immagini originali in una cartella. Nerfstudio, Brush e alcuni altri trainer leggono esattamente questo formato, e puoi così passare i tuoi risultati SfM di RadiancesKit a un altro strumento senza che lì la ricostruzione della fotocamera debba essere ricalcolata. Su scene grandi risparmia ore.

## I E10 — COLMAP Workspace (sparse/0/)

### DOVE

Barra dei menu → Export → Photogrammetry → Export SfM (COLMAP Workspace).... **Dimensione:** tre file binari insieme tipicamente 4–8 MB — `points3D.bin` domina (una riga per punto 3D della nuvola sparsa), `images.bin` e `cameras.bin` sono ciascuno ben sotto i 100 KB. **Compatibile con:** COLMAP stesso, Nerfstudio, Postshot, Meshroom, tutti gli strumenti che si aspettano una directory `sparse/ COLMAP`.

### TECNICO

Scrivi il layout `sparse/0/` standard di COLMAP con tre file binari: `cameras.bin`, `images.bin`, `points3D.bin`. Il riferimento del formato è la documentazione ufficiale di COLMAP. `cameras.bin` contiene la lista degli intrinseci deduplicata (le fotocamere con intrinseci identici + dimensione dell'immagine vengono riassunte in una singola voce); il modello di fotocamera utilizzato è `OPENCV` (modello 4), con `fx/fy/cx/cy` più i quattro coefficienti di distorsione `k1/k2/p1/p2`. `images.bin` elenca per immagine la posa come quaternion `wxyz` più la traslazione, seguita dall'ID della fotocamera e dal nome del file; non vengono memorizzate corrispondenze 2D-3D. `points3D.bin` contiene la nuvola di punti SfM con posizione, colore (RGB 0-255) e valori predefiniti per riproiezione e lunghezza dei track. Tutto viene scritto in little-endian. La re-importazione in RadiancesKit funziona tramite il menu File → «Import COLMAP/Metashape Workspace...» (vedi Q3 nel capitolo Backend SfM).

### IN PAROLE SEMPLICI

Il formato COLMAP ufficiale. Se vuoi continuare il tuo training in Postshot, Nerfstudio o un altro software compatibile con COLMAP, è il percorso giusto. Tre piccoli file più le tue immagini originali, e il programma di destinazione lo accetta come se COLMAP stesso fosse stato il programma sorgente. Più programmi lo comprendono rispetto al formato `transforms.json` (E9), contemporaneamente un po' meno maneggevole, perché binario invece che basato su testo.

## Quale formato quando?

Obiettivo	Formato
Viewer web sulla propria pagina	E7 Web Viewer (.html)
Viewer web con <code>gsplat.js</code>	E6 Splat (.splat)
Riutilizzo pipeline in Postshot / Nerf-studio	E9 transforms.json + E10 COLMAP Workspace
Modifica SuperSplat	E1 PLY o E2 Compressed PLY
Niantic Scaniverse / Spatial Fields	E3 SPZ
Massima compressione	E4 SOG (cwebp richiesto)
Video marketing/social	E8 Orbit Video

## Confronto rapido

Forma- to	Estensione	Sandbox	Dimensione (1M gauss)	Uso mi- gliore
E1 PLY	.ply	sì	~250 MB	Archivio, massima compatibili- tà
E2 Com- pressed PLY	.ply	sì	~40 MB	Web + Su- perSplat
E3 SPZ	.spz	sì (gzip- spawn)	~40 MB	Niantic + Mobile
E4 SOG	.sog	condi- zionato (cwebp)	~20 MB	Massima compres- sione
E5 glTF	.glb	sì	~250 MB	Pipeline Kh- ronos
E6 Splat	.splat	sì	~32 MB	Viewer web gsplat.js
E7 Web Viewer	.html	sì	~45 MB	File browser standalone
E8 Orbit Video	.mp4 / .mov	sì	variabile	Social/mar- keting
E9 SfM Tran- sforms	.json	sì	~5 KB	Trasferi- mento pose
E10 COLMAP Work- space	Directory	sì	~4–8 MB	Trasferi- mento pose binario

La colonna dimensione è un'indicazione approssimativa per 1 mln di gaussiane con grado SH 3. I valori reali variano a seconda della comprimibilità della scena; il grado SH 0 riduce PLY/glTF di un fattore 4.

## CAPITOLO

# Capitolo 9 — Backend SfM

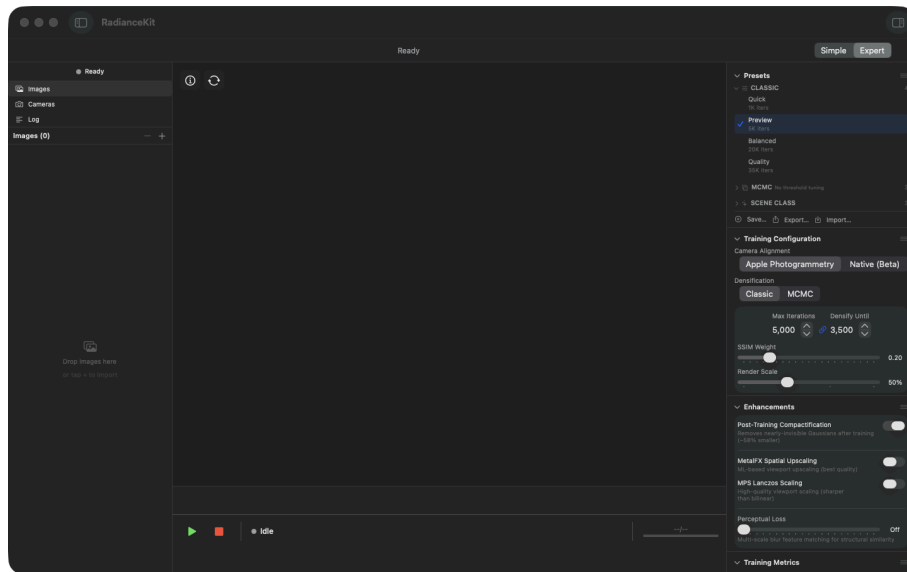


Figura 33: Modalità esperto con picker Camera Alignment nell'Inspector (Apple Photogrammetry / Native (Beta))

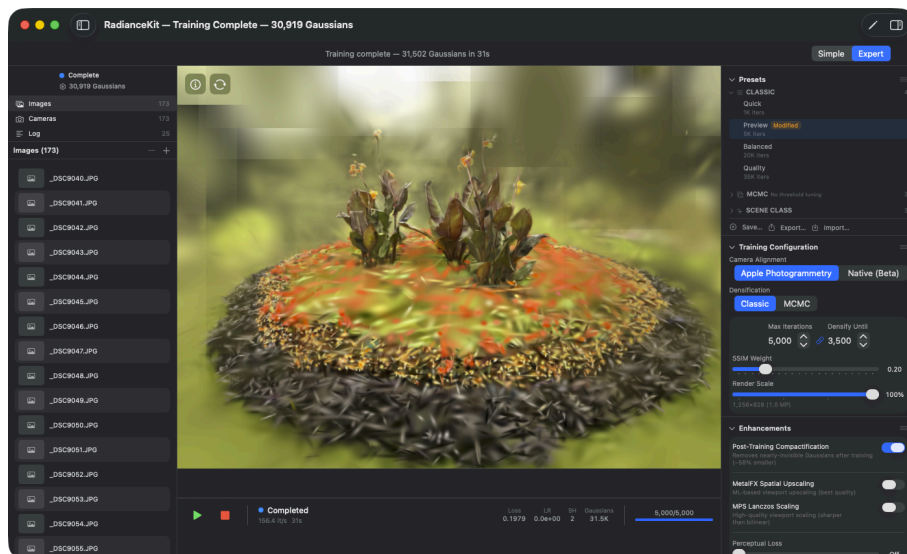


Figura 34: Inspector con Native (Beta) attivo — picker Camera Alignment seconda opzione selezionata, tutti gli altri parametri di configurazione del training invariati

**COSA MOSTRA L'IMMAGINE** Il picker Camera Alignment nell'Inspector è un segmented control con due opzioni — Apple Photogrammetry (predefinito per le build dell'App Store, pienamente conforme al sandbox) e Native (Beta) (il backend pipeline FA-

ST+BRIEF+GLOMAP proprio di RadianceKit, sviluppato durante la Fase 3.8/3.9, allo stato 2026-05). Native (Beta) è validato solo su orbite ed è più veloce di Apple Photogrammetry con  $\geq 1000$  frame, ma non soddisfa ancora il gate di qualità Phase-3 §5 (finalLoss  $\leq 0.0115$ ) — da cui il tag Beta. I risultati SfM esterni provenienti da Metashape, COLMAP o altro software di fotogrammetria possono inoltre essere importati tramite il menu File (Q3 formato testo COLMAP, Q6 importazione Workspace) — il picker non cambia, ma le pose importate sostituiscono il risultato SfM.

SfM sta per **Structure from Motion**. Da un insieme di foto sovrapposte, il software ricostruisce per ogni immagine la posizione e la direzione di vista della fotocamera in un sistema di coordinate 3D condiviso. In questo processo viene generata una nuvola di punti 3D grossolana, che inizializza il training con Gaussian Splatting. Il risultato SfM è l'input per il training vero e proprio e determina in modo decisivo la qualità finale dell'immagine.

RadianceKit offre cinque percorsi SfM: due backend integrati nell'app (Q1 Apple Photogrammetry, Q4/Q5 Native), due percorsi di importazione da strumenti esterni (Q3 formato testo COLMAP, Q6 importazione binaria Workspace) oltre a Q2 COLMAP binario, che è disponibile solo nelle build per sviluppatori al di fuori dell'App Store. Quale sia quello giusto dipende dal tipo di scena (orbita attorno a un oggetto, spazio interno, volo di drone) e dal fatto che un software esterno fornisca già una ricostruzione.

## I Q1 — Apple Photogrammetry



Vista Esperto → Inspector → Configurazione del training → picker Camera Alignment, voce «Apple Photogrammetry».



Avvolge il framework di fotogrammetria integrato di Apple, sviluppato originariamente per Object Capture. Apple estrae internamente le feature con una pipeline proprietaria (i passaggi non sono documentati pubblicamente), le verifica tramite matching multi-vista e risolve il bundle adjustment sulla Neural Engine + GPU Apple Silicon. Il backend è pienamente conforme all'App Store (nessun binario esterno, Sandbox=true, on-device), ma fornisce solo le pose delle fotocamere più una nuvola di punti grossolana — nessuna metrica diagnostica come lunghezza dei track o errore di proiezione. Scala secondo la raccomandazione di Apple fino a qualche centinaio di immagini. Con più di ~500 frame in voli di drone lineari o grandi scene outdoor sono stati osservati crash riproducibili o lo scarto silenzioso di singole fotocamere.

### IN PAROLE SEMPLICI

È il modo più semplice. Inserisci le immagini, l'app calcola. Funziona molto bene per scansioni di oggetti classiche — quando giri attorno a un mobile o a una scultura e scatti 50–200 foto. Con voli di drone su paesaggi o con un numero molto elevato di immagini (oltre 500), tuttavia, il metodo di Apple tende a diventare instabile. Per queste scene prova il backend Native (Q4/Q5) o calcola le fotocamere in Metashape e caricale tramite l'importazione Workspace (Q6).

**POWER-USER**

Q2 COLMAP binario — avvia il programma COLMAP esterno come sottoprocesso e quindi **non è disponibile** nella versione App Store (sandbox). Funziona solo nelle build per sviluppatori al di fuori dell'App Store. Per la qualità che COLMAP offre, nella versione App Store c'è l'importazione Workspace (Q3 o Q6): calcola l'SfM in COLMAP o Metashape esternamente e carica il risultato.

**Q3 — Formato testo COLMAP (Metashape / ETH3D)** **DOVE**

Menu «File → Import COLMAP / Metashape Workspace...» (Cmd+⇧+I) OPPURE trascina una cartella contenente `sparse/0/cameras.txt`.

 **TECNICO**

Legge l'esportazione testo COLMAP standardizzata — tre file di testo `cameras.txt`, `images.txt`, `points3D.txt` nella sottocartella `sparse/0/` — e converte nel modello interno del risultato SfM. Stessa definizione di formato dell'esportazione binaria COLMAP, solo in ASCII anziché in binario. Viene esportato in questo preciso layout da Agisoft Metashape, RealityCapture, PolyCam e dal benchmark ETH3D. Il parser condivide il riconoscimento del modello di fotocamera con il parser binario (tutti i modelli comuni: SIMPLE\_PINHOLE, PINHOLE, OPENCV, OPENCV\_FISHEYE, FULL\_OPENCV). Robusto rispetto a righe di commento e righe vuote. Scala nei test fino a ~1 400 fotocamere (ETH3D Tunnel) senza problemi.

 **IN PAROLE SEMPLICI**

Se hai già lavorato con Metashape, RealityCapture o un altro software commerciale di foto-3D ed esportato il risultato — questo export puoi caricarlo direttamente in RadianceKit, senza che l'app debba ricalcolare. Questo fa risparmiare ore di attesa. Basta caricare l'intera cartella tramite il menu File o trascinarla nella finestra.

## I Q4 — SfM Native (incrementale)



Vista Esperto → Inspector → Configurazione del training → picker Camera Alignment, voce «Native (Beta)». Incrementale è la modalità predefinita di questo backend — nell'Inspector non c'è un picker separato per il mapper. Tramite CLI la modalità può essere impostata esplicitamente con `--native-sfm` o `--sfm-mapper incremental`.



Implementazione propria GPU-accelerata dell'intera pipeline SfM: feature FAST+BRIEF OPPURE SuperPoint+LightGlue tramite CoreML (con `--coreml-features`), seguito da matching Hamming-KNN, matrice fondamentale RANSAC, costruzione dei track, selezione della coppia iniziale, bootstrap a due viste (F→E più DLT), mapper incrementale greedy con registrazione PnP e triangolazione multi-vista, e bundle adjustment finale tramite Levenberg-Marquardt ridotto secondo Schur con loss di Huber e Jacobiani analitici tramite risoluzione Cholesky. Pienamente conforme all'App Store: nessun binario esterno, Sandbox=true. Con il rilevatore di collasso R2 distribuito nella Fase 3.10: se l'app registra meno del 60 % dei frame di input o se il tasso di punti per fotocamera scende sotto 13, ricade automaticamente sul mapper globale (Q5). Empiricamente pulito su scene orbita/turntable; su movimenti più generali (volo di drone, ambienti interni con geometria complessa) il tasso di successo è più basso — il rilevatore intercetta tuttavia questi casi. Scala in modo affidabile fino a ~200 fotocamere, oltre con tempi di esecuzione decisamente più lunghi.

### IN PAROLE SEMPLICI

I punti di forza di Apple (compatibile con l'App Store, veloce per le orbite) con valori diagnostici aggiuntivi. Funziona particolarmente bene quando giri attorno a un soggetto come per un Object Capture. Con riprese più complicate (volo di drone o soggiorno), RadianceKit rileva automaticamente che non funzionerà e passa al metodo globale. Contrassegnato «Beta» perché ancora in fase di test — la raccomandazione standard rimane Apple Photogrammetry per semplici scansioni di oggetti e l'importazione Workspace (Q3 o Q6) per set outdoor impegnativi.

## I Q5 — SfM Native (globale)



Viene chiamato automaticamente quando il mapper incrementale (Q4) fa scattare il rilevatore di collasso (meno del 60 % dei frame di input registrati o tasso di punti per fotocamera sotto 13). Forzabile manualmente solo tramite CLI `--sfm-mapper global`. Nell'Inspector il metodo globale non è raggiungibile tramite un picker separato — l'app decide da sola quando passare.



Variante globale della pipeline nativa. Prima estrazione delle feature + matching come in Q4, poi stima della posa relativa per tutte le coppie verificate, seguita da rotation averaging (sincronizza tutte le rotazioni delle fotocamere nel sistema di coordinate mondo) e translation averaging (basato su LSQR su una formulazione sparsa matrix-free per evitare overflow di interi con grandi numeri di fotocamere). Scala in linea di principio a ~5 000 fotocamere, in pratica con qualità degradata sopra alcune centinaia di fotocamere — la misurazione del gate di accettazione Fase 3.8 §5 su K-1351 ha dato finalLoss 0.07 invece dei 0.0115 mirati. Trattato come «fallback tier»: entra in gioco quando il mapper incrementale degenera, ma non viene esso stesso ricontrollato per la qualità.

### IN PAROLE SEMPLICI

Il percorso piano-B per il motore nativo. Viene chiamato automaticamente quando il percorso incrementale più veloce fallisce. Fornisce un risultato utilizzabile, ma con scene molto grandi o difficili di solito non è preciso quanto quello che ottieni da Metashape o da un'installazione COLMAP esterna. Se Native diventa il tuo flusso di lavoro standard, in questi casi vale la pena fare la deviazione tramite l'importazione Workspace (Q3 o Q6).

## I Q6 — Importazione Workspace Metashape / COLMAP testo (Fase Q7)

### DOVE

Menu File → «Import COLMAP / Metashape Workspace...» (Cmd+⇧+I). Trascinamento di una cartella contenente `sparse/0/cameras.{bin,txt}` e `images/`.

### TECNICO

Rileva automaticamente se una cartella selezionata tramite trascinamento o pannello di apertura corrisponde a uno dei tre layout di workspace COLMAP (`sparse/0/`, `sparse/` o `root`) e se la ricostruzione è binaria (`cameras.bin`) o testo (`cameras.txt`). Il percorso binario utilizza il parser binario COLMAP, il percorso testo il loader ETH3D — entrambi producono lo stesso modello di risultato SfM e il resto della pipeline (importare le immagini, avviare il training MCMC) è agnostico rispetto alla sorgente. Le immagini vengono aperte tramite il sistema di bookmark sandbox dell'app in modo security-scoped, così l'importazione funziona anche nella versione App Store. Pensato specificamente per il caso «esportazione Metashape senza ricalcolare la ricostruzione». Il riconoscimento menzionato nella voce di menu File avverte nel log dell'app se la cartella scelta non è un workspace riconoscibile.

### IN PAROLE SEMPLICI

È specificamente la funzione per utenti Metashape. Se hai una licenza per Metashape o RealityCapture e hai fatto lì la ricostruzione delle fotocamere, puoi semplicemente trascinare la cartella di esportazione qui e iniziare subito il training. Risparmia diverse ore di tempo di calcolo su scene grandi, perché RadianceKit non esegue allora l'SfM da sé.

## Quale backend quando?

Scenario	Backend consigliato
Scansione oggetto, 50–200 foto	Q1 Apple Photogrammetry
Grandi outdoor / drone / >500 immagini	Q6 Importazione Workspace (calcolare in Metashape o COLMAP, poi caricare)
Disponibile esportazione Metashape/RealityCapture	Q6 Importazione (nessun SfM necessario)
Set ETH3D / COLMAP testo accademico	Q3 Importazione testo COLMAP
Strettamente conforme App Store + scena orbita	Q4 Native incrementale
Q4 fallisce	Q5 Native globale (automatico)
Dati benchmark ETH3D	Q3 (autotest precomputato)

## Confronto rapido

Backend	App Store	Sandbox	Binario esterno	Uso migliore	Cam max ~
Q1 Apple PG	✓	✓	—	Oggetto-orbita	~300
Q2 COLMAP binario	✗ (solo build sviluppatore)	—	colmap/glomap	Outdoor grande	~5 000
Q3 Importazione testo COLMAP	✓	✓	—	Bench rig	~1 500
Q4 Native incrementale	✓	✓	—	Oggetto-orbita	~200
Q5 Native globale	✓	✓	—	Fallback Q4	~1 351
Q6 Importazione Workspace	✓	✓	—	Riuso Metashape	per sorgente

## CAPITOLO

## Capitolo 10 — Modalità principiante

La Modalità principiante (Simple Mode, Cmd+1) è il flusso di lavoro guidato per chi ricostruisce per la prima volta una scena 3D Gaussian Splatting. Invece di mostrare una sidebar piena di campi Inspector, l'app guida attraverso quattro passaggi: prima importare immagini o un video e scegliere un preset di qualità, poi gira l'elaborazione (SfM + training), successivamente la scena finita può essere valutata in un'anteprima 3D, e infine viene esportata nel formato desiderato. Una stretta barra di progresso al margine superiore della finestra mostra in qualsiasi momento in quale passaggio ti trovi.

Rispetto alla Modalità esperto (Cmd+2), che mostra tutti i controlli contemporaneamente, la Modalità principiante nasconde le opzioni non utilizzate, dà avvisi di validazione con immagini troppo poche o scadenti e offre in ogni passaggio solo i pulsanti che hanno senso nello stato attuale. Puoi passare in qualsiasi momento tra Modalità principiante ed esperto (Cmd+1 / Cmd+2), l'intero stato — immagini importate, preset scelto, training in corso, nuvola di punti finita — viene preservato ed è immediatamente disponibile nell'altra modalità.

### Z1 — Importazione (scegliere immagini e preset)

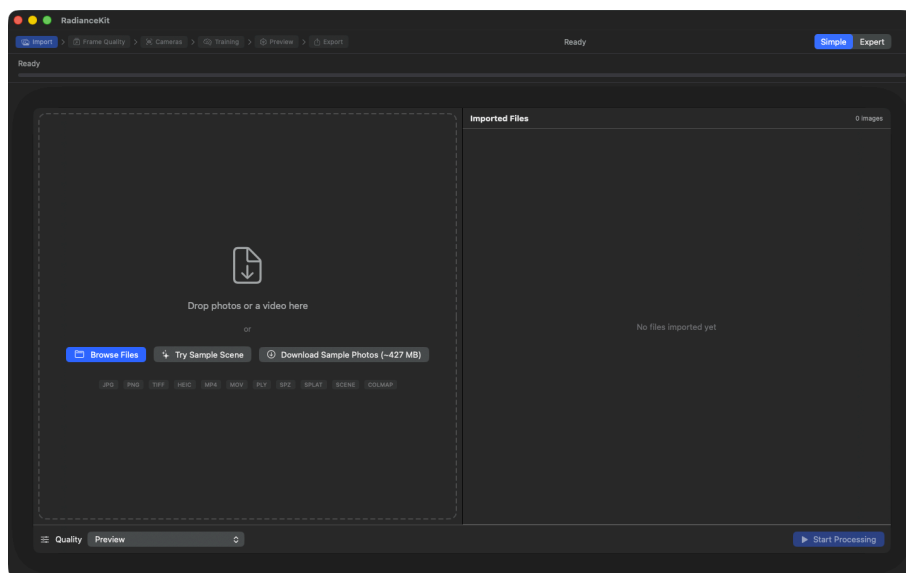


Figura 35: Modalità principiante passaggio 1 — drop zone vuota prima dell'import, crumb trail in alto (Import → Frame Quality → Cameras → Training → Preview → Export), pillole formato JPG/PNG/TIFF/HEIC/MP4/MOV/PLY/SPZ/SPLAT/SCENE/COLMAP

**COSA MOSTRA L'IMMAGINE** Il crumb trail (Import attivo) mostra il flusso di lavoro a quattro passaggi. Drop zone a sinistra con tre CTA: «Browse Files» (NSOpenPanel), «Try Sample Scene» (demo bundled), «Download Sample Photos (~427 MB)» (sottoinsieme di fiori Mip-NeRF360). Le pillole formato sotto elencano tutti i tipi di file accettati. A destra «Imported Files» con contatore «0 images» ed empty state «No files imported yet». In basso picker Quality (default: Preview) e «Start Processing» (disabilitato finché non ci sono immagini).

Il primo passaggio consiste nel dare all'app del materiale fotografico. Tramite trasciamento nel grande campo tratteggiato al centro, tramite il pulsante «Browse Files» o tramite clic sulla scena di esempio fornita. A destra appare una lista di tutte le immagini importate con risoluzione e dimensione del file; in basso nella barra degli strumenti fluttuante scegli il preset di qualità e avvii la pipeline con «Start Processing». Gli avvisi di validazione (rosso con < 3 o < 10 immagini, arancione con 10–19) mostrano se l'app si aspetta una ricostruzione sensata o no.

### C-01 ProgressIndicator (visualizzazione del passaggio)



In alto sopra il workflow, sempre visibile.



Mostra una barra di progresso orizzontale su tutta la pipeline (frame quality → SfM → training) con stage allocation: frame quality occupa 0–5% (fase 3.11, molto breve), SfM occupa 0–30% della barra, training 30–100%. Accanto testo di stato e visualizzazione percentuale nominata per fase («SfM 41%», «Training 12 500/20 000»), in modo che gli utenti non leggano come errore il passo indietro apparente «41% SfM → 25% training» — la barra mostra il progresso dell'intera pipeline, non la sub-stage. Il calcolo ETA inizia non appena è misurato abbastanza training tempo (tipicamente dopo le prime 100 iterazioni). La stessa visualizzazione è anche usata in Modalità esperto sopra l'Inspector.

#### IN PAROLE SEMPLICI

La barra stretta in cima è la tua mappa attraverso il workflow. Non ti dice solo cosa sta facendo l'app (allineare fotocamere, training in corso, ...), ma anche quanto è già avanti nel complesso. La suddivisione è deliberatamente fatta in modo che il calcolo della fotocamera occupi il primo terzo della barra e il vero training i due terzi successivi — altrimenti sembrerebbe che il progresso sia improvvisamente tornato a zero dopo SfM. Puoi quindi rilassarti, un'occhiata alla barra basta per vedere la fase approssimativa. Il testo accanto ti dice se sei attualmente nello stadio SfM (ad es. «SfM 41%») o nel training (ad es. «Training 12 500/20 000»), in modo che i numeri non sembrino confusi. Se l'ETA non viene mostrato, il training è semplicemente ancora troppo giovane — l'app stima solo appena ha misurato abbastanza velocità.

**C-03 DropZoneView (area trascinamento)**

Lato sinistro del passaggio di import, grande rettangolo tratteggiato con simbolo. Mostrato in Modalità principiante con l'etichetta «Drop photos or a video here».



Area di drop che fa saltare brevemente il simbolo e colora lo sfondo, non appena gli elementi di drag si librano sopra il campo. Accetta JPG, PNG, TIFF, HEIC, MP4, MOV, PLY, SPZ, .splat, bundle .radiance-scene e directory. Drop routing per tipo: le immagini vengono raccolte e passate ordinate, i video attivano il percorso di frame sampling, i file splat aprono direttamente l'anteprima, i bundle di scena vengono caricati. Le directory vengono enumerate e tutte le immagini contenute importate. I bookmark security-scoped per l'accesso sandbox-compatibile vengono correttamente ripresi e rilasciati. Le estensioni non supportate vengono mostrate come banner di avvio per 5 secondi.

**IN PAROLE SEMPLICI**

Il grande campo tratteggiato è il controllo principale del primo passaggio. Trascina semplicemente foto o un video al suo interno, o un'intera cartella — l'app prende tutto ciò che conosce e ignora il resto. Quando il campo diventa blu e il simbolo salta brevemente, l'app ha riconosciuto il drag. Lascia, e l'import inizia subito: le immagini vanno nella lista a destra, i video attivano automaticamente il passo di frame sampling, e i file .ply / .spz / .splat già addestrati aprono direttamente l'anteprima. Se un formato non si adatta proprio (ad es. PDF o BMP), appare una breve nota in alto — l'app non inghiotte materiale sconosciuto in silenzio.

**C-05 Pulsante Browse Files**

All'interno della drop zone, pulsante prominente.



Pulsante che apre il dialogo file macOS con selezione multipla e i tipi di file JPG, PNG, TIFF, MP4, MOV, cartelle e il formato scena proprio dell'app. Gli URL risultato sono security-scoped e vengono inoltrati attraverso gli stessi percorsi di import del trascinamento. Se l'utente seleziona una cartella, viene enumerata ricorsivamente per immagini.

**IN PAROLE SEMPLICI**

Se il trascinamento ti è scomodo, clicca semplicemente questo pulsante e naviga nel dialogo file macOS alle tue foto. Puoi scegliere più file contemporaneamente (Cmd-clic sulle singole immagini) o selezionare un'intera cartella — l'app cerca poi ricorsivamente nella cartella tutti i tipi di immagine supportati. È particolarmente pratico se le tue riprese si trovano annidate in sottocartelle (ad es. «shoot-day1/», «shoot-day2/») — un clic sulla cartella principale basta. Funzionalmente il pulsante fa esattamente ciò che fa il trascinamento; scegli semplicemente la strada che ti è più comoda.

**C-06 Pulsante Try Sample Scene**

All'interno della drop zone, visibile solo se il bundle dell'app contiene la scena di esempio e non sono ancora state importate immagini/splat.



Appare solo se (a) una `sample-scene.splat`, `.spz` o `.ply` è presente nel bundle dell'app E (b) nessuna immagine o video è ancora importato e nessuna nuvola di punti è presente. Al clic carica la nuvola di punti finita (preferibilmente il formato più piccolo — `.splat` ~3 MB, `.spz` ~1.4 MB, fallback `.ply`) e imposta dopo 400 ms valori di fotocamera hard-coded dai metadati originali della scena di fiori per una prospettiva di ingresso esteticamente sensata.

**IN PAROLE SEMPLICI**

Se avvii l'app per la prima volta e vuoi semplicemente vedere cosa esce alla fine — clicca qui. Apre una scena di fiori già addestrata che puoi subito ruotare ed esportare, senza che l'app debba calcolare. La fotocamera è preimpostata su una prospettiva di ingresso esteticamente sensata, in modo che tu veda subito qualcosa di bello. Perfetto per provare senza rischi il controllo 3D e il passo di esportazione, prima di affrontare le tue riprese. Non appena importi immagini proprie, il pulsante scompare automaticamente — viene mostrato solo finché il progetto è completamente vuoto.

**C-07 Pulsante Download Sample Photos**

All'interno della drop zone, accanto a «Try Sample Scene»; stesse condizioni di visibilità.



Attiva un download (repo [github.com/bkindler/radiancekit-sample-photos](https://github.com/bkindler/radiancekit-sample-photos)), che carica circa 427 MB di 960 frame a piena risoluzione e li alimenta nell'app. Durante il download il pulsante è disabilitato. Il progresso appare nella barra di progresso superiore come «Downloading X%» in uno stadio proprio, perché questo stadio mantiene la propria scala 0–100% e non si sovrappone allo stadio SfM successivo.

**IN PAROLE SEMPLICI**

Esattamente come la scena di esempio, solo con le foto di partenza invece del risultato finito. Così puoi far girare tu stesso l'intera pipeline una volta e vedere quanto durano realmente SfM e training sul tuo Mac. Il download è grande (circa mezzo DVD = 427 MB), ma avviene solo una volta — dopo le foto sono in locale e puoi riavviare la pipeline all'infinito con diversi preset. Mentre il download è in corso, la barra di progresso superiore mostra lo stato attuale del download in percentuale, in modo che tu possa stimare quando inizia. Suggerimento: prendi a tale scopo meglio un WLAN veloce o una rete via cavo — altrimenti i 427 MB si tirano.

### C-09 Picker Quality Presets



DOVE

Barra strumenti fluttuante inferiore dell'overlay di import, a sinistra del pulsante Start.



TECNICO

Controllo con etichetta «Quality» raggruppa i preset disponibili per categoria (Classic / MCMC / Custom). I preset integrati vengono raggruppati per categoria; gli header di sezione sono hard-coded. I preset Custom visibili solo se ne esistono. Stato bloccato: i preset non presenti nella lista free (Quick + Preview) ricevono un suffisso «🔒» al nome se l'utente non ha acquistato; alla selezione il picker torna a Preview e apre automaticamente il purchase sheet. Alla scelta il preset viene applicato, il che sostituisce l'intera configurazione di training.

#### IN PAROLE SEMPLICI

Qui scegli quanto precisamente e quanto a lungo l'app deve calcolare. «Quick» e «Preview» sono utilizzabili senza acquisto e forniscono in pochi minuti un primo risultato — ideale per testare se le tue immagini hanno in generale senso. «Balanced» e «Quality» hanno bisogno della versione completa e forniscono modelli decisamente più puliti, ma durano ore invece di minuti. MCMC è una strategia diversa che si accontenta di meno gauss-splat — buono se vuoi esportare il modello compatto in seguito o metterlo sul web. I preset premium li riconosci dal piccolo simbolo del lucchetto sul nome; se ne tocchi uno senza licenza, il picker torna a Preview e il purchase sheet si apre automaticamente. Regola pratica: parti sempre con Preview, guarda il risultato e decidi poi se vale la pena un run più lungo.

### C-10 Pulsante Start Processing



DOVE

Barra strumenti fluttuante inferiore dell'overlay di import, a destra del picker preset.



TECNICO

Pulsante che rimane grigio finché non sono importate né immagini né un video. Al clic avvia la pipeline e passa la macchina a stadi all'ordine frame quality → SfM → training. Il pulsante stesso non ha ulteriore stato; un'elaborazione in corso appare invece come schermo di elaborazione separato.

#### IN PAROLE SEMPLICI

Il pulsante «via». Finché è grigio, mancano ancora immagini di input o un video. Non appena hai trascinato dentro foto, diventa attivo e cliccandoci avvii SfM e training in sequenza. Da lì l'app prende in mano l'intero flusso e atterra automaticamente sullo schermo di elaborazione (Z2). Non devi cliccare altro — solo dopo la fine del training l'app passa di nuovo all'anteprima (Z3). Se cambi idea, puoi anche dopo interrompere in qualsiasi momento tramite Cancel.

**C-11** Slider Video Sampling

DOVE

Lista immagini destra, visibile solo se è stato importato un video (invece di immagini).



TECNICO

Slider da 0.5 fps a 30 fps in passi di 0.5. Alla modifica viene aggiornata la densità di frame e in più calcolato il numero di frame target (almeno 10) da densità e lunghezza video. Lo slider si trova al di fuori della lista delle immagini, perché gli elementi della lista bloccherebbero gli eventi del mouse degli slider. Sotto lo slider stanno i frame target calcolati («247 frames») e la lunghezza del video («1m23s video»). Il tooltip avverte: «Doubling the density doubles the number of frames and increases SfM time by ~100%.»

**IN PAROLE SEMPLICI**

Se invece di foto hai importato un video, questo slider decide quante singole immagini l'app deve estrarre dal video. Più immagini = qualità migliore, ma linearmente più tempo di calcolo. Per un video orbit da 30 secondi 5 fps (150 immagini) è un buon inizio; con riprese di 1 minuto spesso basta 3 fps. Sotto il regolatore l'app mostra in tempo reale quanti frame escono con l'impostazione attuale — così vedi subito se colpisci la zona sensata di circa 100–300 immagini. Se il risultato diventa scadente, tira il regolatore a destra e riprova; il raddoppio del frame rate raddoppia però anche grossolanamente la durata SfM.

**C-12** Pulsante Clear All

DOVE

Lista immagini destra, in basso a destra; visibile solo se sono state importate immagini.



TECNICO

Pulsante rosso. Il clic apre una finestra di conferma con titolo «Clear all imported files?» e messaggio «N images will be removed.». La conferma svuota tutte le immagini/video importati, le directory di staging, la nuvola di punti, lo stato del training, il risultato SfM e tutte le cache; lo stadio torna a Import. Su Cancel rimane tutto. La finestra è configurata come percorso default non distruttivo (pulsante distruttivo marcato in rosso).

**IN PAROLE SEMPLICI**

Se vuoi ricominciare completamente, clicca qui. La domanda di conferma appare perché l'eliminazione scarta tutti gli import attuali compresi fotocamere e risultati di training eventualmente già calcolati — non puoi annullarla. Sensato se vuoi sostituire completamente il materiale fotografico scelto o liberarti di un vecchio progetto prima di iniziarne uno nuovo. Nota: rimuovere una singola immagine si fa tramite la lista a destra (vedi punto successivo), non tramite questo pulsante. I tuoi file sul disco non vengono eliminati — l'app dimentica solo i suoi riferimenti.

**C-13 File List ForEach (rimozione immagine singola)**

DOVE

Lista immagini destra, ogni voce.



TECNICO

Lista sulle immagini importate con swipe per eliminare. Per immagine una riga con icona, nome file, risoluzione («1920 × 1080») e dimensione file (formattata KB/MB). La risoluzione proviene da una cache di metadati che viene riempita in modo asincrono dagli header delle immagini, in modo che l'interfaccia non si blocchi. L'azione di eliminazione offre swipe delete tipico macOS (swipe trackpad a sinistra su una riga) così come delete da tastiera con riga selezionata. Nota: il percorso esteso di eliminazione immagine con pulsante meno esplicito, Backspace e Cmd-Z per annullare è stato aggiunto *solo in Modalità esperto* nel Project Navigator — in Modalità principiante rimane swipe delete.

**IN PAROLE SEMPLICI**

La lista a destra mostra ogni immagine importata con risoluzione e dimensione del file — pratico per vedere a colpo d'occhio se hai mischiato insieme materiale ad alta e bassa risoluzione. Per rimuovere una singola immagine, fai swipe a sinistra con due dita sul trackpad — come in iOS Mail — o selezionala e premi Delete. L'app non elimina il file stesso; lo toglie solo dal progetto attuale. Se hai bisogno di un vero pulsante meno o annullamento Cmd-Z, passa in Modalità esperto (Cmd+2), lì c'è nel Project Navigator. In Modalità principiante rimane deliberatamente il semplice pattern di swipe.

**C-15 Validation Warnings (3 livelli)**

DOVE

Sotto la lista delle immagini, sopra il pulsante Clear All.



TECNICO

Tre soglie consecutive basate sul numero di immagini importate (attive solo se sono presenti immagini e nessun video): - < 3 immagini: banner rosso (octagon rosso), testo «At least 3 images are required. Camera alignment cannot be computed from fewer images.» - 3–9 immagini: banner rosso, testo «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 immagini: banner arancione (triangolo warning), testo «Workable, but quality usually improves with 20+ images and good coverage around the scene.»

Da 20 immagini il banner sparisce. Le soglie sono hard-coded e basate su 560+ esperimenti empirici di training.

**IN PAROLE SEMPLICI**

L'app guarda quante immagini hai importato e ti dà una valutazione colorata. Rosso significa: con alta probabilità non succede nulla — o SfM non può calcolare fotocamere o il training va in overfitting su troppo poco materiale. Arancione significa: potrebbe funzionare, ma non aspettarti una qualità top, perché l'algoritmo trova poco overlap tra le immagini. Nessun banner significa: buone premesse, hai abbastanza materiale. Se vuoi davvero modelli puliti, mira ad almeno 30–50 riprese distribuite uniformemente attorno al tuo soggetto — volentieri anche decisamente di più con scene esterne o grandi ambienti. Puoi avviare nonostante l'avviso, ma non sorprenderti se SfM si interrompe senza commenti o il modello appare bucato.

**C-16 COLMAP Workspace Detection** **DOVE**

Al drop di una cartella — nessun pulsante visibile, ma logica di riconoscimento.

 **TECNICO**

Al drop di una directory viene controllato se al suo interno si trova uno dei tre layout di workspace canonici: `sparse/0/cameras.bin`, `sparse/cameras.bin` o direttamente `cameras.bin` nella root. Se sì, l'enumerazione standard delle immagini viene interrotta e viene invece aperto un alert modale che chiede all'utente se usare la ricostruzione esistente o rimandare le immagini attraverso Apple Photogrammetry. Stesso percorso anche per workspace in formato testo (`cameras.txt`) ed export ETH3D. Vedi Capitolo 9 Backend Q6 per dettagli. Funziona in Modalità principiante esattamente come in Modalità esperto.

 **IN PAROLE SEMPLICI**

Se hai già lavorato con MetaShape, RealityCapture o COLMAP e hai fatto girare lì il calcolo delle fotocamere, puoi semplicemente trascinare la cartella di export qui. RadianceKit riconosce automaticamente dal contenuto che si tratta di un workspace COLMAP (verifica `sparse/0/`, `cameras.bin` ecc.) e ti chiede se assumere il calcolo finito o calcolare da solo nuovamente. Assumere risparmia con scene grandi ore di attesa, perché SfM viene completamente saltato — il training parte subito. Anche workspace in formato testo (`cameras.txt`) ed export ETH3D vengono riconosciuti. Questa funzione è disponibile in Modalità principiante esattamente come in Modalità esperto; più dettagli sono nel Capitolo 9 sotto Backend Q6.

**Quando al prossimo stadio?**

Puoi cliccare Start Processing non appena (a) almeno un'immagine o un video è importato e (b) il banner di validazione è arancione o sparito. Con banner rosso l'app ti lascia comunque avviare, ma puoi con alta probabilità interrompere subito l'elaborazione. Consigliato: almeno 20 immagini, nitide, con notevole overlap tra le riprese consecutive, tutte da circa la stessa distanza dal soggetto. Scegli prima dell'avvio un preset che si adatta al tuo budget di tempo — con 30 immagini e preset Quick sei pronto in pochi minuti, con Quality dura piuttosto 1–2 ore.

## Z2 — Elaborazione (SfM + training)

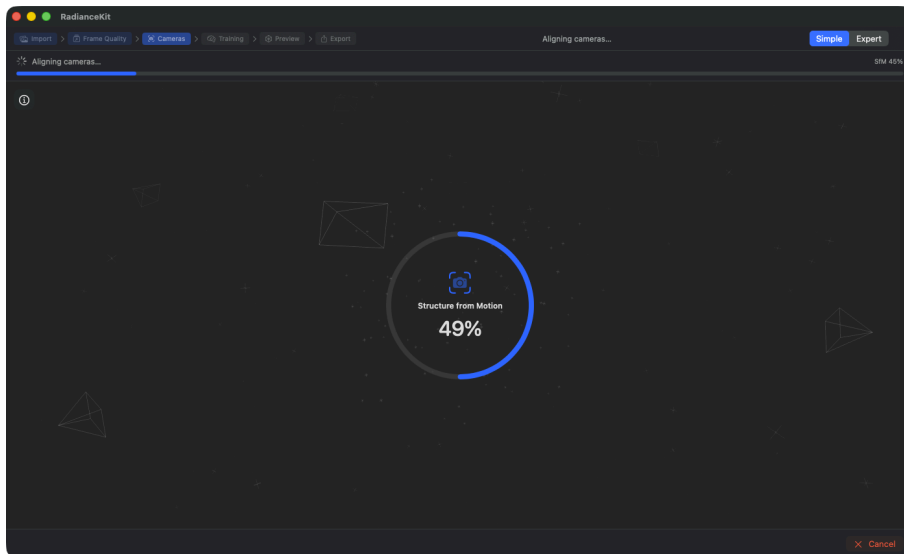


Figura 36: Z2 fase SfM — stage icon «Structure from Motion» con 41% nel grande cerchio, barra di stato superiore a «SfM 25%», pulsante Cancel in basso a destra

**Fase SfM (le fotocamere vengono allineate):** Il grande cerchio di progresso mostra il sub-stage progress (qui 41% della sessione Apple Photogrammetry in corso). Testo di stato «Aligning cameras...» in alto a sinistra. Il crumb trail segna «Cameras» come stadio attivo. La barra di stato superiore mostra il progresso totale della pipeline (25%) — SfM occupa la prima metà della barra. Fotocamere wireframe fluttuanti sullo sfondo indicano che le pose vengono stimate.

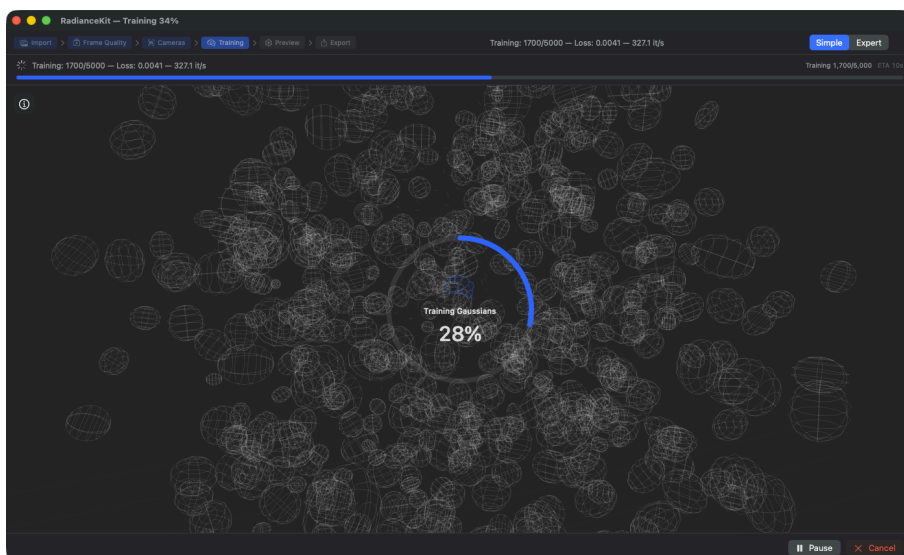


Figura 37: Z2 fase di training — stage icon «Training Gaussians» con 6%, metriche live in alto (Training: 400/5000 — Loss: 0.1642 — 138.7 it/s), ETA 33s, Pause/Cancel in basso

**Fase di training (le gaussiane vengono ottimizzate):** Il sub-stage icon passa a «Training Gaussians», la percentuale conta le iterazioni dal preset scelto (qui 400 / 5 000 per preset Preview = 8% dello stadio). La riga delle metriche live mostra il valore loss (0.1642), iterazioni al secondo (138.7 it/s) ed ETA (33 s). Il progresso totale della

pipeline sale dal 50% al 100% durante questa fase. Il pulsante Pause (invece di Cancel-only nella fase SfM) permette Resume in seguito; Cancel scarta il risultato del training e torna a Z1.

Non appena la pipeline gira, l'app nasconde l'overlay di import e mostra uno schermo di elaborazione a tutta pagina. Al centro gira un grande cerchio di progresso (220 × 220 pixel) con stage icon, testo di stato e cifra percentuale; sullo sfondo un'animazione discreta di splat visualizza simbolicamente il calcolo in corso. In alto a sinistra può essere mostrato un pannello di info che mostra metriche live da training e SfM. In basso ci sono Pause/Resume, Cancel e in caso di errore un pulsante Retry.

### C-18 SplatTrainingView (animazione di sfondo)

#### DOVE

Sfondo a tutta pagina dietro il cerchio di progresso, nascosto in caso di interruzione o errore.

#### TECNICO

Animazione decorativa che a seconda del progresso della pipeline (0...1) renderizza un numero crescente di piccole particelle splat animate. La fonte è un valore di progresso calcolato che mappa le fasi SfM su 0-0.2 e il training su 0.2-1.0 (frame quality su 0-0.05). Così gli splat si «costruiscono» visibilmente mentre il training gira. Esclusivamente decorativo — la visualizzazione non mostra risultati intermedi reali del training attuale (sarebbe live preview in Modalità esperto). In caso di cancel o failure viene nascosta e rimane visibile solo il cerchio di stato.

#### IN PAROLE SEMPLICI

Sullo sfondo gira una piccola animazione di punti danzanti, in modo che lo schermo non sembri così vuoto durante il calcolo. Non è il tuo vero modello 3D — lo vedi solo dopo il training nel passo Z3. L'animazione ha però la stessa tonalità, così che dal grado approssimativo di addestramento puoi leggere quanto è avanzato il training. All'inizio sono visibili solo pochi punti, verso la fine lo sfondo si riempie notevolmente più denso — un grazioso indicatore visivo aggiuntivo alla visualizzazione percentuale nel cerchio. Se l'animazione ti dà fastidio (ad es. perché vuoi lavorare in parallelo sullo sfondo), puoi passare in Modalità esperto, dove viene a mancare.

**C-19 Grande cerchio di progresso**

DOVE

Al centro dello schermo di elaborazione, 220 × 220 pixel.



TECNICO

Due anelli renderizzati sovrapposti: all'esterno un anello traccia attenuato, all'interno un anello di progresso pieno con stroke di accento o rosso (rosso in caso di errore). All'interno del cerchio uno stage icon (cervello per training, fotocamera per SfM, film per estrazione frame video, sparkles per frame quality), titolo dello stadio e la cifra percentuale animata live in font rounded da 32 pt. L'icona pulsa delicatamente finché l'elaborazione è attiva. La visualizzazione interpola su un timer 30 Hz dolcemente verso il vero progresso attuale — con constant creep (0.0003/frame) più una quota proporzionale (4% del gap) e un soft ceiling che si imposta all'80% del prossimo milestone atteso (per SfM da una tabella hard-coded di milestone). Così il progresso appare fluido, anche se gli update SfM reali arrivano solo ogni pochi secondi.

**IN PAROLE SEMPLICI**

Il grande cerchio al centro è la tua visualizzazione principale mentre l'app calcola. Si riempie dolcemente, anche se gli update di calcolo reali arrivano solo ogni pochi secondi — ciò ti dà la sensazione che stia accadendo qualcosa, invece di fissare per minuti una percentuale congelata. Il simbolo al centro cambia a seconda che vengano estratti frame (icona film), vengano allineate fotocamere (icona fotocamera) o vengano addestrate gaussiane (icona cervello). La cifra percentuale si riferisce al sotto-step attuale — la pipeline totale la vedi nella barra stretta in alto. In caso di errore l'anello diventa rosso invece di blu, e l'icona non pulsa più, così che noti subito che qualcosa è andato storto.

**C-22 Pulsante Info (mostrare metriche)**

DOVE

In alto a sinistra sullo schermo di elaborazione, 32 × 32 pixel.



TECNICO

Pulsante semplice con sfondo materiale. Attiva o disattiva il pannello di info. L'icona cambia tra info-cerchio outline e info-cerchio pieno quando attivo. Animazione di ingresso delicata. Nel tooltip «Show detailed processing metrics».

**IN PAROLE SEMPLICI**

Di default lo schermo è deliberatamente ordinato — solo il grande cerchio di progresso, di più non vedi all'inizio. Se come utente tecnicamente interessato vuoi sapere più precisamente cosa succede (quale iterazione, quanto è alto il loss, quante gaussiane), clicca sul simbolo i in alto a sinistra. Un piccolo pannello si apre in basso e mostra tutti i valori live. Un nuovo clic lo nasconde di nuovo. L'impostazione non è persistente — ad ogni nuova esecuzione di training il pannello è di nuovo nascosto, deliberatamente scelto così per non spaventare i principianti.

**C-23** Pannello info (metriche live)

In basso a sinistra sullo schermo di elaborazione, visibile solo se `showProcessingInfo == true`.



Pannello a due colonne con sfondo ultra-thin material. Colonna sinistra: righe info specifiche per stadio — per SfM testo di stato e percentuale; per training iterazione, loss combinato, loss L1, loss D-SSIM, gaussian count (colorato arancione), speed (it/s), elapsed time, ETA calcolato, SH degree e learning rate. Colonna destra: testo di stato, stringa time info, chart di loss inline (vedi C-28) e un nudge di discoverability (vedi C-32). Tutti i valori vengono letti dallo stato del training, aggiornato ad ogni tick di training.

**IN PAROLE SEMPLICI**

Il pannello info mostra tutti i valori live che in Modalità esperto sarebbero permanentemente nella sidebar Inspector: iterazione attuale, valore loss (più piccolo = meglio), numero di gaussiane, velocità, tempo residuo stimato, SH degree e learning rate. Sul lato destro gira inoltre una piccola curva loss che ti rivela a colpo d'occhio se il training sta andando nella direzione giusta. Se il training sembra fiacco, un'occhiata qui aiuta — un loss che non scende più o un ETA che non diminuisce più indicano problemi. Se il loss esplode (diventa improvvisamente enorme) o mostra NaN, il training è diventato instabile e un cancel + retry o un cambio a un altro preset è sensato.

**C-25** Pulsante Pause/Resume

Barra di navigazione inferiore, visibile solo durante lo stadio di training (NON durante SfM) e finché l'elaborazione gira.



Bordered button. A seconda dello stato chiama Pause o Resume. L'etichetta cambia tra «Pause» (con icona pause) e «Resume» (icona play). Durante il passo SfM il pulsante non viene mostrato, perché Apple Photogrammetry non conosce semantica di pausa. Lo stato di pausa preserva completamente iterazione, stato gaussian e momentum optimizer — Resume prosegue dove ci si è fermati.

**IN PAROLE SEMPLICI**

Mentre il training gira, puoi metterlo in pausa in qualsiasi momento e riprenderlo in seguito. Sensato se vuoi fare nel frattempo qualcos'altro sul Mac che ha bisogno di molta GPU — ad es. montaggio video, test di giochi o un export di render da un'altra app. Clicca Pause, fai la tua cosa, clicca Resume, il training prosegue esattamente dove era. Contatore di iterazione, numero di gaussiane e momentum optimizer rimangono completamente preservati, lo stato di pausa non ti costa nulla in qualità. Durante la fase SfM Pause non è disponibile — Apple Photogrammetry non conosce funzione di pausa, lì in caso di necessità devi lavorare con Cancel.

**C-26 Pulsante Cancel**

Barra di navigazione inferiore, visibile mentre l'elaborazione gira (SfM o training).



Pulsante bordered rosso. Apre una finestra di conferma con titolo «Stop and discard progress?», pulsanti «Discard Progress» (distruttivo) e «Keep Running» (cancel). Alla conferma viene impostato il flag cancel, il task di training terminato, il sottoprocesso SfM se necessario terminato e una riga di summary con status di interruzione scritta nel log JSONL. A differenza di Pause i buffer di training e lo stato vengono scartati.

**IN PAROLE SEMPLICI**

Il pulsante di interruzione. A differenza di Pause è definitivo — se vuoi poi ricominciare, l'elaborazione gira da capo, tutte le iterazioni già addestrate sono perdute. Sensato se hai sbagliato il preset, il training gira troppo lentamente o l'app produce ovviamente risultati di scarto e non vuoi aspettare. Prima della vera interruzione l'app chiede di nuovo tramite una finestra di conferma, in modo che tu non perda per errore ore di tempo di calcolo. Se vuoi solo interrompere brevemente, prendi piuttosto Pause.

**C-27 Pulsante Retry**

Barra di navigazione inferiore, visibile se la pipeline è fallita (lo stato SfM inizia con «SfM failed» o il training è in stato di errore).



Pulsante accent. Avvia di nuovo l'intera pipeline. Prima dell'avvio viene controllato se sono ancora presenti immagini/video importati. I log degli errori precedenti rimangono nella directory JSONL; un nuovo run scrive un nuovo file di log con timestamp attuale.

**IN PAROLE SEMPLICI**

Se SfM o training si interrompe con un messaggio di errore, puoi riprovare qui. A volte aiuta, perché molti passaggi (RANSAC, densification) hanno componenti casuali e un secondo tentativo può avere successo dove il primo è fallito. L'intera pipeline gira poi nuovamente da capo — SfM e training, in un fresco file di log JSONL. Se anche il secondo tentativo fallisce, di solito le immagini di input sono il problema (troppo poche, troppo poco overlap, sfocatura da movimento, illuminazione scadente); torna allora indietro con Back e sostituisci il tuo materiale. Suggerimento: guarda in parallelo nei log di training (Help → Open Training Logs), lì sta più dettagliatamente dove si è bloccato concretamente.

**C-28** Chart loss inline

Nel pannello info, colonna destra, visibile solo durante training con storia di andamento non vuota.



Area di disegno compatta (40 pixel di altezza), disegna la loss history come linea da 1 pixel in colore accent. I dati vengono filtrati su valori finiti (protezione NaN per training instabili). Min/max vengono calcolati sull'intera history — il chart fa quindi auto-zoom sull'intervallo dei valori. L'ultimo valore loss sta in alto a destra sopra il chart. La history stessa viene costruita nello stato dell'app ad ogni tick di training (tipicamente ogni 100 iterazioni).

**IN PAROLE SEMPLICI**

Una piccolissima curva loss che ti mostra a colpo d'occhio se il training «converge» (linea scende a destra) o se si blocca o esplose (linea piatta o sale). Con un training sano la linea scende ripida all'inizio e poi si appiattisce — è l'andamento atteso, simile a una curva di dimezzamento. Il chart fa zoom automatico sull'intervallo di valori attuale, in modo che anche piccoli miglioramenti alla fine del training rimangano visibili. Se la linea schizza improvvisamente in alto o si congela, è un buon segnale che qualcosa non va — o il materiale è problematico o un altro preset sarebbe più adatto. Il chart lo trovi nel pannello info, che mostri in alto a sinistra con il simbolo i.

**C-32** Discoverability Nudge (suggerimento Modalità esperto)

Nel pannello info, colonna destra in basso, visibile solo durante training E in Modalità principiante.



Piccola riga con icona occhio e testo caption «Switch to Expert Mode (⌘2) for live splat preview», in tonalità sobria e font 10 pt. Nessun elemento interattivo, solo nota. Non reagisce al clic — l'utente deve effettivamente premere Cmd+2 o cliccare il menu Mode → Expert Mode.

**IN PAROLE SEMPLICI**

Una nota discreta che in Modalità esperto durante il training la versione intermedia attuale del tuo modello 3D è visibile dal vivo nel viewport. In Modalità principiante viene deliberatamente nascosta per mantenere l'interfaccia calma — ma molti utenti non sanno affatto che questa funzione esiste, quindi qui lo segnaliamo dolcemente. Premi Cmd+2 e il training prosegue in background mentre puoi guardare come il tuo modello si compone davanti ai tuoi occhi. È anche un buon strumento per stimare già dopo poche migliaia di iterazioni se il risultato diventerà qualcosa, o se preferisci interrompere e ricominciare. Cmd+1 ti riporta in qualsiasi momento alla vista principiante.

## Quando al prossimo stadio?

L'app passa automaticamente a Z3 (anteprima) non appena il training è terminato con successo — non devi cliccare nulla. La barra di navigazione inferiore passa poi da Pause/Cancel a un pulsante Back (torna a Import) e un pulsante Export (avanti a Export). In caso di errore (messaggio di errore rosso, stage icon è X) appare invece Retry, e devi decidere se riavviare o tornare a Import con Back per modificare il materiale.

## Z3 — Anteprima (ruotare il modello 3D)

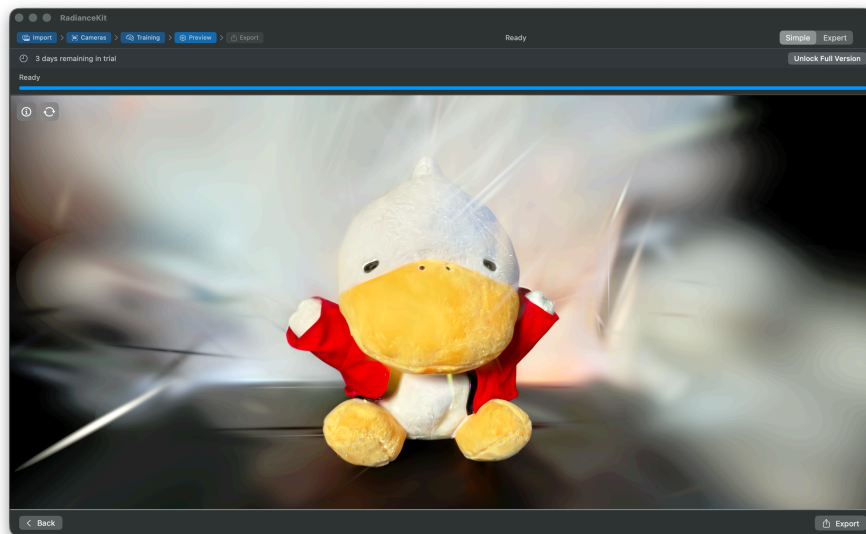


Figura 38: Passaggio anteprima Modalità principiante con viewer 3D

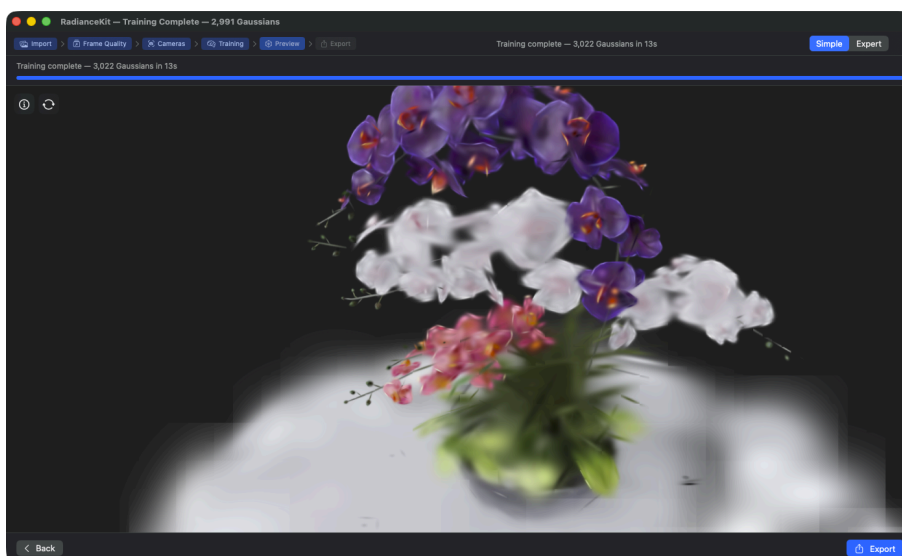


Figura 39: Z3 anteprima dopo il completamento del training — bouquet ricostruito, header mostra «Training complete — 3.022 Gaussians in 13s», pulsanti Back ed Export in basso

**COSA MOSTRA L'IMMAGINE** Il crumb trail segna «Preview» come stadio attivo. Viewport 3D a tutta pagina renderizza la scena del bouquet finita di training (set di test sintetico,

sottoinsieme di 60 frame da 960 cam emisferiche). Barra di stato header: «Training complete — 3 022 Gaussians in 13 s» — indica il numero finale di gaussiane e il tempo di training. Il drag nel viewport ruota la fotocamera (yaw/pitch); lo scroll-wheel fa zoom lungo la direzione di vista. Il pulsante «Back» (in basso a sinistra) torna a Z2 per resume o re-run; il pulsante «Export» (in basso a destra, primary) naviga avanti a Z4.

Dopo il completamento del training l'app atterra automaticamente nell'anteprima. Qui vedi il tuo modello Gaussian Splatting finito in una vista Metal a tutto schermo e puoi ruotarlo, zoomarlo e panare con mouse e trackpad. Sulla parte superiore del viewport si trova un piccolo overlay con controllo fotocamera e info — auto-rotation, statistiche di training, pulsante reset. Prima del prossimo passo (export) è sensato controllare il modello da vari angoli, per assicurarsi che la ricostruzione sia pulita.

### C-36 SplatViewportView (vista 3D principale)

#### DOVE

Sfondo a schermo intero del passaggio anteprima.

#### TECNICO

Viewport 3D basato su Metal che renderizza la nuvola di punti finita. Il renderer è il rasterizzatore ForwardPass PROPRIO di RadianceKit — lo stesso che renderizza già gli splat durante l'addestramento — quindi è vero WYSIWYG (ciò che viene addestrato viene mostrato ed esportato esattamente allo stesso modo). Pipeline di rendering basata su tile con order-independent transparency. Se il renderer non può essere inizializzato (ad es. perché Metal non è disponibile sul sistema), appare invece uno sfondo nero con testo «Metal not available». La vista ignora la safe area, in modo che il modello arrivi fino al bordo della finestra.

#### IN PAROLE SEMPLICI

Il viewport principale. Qui vedi il tuo modello 3D finito ricostruito dalle tue foto, renderizzato sulla GPU in tempo reale. Clicca e trascina con il pulsante sinistro del mouse per ruotare. Rotella di scorrimento o gesto trackpad con due dita per zoomare. Pulsante destro del mouse o Cmd+drag per panare. Il modello è composto da decine di migliaia di ellissoidi 3D semi-trasparenti («gaussiane») che ricostruiscono fotorealisticamente la tua scena — ognuno ha una posizione, orientamento, forma e colore che il training ha appreso. Nel raro caso che il tuo Mac non supporti Metal, vedi invece uno sfondo nero con un messaggio di nota — RadianceKit ha bisogno obbligatoriamente di una GPU Metal-capable.

**C-37 CameraControlsOverlay (overlay di controllo)**

Sopra il viewport, fluttuante.



Overlay UI compatto con pulsanti per auto-rotation (turntable), reset-camera, scelta dello sfondo (Gray/Black/White), save screenshot, toggle info panel. Si lega ai parametri della fotocamera (distanza, azimuth, elevazione, target, FOV) e controlla l'auto turntable. Durante il training (quando l'utente vuole vedere girare con sé il viewport in Modalità esperto) l'overlay mostra in più una compatta riga di stato del training.

**IN PAROLE SEMPLICI**

La piccola barra fluttuante sopra il modello. Qui avvii l'auto-rotation (il modello ruota da solo, buono per screenshot e brevi demo), riporti la fotocamera tramite reset alla posizione iniziale (se ti sei perso), cambi lo sfondo (grigio per neutro, nero per massimo contrasto, bianco per modelli chiari), e fai direttamente screenshot, che vengono salvati sotto /Pictures. Pratico se vuoi mostrare un certo dettaglio da un angolo molto preciso, senza esportare tutto il modello. L'auto-rotation è anche un buon test se il modello appare ugualmente buono da tutti i lati o se c'è un «lato sporco» sorto da riprese mancanti.

**C-38 Pulsante Export (barra di navigazione)**

Barra di navigazione inferiore in Z3.



Pulsante accent con etichetta «Export» e icona share. Il clic attiva il passaggio a Z4. Prima la vista superiore controlla se la versione completa è sbloccata — se no, viene mostrata la vista di blocco invece dello stadio di export (vedi U-06).

**IN PAROLE SEMPLICI**

Se sei soddisfatto del risultato, clicca Export e atterri nell'ultimo passaggio, dove scegli il formato e salvi. Senza versione completa acquistata atterri invece su un blocco schermo con suggerimento di sblocco e pulsante di acquisto — l'app non vuole importi la versione completa, ma l'export è una delle funzioni premium. Non appena hai completato l'acquisto, l'app prosegue direttamente nello stato sbloccato e atterri nello stadio di export usuale. Se cambi idea, tramite il pulsante Back torni all'anteprima e puoi continuare a ruotare il modello.

## Quando al prossimo stadio?

Prima di esportare, ruota il modello una volta completamente e verifica: sono presenti tutte le aree che hai coperto nelle tue immagini di input? Ci sono «floater» fluttuanti (nuvole di splat liberamente fluttuanti nell'aria)? Lo sfondo/cielo appare pulito o sbavato? Problemi gravi possono essere risolti solo con un nuovo training — o con più immagini, un altro preset, o in Modalità esperto con le impostazioni di floater reduction.

## Z4 — Esportazione (scegliere formato e salvare)

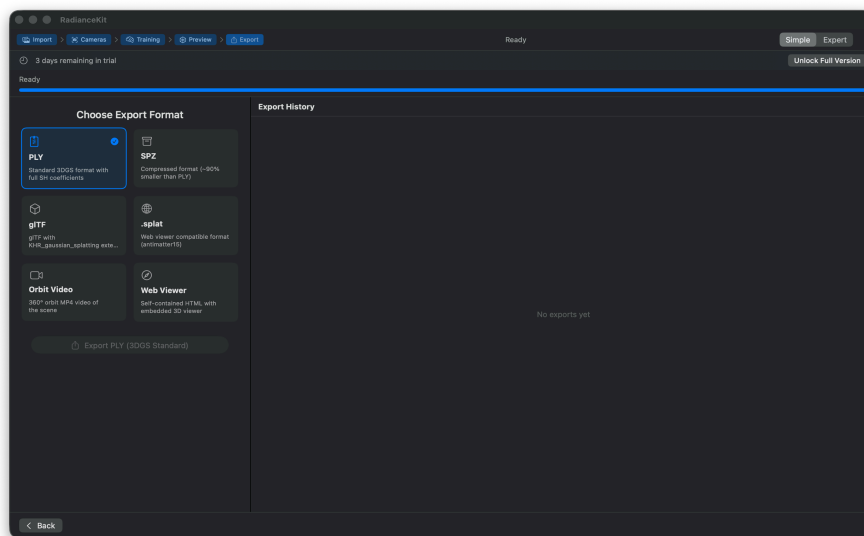


Figura 40: Passaggio esportazione Modalità principiante con schede formato

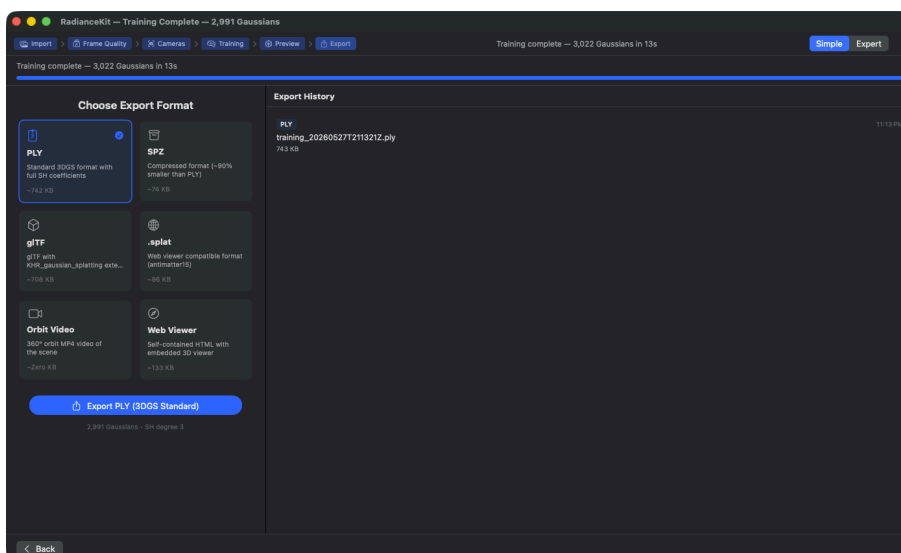


Figura 41: Z4 schede esportazione — 6 formati (PLY 742 KB selezionato, SPZ 74 KB, glTF 708 KB, .splat 96 KB, Orbit Video, Web Viewer 133 KB), sidebar Export History a destra con PLY già esportato

**COSA MOSTRA L'IMMAGINE** Il crumb trail segna «Export» come stadio attivo. Griglia di schede a sinistra «Choose Export Format» con tutte e sei le opzioni: PLY (standard 3DGS, 742 KB, con coefficienti SH completi — qui preselezionato con segno di spunta blu), SPZ (formato 3DGS compresso, ~90% più piccolo di PLY, 74 KB), glTF (con estensione `KHR_gaussian_splatting`, 708 KB), `.splat` (compatibile con viewer web via `antimatter15`, 96 KB), Orbit Video (MP4 360° della scena, calcolo dimensione live), Web Viewer (HTML autonomo con viewer 3D incorporato, 133 KB). Le indicazioni di dimensione vengono calcolate live dal gaussian count attuale e dall'overhead del formato. A destra «Export History» elenca i export già completati con pillola formato, nome file e timestamp — il clic rivela nel Finder. Primary CTA in basso a sinistra: «Export PLY (3DGS Standard)» con sottotitolo gaussian «2,991 Gaussians · SH degree 3».

Nell'ultimo passaggio scegli tra 6 formati di esportazione (PLY, SPZ, glTF, `.splat`, orbit video, web viewer) tramite griglia di schede a 2 colonne, clicchi Export e scegli il luogo di salvataggio nel dialogo macOS. A destra gira una cronologia di tutti gli export precedenti — alla selezione delle schede viene mostrata immediatamente sotto ogni scheda la dimensione del file stimata, in modo che tu preferisca ad es. SPZ se vuoi andare nel web (piccolo) e PLY se vuoi importare in un altro software (SuperSplat, Postshot, Blender via plugin) (grande e completo).

### C-39 Griglia formato a 2 colonne

#### DOVE

Lato principale sinistro del passaggio di esportazione.

#### TECNICO

Griglia di schede con due colonne flessibili e 12 punti di spaziatura. Itera sui formati offerti in Modalità principiante — un sottoinsieme filtrato della lista completa dei formati che contiene solo i 6 formati più importanti: PLY, SPZ, glTF, `.splat`, orbit video, web viewer. Compressed PLY e SOG vengono offerti SOLO in Modalità esperto.

#### IN PAROLE SEMPLICI

Una griglia di schede con i 6 formati rilevanti in Modalità principiante: PLY (formato standard per altri tool 3D), SPZ (variante compressa per web), glTF (standard ufficiale Web3D), `.splat` (per viewer web `antimatter15`), orbit video (MP4 finito da mostrare) e web viewer (file HTML autonomo con player 3D incorporato). Con questi copri il 90% dei casi d'uso. Se hai bisogno di uno dei formati meno comuni (Compressed PLY o SOG per compressione estrema), passa in Modalità esperto, lì sono disponibili tutti gli 8 formati. La selezione compatta qui è deliberata, in modo che i principianti non siano sopraffatti dalla varietà.

**C-40 Pulsante scheda formato**

Ogni scheda nella griglia.



Pulsante semplice con layout a scheda: icona (ad es. zipper documento per PLY, scatola di archivio per SPZ, icona video per orbit video) in alto, nome del formato come headline, caption di descrizione (accorciata su 2 righe), sotto la dimensione del file stimata (calcolata live da formato, gaussian count e SH degree e formattata come KB/MB). Al clic il formato viene selezionato. La scheda selezionata riceve sfondo accent, bordo accent e un'icona di spunta in alto a destra. Tooltip è la descrizione del formato.

**IN PAROLE SEMPLICI**

Una scheda per formato. Cliccane una, viene evidenziata con colore accent e una spunta, e il pulsante Export sotto adatta il suo testo («Export PLY», «Export SPZ» ecc.). Ogni scheda mostra un simbolo adatto, il nome, una breve spiegazione su due righe e la dimensione del file stimata con il tuo risultato di training attuale. La dimensione ti aiuta a scegliere sensatamente — se vuoi inviare il risultato per email, prendi la variante più piccola (di solito SPZ o .splat); se vuoi continuare a lavorare in un altro software 3D, prendi quella con la migliore compatibilità (tipicamente PLY). Passando il mouse su una scheda il tooltip mostra una descrizione più dettagliata, se trovi poco chiare le differenze tra i formati.

**C-41 Slider Video Duration**

Sotto la griglia formato, visibile solo se è selezionato un formato video (orbit video o social video).



Slider 3–30 secondi in passi di 1 secondo, lega alla durata video nello stato dell'app. Larghezza massima 300 pixel. Viene mostrato solo se è selezionato un formato video. Con formati non video lo slider viene completamente rimosso dalla vista — nessun spazio morto.

**IN PAROLE SEMPLICI**

Se scegli un orbit video come export, qui puoi determinare la lunghezza. 3 secondi = rotazione molto veloce, 30 secondi = rotazione lenta e calma attorno al tuo modello. Per reel social media (Instagram, TikTok) di solito 6–10 secondi è ideale — abbastanza lungo per mostrare il modello, abbastanza corto perché gli spettatori non saltino. Per presentazioni o video portfolio puoi tranquillamente prendere 15–20 secondi. Lo slider compare solo se è selezionato un formato video; con formati file come PLY o SPZ sarebbe senza senso ed è nascosto.

**C-42 Pulsante Export**

Sotto la griglia formato (e sotto lo slider duration se video selezionato).



Grande pulsante accent. Etichetta: «Export {nome-formato}», icona share. Al clic viene aperto il dialogo di salvataggio macOS con estensione adatta al formato e nome file predefinito «scene.{ext}»; alla conferma l'export viene scritto nell'URL scelto. Disabilitato se non è presente alcun risultato di training o se un export è già in corso.

**IN PAROLE SEMPLICI**

Clic, scegli luogo di salvataggio nel dialogo macOS, fatto — l'app scrive il file nel formato scelto nella posizione selezionata. Il nome di default è «scene.{estensione}» (ad es. «scene.ply» o «scene.spz»), che puoi modificare a piacere nel dialogo prima di salvare. Il pulsante è grigio finché non c'è un risultato di training (qui non dovrebbe mai succedere, altrimenti non saresti nel passo di export) o un altro export è già in corso. Non appena l'export gira, appare sotto un indicatore di progresso; l'app rimane utilizzabile, quindi puoi già preparare il prossimo export.

**C-43 Export Progress Bar**

Sotto il pulsante export, visibile solo mentre un export è in corso.



Indicatore di progresso con larghezza max 300 pixel, sotto caption «Exporting... N%». Il valore va da 0 a 1 e viene aggiornato durante la scrittura — con PLY in chunk di 10 000 gaussiane, con SPZ una volta dopo la quantizzazione, con orbit video in intervalli di frame.

**IN PAROLE SEMPLICI**

Mentre l'export gira, vedi qui il progresso come stretta barra più visualizzazione percentuale. PLY è solitamente pronto in pochi secondi, perché il file viene semplicemente scritto in binario. SPZ ha bisogno di un po' più tempo perché i dati vengono inoltre quantizzati e compressi. Orbit video è l'export più dispendioso in tempo — qui ogni singolo frame viene re-renderizzato; a seconda di risoluzione e lunghezza può durare un minuto o più. Durante l'export l'app rimane utilizzabile, quindi puoi già preparare il prossimo formato o continuare a cliccare nel viewport.

**C-44** Export Error Display

Sotto la progress bar, visibile solo se nell'ultimo export è avvenuto un errore.



Riga rossa con icona warning e testo errore. Sfondo rosso all'8% di opacità, angoli arrotondati. Larghezza max 400 pixel. Cause di errore frequenti: SOG si aspetta `cwebp` nel PATH di sistema (non conforme App Store); errore di scrittura con disco pieno; errore sandbox con destinazioni di salvataggio fuori dall'area consentita.

**IN PAROLE SEMPLICI**

Se l'export va storto, appare qui in rosso una breve descrizione in chiaro del problema. Di solito la causa è ovvia — niente spazio sul disco, niente permessi di scrittura per la cartella di destinazione o una destinazione fuori dalle aree consentite dal sandbox. In particolare con il formato SOG capita che `cwebp` manchi nel sistema; in questo caso SOG non è utilizzabile e devi ripiegare su SPZ. Se il messaggio di errore non è chiaro, guarda nella directory di log (Help → Open Training Logs), lì sta più dettagliatamente cosa è andato storto. In caso di dubbio aiuta semplicemente scegliere un'altra destinazione — ad es. la Scrivania.

**C-46** Lista Export History

Lato destro del passaggio di esportazione.



Lista sulla cronologia di esportazione (salvata in modo persistente come JSON in UserDefaults, gestita dopo ogni export riuscito). Ogni riga mostra badge formato (piccolo, colorato accent), timestamp (HH:mm), nome file (1 riga accorciata) e dimensione file formattata. Il clic su una riga apre il Finder con il file selezionato. Empty state: «No exports yet».

**IN PAROLE SEMPLICI**

Una lista dei tuoi export precedenti — formato, ora, nome file, dimensione, in ordine cronologico. Clicca una riga e il file viene mostrato evidenziato nel Finder, senza che tu debba navigare tu stesso attraverso le cartelle. Pratico se un'ora dopo hai di nuovo bisogno dell'ultimo export e non sai più dove l'hai salvato — la history se lo ricorda. Se non hai mai esportato nulla, qui sta una nota amichevole «No exports yet». La lista sopravvive ai riavvii dell'app, perché è salvata in UserDefaults.

**C-48 History Context Menu (clic destro)**

DOVE

Clic destro su una riga history.



TECNICO

Menu contestuale su ogni voce di lista con due azioni: «Reveal in Finder» (apre Finder con file selezionato, come il clic semplice) e «Copy Path» (mette il percorso completo del file come testo negli appunti). Quest'ultimo è utile per trascinarsi in altre app o per il passaggio alla riga di comando.

**IN PAROLE SEMPLICI**

Il clic destro su una voce history apre un piccolo menu con due azioni. «Reveal in Finder» fa lo stesso di un clic normale — apre il Finder con il file selezionato, in modo che tu lo veda subito. «Copy Path» mette il percorso completo del file negli appunti, in modo che tu possa incollarlo ad es. in comandi terminale, in altre app o in una nota. Particolarmente pratico se vuoi passare l'export a qualcuno o aprirlo in un altro programma che lavora con input di percorso. Funzionalmente un piccolo ma utile dettaglio che punta sui pattern operativi tipici Mac.

**Quando il workflow è concluso?**

Dopo un export riuscito hai il tuo modello 3D come file sul disco e la history mostra una nuova voce. Non c'è un pulsante «Done» — puoi aggiungere quanti export vuoi in diversi formati senza riaddestrare. Se vuoi tornare all'anteprima (ad es. per controllare ancora una prospettiva di fotocamera), usa il pulsante Back nella barra di navigazione inferiore. Se vuoi iniziare una scena completamente nuova, vai tramite Back fino a Z1 e usa lì Clear All, o File → New Project (Cmd+⇧+N).

**Passaggio a Modalità esperto**

Premi in qualsiasi momento Cmd+2 o scegli Mode → Expert Mode ( M8 ). L'intero stato viene preservato: immagini importate, preset scelto, training in corso o terminato, nuvola di punti finita, history di esportazione, persino lo stadio attuale. In Modalità esperto viene mostrata la sidebar Inspector completa con tutti i ~150 controlli invece dello stadio a quattro passaggi. In particolare: il Project Navigator (vedi Capitolo 2) offre le operazioni estese sulle immagini (pulsante meno, Backspace delete, Cmd-Z undo, anteprima Quick Look), la live preview nel viewport durante il training, così come tutti i parametri loss, MCMC, densification e Mip-Splatting. Cmd+1 torna in Modalità principiante — anche questo non perde alcun stato.

## Domande frequenti

### Perché il mio pulsante **Start Processing** rimane grigio?

Non hai ancora importato immagini o un video. Trascina almeno un file nella drop zone o usa «Browse Files». Non appena la lista delle immagini a destra contiene almeno una voce, il pulsante diventa attivo. (Con solo 1–2 immagini parte sì, ma SfM si interrompe direttamente con errore — vedi il banner di validazione rosso.)

### Perché il mio pulsante **Export** è bloccato?

In Modalità principiante ci sono due livelli: (a) Se la pipeline di training non è ancora terminata e non hai un, il pulsante è disabilitato — devi prima concludere Z2. (b) Se non hai ancora acquistato la versione completa (`PurchaseManager.hasAccess == false`), vedi invece dello stadio di export una vista di blocco con icona lucchetto e pulsante «Unlock Full Version», che apre il purchase sheet. I preset Quick e Preview permettono training gratuito, ma l'export è premium.

### Perché non posso scegliere un preset?

Puoi sceglierlo — ma se tocchi un preset premium (Balanced, Quality, varianti MCMC) senza versione completa acquistata, il picker torna automaticamente a Preview e il purchase sheet si apre. Quick e Preview sono gli unici preset utilizzabili gratuitamente.

### Perché la mia drop zone è vuota e tratteggiata grigia, anche se trascino dentro immagini?

Probabilmente un mismatch di tipo UTI. L'app accetta JPG, PNG, TIFF, HEIC, MP4, MOV più i formati splat propri dell'app. Altri formati di immagine (BMP, GIF, WebP, formati RAW) NON vengono riconosciuti. Se sei sicuro che il tuo tipo di immagine dovrebbe esserci, controlla l'estensione del nome file — l'app va primariamente per estensione, non per contenuto file.

### Perché SfM dura così tanto, anche se ho solo 30 immagini?

Apple Photogrammetry non scala in modo lineare — con certe costellazioni di immagini (interni con texture complesse, sfocatura da movimento, illuminazione scadente) ha bisogno di notevolmente più tempo di quanto il conteggio delle immagini faccia supporre. Se SfM è ancora bloccato dopo 10+ minuti con 30 immagini, interrompi e riprova con materiale migliore o passa in Modalità esperto e prova COLMAP/Native SfM (Cmd+2 → Inspector → Camera Alignment).

### Dove trovo i miei log di training?

Help → Open Training Logs (Cmd+⇧+L). Apre `~/Documents/RadianceKit/Logs/`. Ogni sessione di training scrive un proprio file JSONL con timestamp nel nome file — la prima riga è la configurazione, poi segue una riga di progresso ogni 100 iterazioni, l'ultima riga è il summary con final loss e success flag.



COLOPHON

*Composto in SF Pro · Codice in SF Mono · Typst  
0.14 · 22. June 2026*

© 2026 Bjoern Kindler · Bischofshofener Str. 9, 82008 Unterhaching, Germania

Made with ❤️ in Unterhaching