



RADIANCEKIT

# Руководство пользователя

Фотореалистичная 3D-реконструкция  
с помощью Gaussian Splatting

---

Версия 1.5.0 · macOS 26.0+ · Май 2026

BJOERN KINDLER · KINDLER-DEV.DE

# Обзор

---

Введение — Что вам следует знать .....	3
Что такое RadianceKit? .....	3
Что такое Gaussian Splatting? .....	3
Глава 1 — Строка меню .....	5
Меню File .....	6
Меню Mode .....	9
Меню Training .....	11
Меню Viewport .....	14
Меню Export .....	20
Меню Help .....	25
Примечание: Cmd-Z в меню Edit .....	29
Обзор клавиатурных сокращений .....	30
Глава 2 — Инспектор (Expert View) .....	31
Секция Look (L1–L5) .....	34
Секция Presets (I1–I11) .....	38
Секция Training Configuration (I12–I22) .....	44
Секция Enhancements (I26–I29, I42–I44) .....	52
Секция Metrics (I30–I38) .....	59
Секция Loss Chart (I39–I41) .....	65
Когда обращаться к Инспектору? .....	68
Глава 3 — Настройки .....	70
Вкладка General .....	71
Вкладка AI Helpers .....	77
Зеркальные настройки Инспектора .....	80
Когда что? .....	81
Глава 4 — Вспомогательные окна .....	82
User Guide (W1–W4) .....	83
Keyboard Shortcuts (W5–W6) .....	86
Manage Storage (W7–W12) .....	88
Pareto Dashboard (W13–W22) .....	92
Holdout Analysis (W23–W29) .....	99
BayesOpt Console (W30–W39) .....	104
Main Window: Loss Curve and Gaussian Count (I39–I41, cross-reference) .....	110
Rule-of-Thumb Box .....	111
Глава 6 — Конфигурация обучения .....	113
Iteration (T1–T2) .....	115
Learning Rates (T3–T10) .....	117


Densification — Classic (T11–T16) .....	124
Loss (T17–T20) .....	128
SH Degree Progression (T21) .....	131
Performance (T22–T25) .....	132
Diagnostics and Point Cloud Preparation (T26–T30) .....	134
Regularization (T31–T37) .....	137
Refinement (T38–T44) .....	140
Sky Dome (T45–T48) .....	144
Adam + LR Schedule (T49–T55) .....	146
Post-Processing + Apple AI (T56–T60) .....	149
MCMC Densification (T61–T73) .....	152
Mip-Splatting (Q1.5) (T74–T76) .....	158
Adaptive Densification (Q5) (T77–T79) .....	161
Curriculum (Q6) (T80–T81) .....	162
Static Presets (TP1–TP9) .....	163
Method: .....	166
Which field for what? (Cheat Sheet) .....	167
Dangerous Fields .....	168
Глава 7 — Встроенные предустановки качества .....	169
Какая предустановка когда? .....	179
Быстрое сравнение .....	180
Пользовательские предустановки .....	181
Глава 8 — Форматы экспорта .....	182
Какой формат когда? .....	196
Быстрое сравнение .....	197
Глава 9 — SfM-бэкенды .....	198
Какой бэкенд когда? .....	205
Быстрое сравнение .....	206
Глава 10 — Режим новичка .....	207
Z1 — Импорт (выбор изображений и предустановки) .....	208
Z2 — Обработка (SfM + Training) .....	217
Z3 — Предпросмотр (повернуть 3D-модель) .....	225
Z4 — Экспорт (выбор формата и сохранение) .....	228
Переключение в Expert Mode .....	233
Часто задаваемые вопросы .....	234


## Как читать это руководство


---


Каждый раздел руководства построен по единой схеме. Слева находятся пути управления и технические детали, а справа, в тёплой боковой колонке, всегда даётся простое объяснение. Маленькие значки в начале строки одним взглядом подсказывают, какого рода информация последует далее.

### ЧЕТЫРЕ ЗНАЧКА

 **Где я это найду?** Конкретный путь по щелчкам в приложении — строка меню, секция Инспектора или шаг режима новичка. Здесь же приводятся соответствующие клавиатурные сокращения. Значок представляет собой булавку на карте и показывает: вот в каком месте пользовательского интерфейса находится данная функция.

 **Подробности.** Значения по умолчанию, диапазоны значений и пути к коду. Это встречается прежде всего в настройках обучения, которые не являются пунктом меню, а представляют собой числовые параметры. Значок показывает маленькую карточку спецификации.

 **Технически.** Что функция делает внутри, какие параметры на неё влияют, на что она реагирует и какие у неё побочные эффекты. Для читателей, желающих понять, что происходит за кулисами. Значок представляет собой блок ползунков и символически отображает регулировочные ручки под капотом.

 **Простыми словами.** Главная мысль, выраженная ясными словами — без жаргона, без кода. Прочтите этот раздел сначала, если вам нужно лишь быстро узнать, для чего предназначена данная функция и когда она вам понадобится. Значок — это речевой пузырь, он означает «вкратце». Эта колонка всегда залита тёплым песочным оттенком, чтобы глаз сразу её находил.

### ЦВЕТА ГЛАВ

У каждой главы свой акцентный цвет, который вы узнаете по идентификатору (например **M1**) слева рядом с заголовком каждой записи и по маленьким значкам перед ним. При перелистывании вы сразу видите, в какой главе сейчас находитесь.

- 1 Меню
- 2 Инспектор
- 3 Настройки
- 4 Вспомогательные окна
- 6 Тренировка
- 7 Предустановки
- 8 Экспорты
- 9 SfM
- 10 Режим новичка

#### СОВЕТЫ ПО НАВИГАЦИИ

**Быстрый старт.** Если вас интересует только управление, переходите сразу к **главе 10 — Режим новичка**. Это пошаговый вариант из четырёх шагов, не требующий никаких предварительных знаний.

**Более глубокое погружение.** **Глава 2 — Инспектор** и **глава 7 — Предустановки** объясняют элементы управления и предустановленные профили качества, которые доступны в Режиме эксперта.

**Справочник.** Содержание и полнотекстовый поиск по PDF помогут найти нужную функцию. Не обязательно читать руководство с начала до конца.

# Введение — Что вам следует знать

---

## Что такое RadianceKit?

RadianceKit — это нативное приложение для macOS, которое из серии обычных фотографий или видео создаёт проходимую 3D-реконструкцию. На вход подаются, например, от 50 до 500 снимков, сделанных вокруг объекта, по комнате или над пейзажем. На выходе — так называемая сцена Gaussian Splatting — 3D-модель, которую можно в реальном времени просматривать на Mac с любого ракурса, экспортировать и встраивать на веб-сайты, и которая в основных аспектах выглядит фотореалистично.

Приложение полностью работает локально на вашем Mac — никакие изображения не загружаются в облако, не требуется ни вход в учётную запись, ни подписка. Оно интенсивно использует GPU вашего Mac на Apple Silicon (серия M): полный цикл обучения может занимать от двух минут до нескольких часов в зависимости от сцены и предустановки. Во время вычислений вы можете спокойно продолжать работу на Mac, RadianceKit выполняется в фоновом режиме и сообщит, когда результат будет готов.

Существует два режима работы: *Режим новичка* (Simple Mode) проводит вас через рабочий процесс Импорт → Выбор предустановки → Тренировка → Экспорт в четыре шага. *Режим эксперта* (Expert Mode) открывает большой Инспектор со всеми регулировками, окном живого предпросмотра и диагностическими графиками. Между режимами можно переключаться в любой момент; данные сцены при этом сохраняются.

## Что такое Gaussian Splatting?

Gaussian Splatting (часто сокращённо 3DGS или просто *Splatting*) — это сравнительно новый метод фотореалистичного 3D-отображения, представленный в 2023 году в статье, подготовленной в Граце и INRIA. Идея состоит в следующем: вместо того чтобы моделировать сцену как классическую полигональную сетку (треугольники) или как воксельную решётку, её собирают из миллионов маленьких, мягких 3D-облачков — каждое отдельное облачко представляет собой 3D-распределение Гаусса (отсюда и название) со своей позицией, размером, формой,

цветом и прозрачностью. Эти облачка обучаются так, чтобы со всех ракурсов ваших входных фотографий вместе давать правильное изображение.

На практике это означает, что Gaussian Splatting способен изображать отражения, блики, мягкую листву, волосы или занавески так, как классическое 3D-моделирование не может или может, но ценой колоссальных усилий. С другой стороны, результат не является редактируемой 3D-моделью в классическом понимании — вы не можете просто сдвинуть одну стену или переставить вазу. Это скорее *замороженный снимок* пространства, по которому вы можете свободно перемещаться. Для многих применений — архитектурная визуализация, презентация продукта, виртуальные туры, криминалистика, культурное наследие — именно в этом и состоит сильная сторона метода.

Чтобы из входных изображений получилась 3D-сцена, нужно два шага. Сначала приложение с помощью метода под названием *Structure-from-Motion (SfM)* вычисляет, где находилась ваша камера при каждой фотографии. При этом попутно получается грубое облако точек сцены. Затем стартует собственно обучение Gaussian Splatting: исходя из этого грубого облака миллионы 3D-облачков поэтапно распределяются, увеличиваются, уточняются и корректируются по положению и цвету, пока со всех входных ракурсов они не дадут подходящее изображение.

Вам не обязательно что-либо знать обо всём этом, чтобы пользоваться RadianceKit. Режим новичка полностью скрывает эти шаги. Но если вы хотите понять, что означают диагностические числа в режиме эксперта (итерация, loss, gaussians, SSIM ...) или почему одни сцены получаются красивее других, то ответы вы найдёте в последующих главах руководства.

## ГЛАВА

# Глава 1 — Строка меню

---

Строка меню RadianceKit организует все функции, которые не находятся непосредственно в главном окне или Инспекторе. Это преимущественно действия, затрагивающие всю сцену (Open, Save, New Project), управляющие обучением (Start, Pause, Continue), работающие с viewport (авто-вращение, скриншот, цвет фона) и запускающие экспорт в различные 3D- и медиа-форматы. Кроме того, здесь есть точки перехода ко всем вспомогательным окнам (User Guide, Pareto Dashboard, Holdout Analysis, BayesOpt Console).

Клавиатурные сокращения перечислены справа от каждого пункта меню. Соглашения:  $\mathbb{C}$  — клавиша Command (клавиша Apple),  $\mathbb{S}$  — Shift,  $\mathbb{O}$  — Option (Alt), а  $\mathbb{C}$  — Control. Пример:  $\mathbb{S}\mathbb{C}\mathbb{T}$  означает Shift+Command+T. Все задокументированные здесь сокращения дополнительно перечислены в отдельном обзорном окне по Help → Keyboard Shortcuts ( $\mathbb{C}/$ ).

Следующие 42 пункта документированы в порядке инвентаря (M1–M42), сгруппированы по родительскому меню, к которому они принадлежат. Все пункты были сверены с текущим состоянием кода в (строки 175–477). Пункты не удалялись и не заменялись относительно инвентаря; новый пункт меню Edit (Cmd-Z для «Remove Image») предоставляется системным фреймворком NSUndoManager и потому не появляется в коде RadianceKitApp (см. примечание в конце главы).

## Меню File

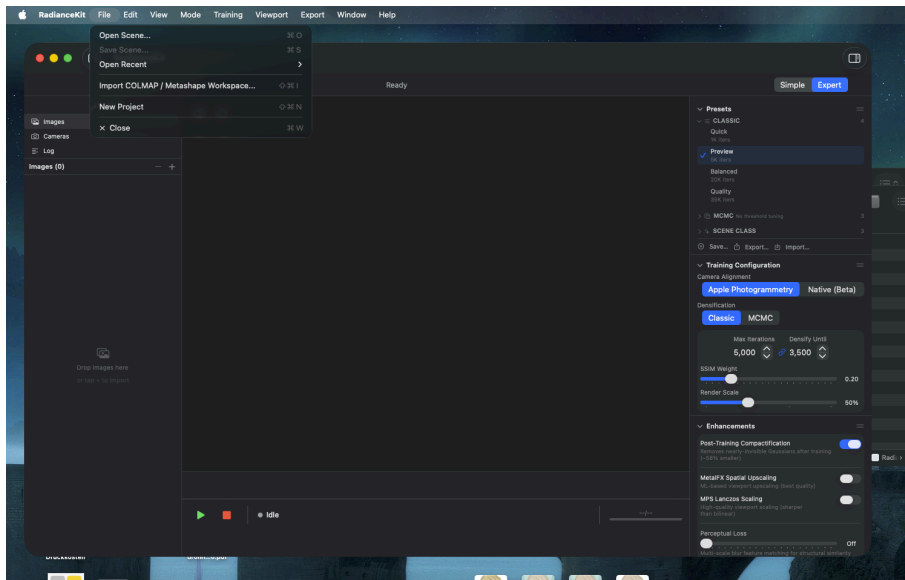


Рисунок 1. Меню File развёрнуто — пункты с M1 по M6

Меню File заменяет стандартный пункт «New Window» от Apple на действия, специфичные для проекта. Оно охватывает загрузку/сохранение сцен, динамический список Recent, импорт рабочей области и жёсткий сброс в пустое состояние.

### M1 File > Open Scene...



Строка меню → File → Open Scene... (⌘O).

### ТЕХНИЧЕСКИ

Открывает файловый диалог для форматов: бандл `RadianceScene`, `.ply`, `.splat` и `.spz`. Одиночный выбор, может показывать и файлы, и директории (для бандла). После успешного выбора путь добавляется в список Recent, и сцена загружается асинхронно — предыдущая заменяется, а конвейер обучения инициализируется загруженным состоянием. Файлы PLY/SPZ/Splat читаются через соответствующие загрузчики формата; бандл `.radianceScene` — это директория с манифестом, снимком облака и результатами SfM.

### ПРОСТЫМИ СЛОВАМИ

Так вы загружаете уже обученную сцену обратно в приложение. Работает с собственным форматом RadianceKit, а также со стандартными форматами PLY, SPLAT и SPZ, которые производят другие splatting-программы. Используйте это, напр., когда обучили сцену на ночь и хотите на следующий день продолжить или экспортировать. При открытии текущего состояния в главном окне заменяется — так что сначала сохраните, если текущая сцена ещё вам важна. Путь автоматически попадает в «Open Recent» (M3), чтобы в следующий раз можно было быстрее к нему вернуться.

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

ГДЕ

Строка меню → File → Save Scene... (⌘S).

**ТЕХНИЧЕСКИ**

Открывает файловый диалог сохранения с типом контента бандл `RadianceScene` и предустановленным именем `scene.radiancecene`. Записывает пакет-директорию с `manifest.json`, сериализованным облаком `Gaussian` (PLY-snapshot) и дампом результата SfM, чтобы продолжение обучения работало и после повторного открытия. Пункт деактивирован, пока не существует ни одного `Gaussian`. Сохранение идёт не в путь `training-logs`, а туда, куда указывает `save`-диалог — обычно под `~/Documents/`.

**ПРОСТЫМИ СЛОВАМИ**

Сохраняет вашу текущую сцену как файл (точнее: как `package`-папку, которая выглядит как файл). Только после этого вы сможете позднее снова открыть сцену через «Open Scene...» (M1). Пакет содержит как облако `Gaussian`, так и результат SfM, поэтому позднее можно дописать продолжение обучения (M12–M14). Пока вы не закончили прогон обучения, пункт серый. Имя по умолчанию — `scene.radiancecene` — но в `Save`-диалоге можно дать своё имя.

**M3** File > Open Recent > [имена сцен]

ГДЕ

Строка меню → File → Open Recent → (список).

**ТЕХНИЧЕСКИ**

Динамическое подменю, сгенерированное из списка недавно открытых путей (хранятся в настройках). Каждый пункт помечен именем файла и загружается по щелчку. Если список пуст, вместо него появляется деактивированная метка «No Recent Scenes». В стиле Apple список держит N последних открытых сцен — ограничение применяется при записи в настройки, а не в самом построителе меню.

**ПРОСТЫМИ СЛОВАМИ**

Здесь видны недавно открытые сцены, и можно вернуться к ним щелчком, не проходя через файловый диалог. Если вы только что начали, список пуст и серый в меню. Каждая открытая через «Open Scene...» (M1) сцена автоматически попадает в этот список. Если список становится слишком полным или вы хотите очистить его по соображениям приватности, используйте «Clear Recent» (M4).

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

ГДЕ

Строка меню → File → Open Recent → Clear Recent.



ТЕХНИЧЕСКИ

Очищает список Recent в настройках. Срабатывает немедленно, без диалога подтверждения. Пункт появляется в подменю только тогда, когда в списке Recent фактически есть записи (находится ниже разделителя после путей).

**ПРОСТЫМИ СЛОВАМИ**

Очищает список недавно открытых сцен. Удобно, если вы поигрались с тестовым набором данных и больше не хотите видеть пути. Сами файлы сцен не удаляются — удаляется только ссылка в меню. Действие срабатывает немедленно, без подтверждения; затем подменю показывает «No Recent Scenes». Пункт появляется только если в списке действительно есть сцены — когда список пуст, он невидим.

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

ГДЕ

Строка меню → File → Import COLMAP / Metashape Workspace... (⇧⌘I).



ТЕХНИЧЕСКИ

Открывает folder picker. Ожидает папку с раскладкой COLMAP-workspace (напр., `sparse/0/cameras.{bin,txt}` плюс `images/`). После выбора запускается предварительная проверка рабочей области — она определяет три варианта раскладки (`sparse/0/`, `sparse/`, корень) и является ли реконструкция двоичной (`cameras.bin`) или в текстовой форме ETH3D (`cameras.txt`). При успехе рабочая область импортируется; иначе в журнал приложения попадает только предупреждение. См. также главу 9 «SfM-бэкенды», Q6 для полной логики конвейера.

**ПРОСТЫМИ СЛОВАМИ**

Если вы используете Metashape, COLMAP, RealityCapture или подобное ПО для реконструкции камер и у вас есть экспорт, загрузите папку здесь. RadianceKit тогда пропускает этап SfM и сразу начинает обучение — что экономит часы на больших сценах. Перетаскивание на главное окно работает аналогично. Ожидается папка с раскладкой COLMAP (т.е. `sparse/0/` с `cameras.*` плюс папка `images/`). Подробнее о поддерживаемых раскладках и рабочих процессах — в главе 9 «SfM-бэкенды».

## M6 File > New Project



Строка меню → File → New Project (⇧⌘N).

### ТЕХНИЧЕСКИ

Проверяет, есть ли несохранённая работа. Если да, перед тем как что-то потерять, появляется диалог подтверждения. Если сохранять нечего, сброс выполняется немедленно — очищает импортированные изображения, результат SfM, облако Gaussian, состояние обучения и все зависимые индикаторы UI. Примечание: созданная пользователем библиотека предустановок сохраняется, поскольку живёт в настройках приложения, а не в состоянии проекта.

### ПРОСТЫМИ СЛОВАМИ

Сбрасывает всё к пустому старту — как если бы вы только что свежо открыли приложение. Если у вас есть несохранённая работа, приложение сначала спросит. Используйте это, когда хотите начать с нуля с совершенно другой сценой. Импортированные изображения, результат SfM, облако Gaussian и состояние обучения полностью очищаются. Ваши собственные предустановки, однако, сохраняются, потому что живут в настройках приложения и не относятся к сцене.

## Меню Mode

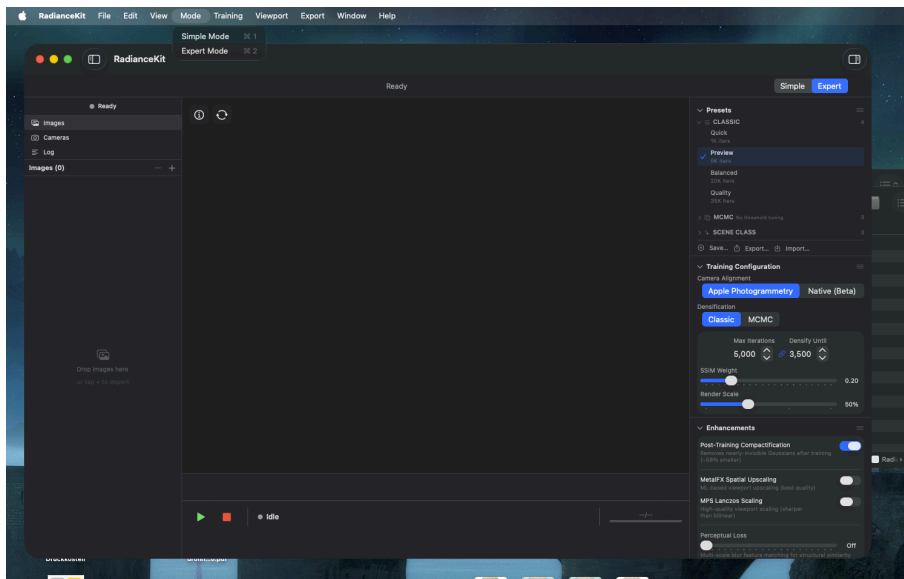


Рисунок 2. Меню Mode с переключателями Simple и Expert Mode

Два простых переключателя между направляемым Simple Mode (wizard-like, 4 шага) и полным Expert Mode (классическая раскладка Инспектора со всеми элементами управления).

**M7 Mode > Simple Mode**

Строка меню → Mode → Simple Mode (⌘1).



Переключает состояние приложения в Simple Mode. Главная область приложения затем показывает направляемый рабочий процесс вместо раскладки Expert. Состояние режима хранится в настройках (см. S1 «Default Mode» в главе 3 «Настройки»).

**ПРОСТЫМИ СЛОВАМИ**

Переключает на пошаговый вариант, где приложение ведёт вас через импорт, обработку, предпросмотр и экспорт. Рекомендуется, если вы только начинаете или нужен быстрый результат. Большинство детальных элементов управления скрыты — вы работаете с разумными значениями по умолчанию. Если позднее захочется копать глубже, просто переключитесь в Expert Mode (M8). Какой режим активен при запуске приложения, можно задать в настройках (глава 3, S1).

**M8 Mode > Expert Mode**

Строка меню → Mode → Expert Mode (⌘2).



Переключает состояние приложения в Expert Mode. Это выводит полную раскладку Инспектора со всеми секциями (Presets, TrainingConfig, Enhancements, Metrics, LossChart, ProjectNavigator). В Expert Mode доступны все параметры обучения, COLMAP-picker, mid-compact-переключатели и диагностика. Живой предпросмотр тоже работает только в этом режиме.

**ПРОСТЫМИ СЛОВАМИ**

Переключает на полный вид со всеми элементами управления. Здесь видны loss-графики в реальном времени, можно тонко настраивать все параметры и параллельно управлять несколькими сравнительными конфигурациями через предустановки. Рекомендуется, если хочется понять, что обучение делает внутри, или если хочется целенаправленно экспериментировать. Живой предпросмотр, COLMAP-picker и диагностика тоже доступны только здесь. Если ощущаете перегруз, вернитесь к Simple Mode через M7 — ваша сцена при этом сохраняется.

## Меню Training

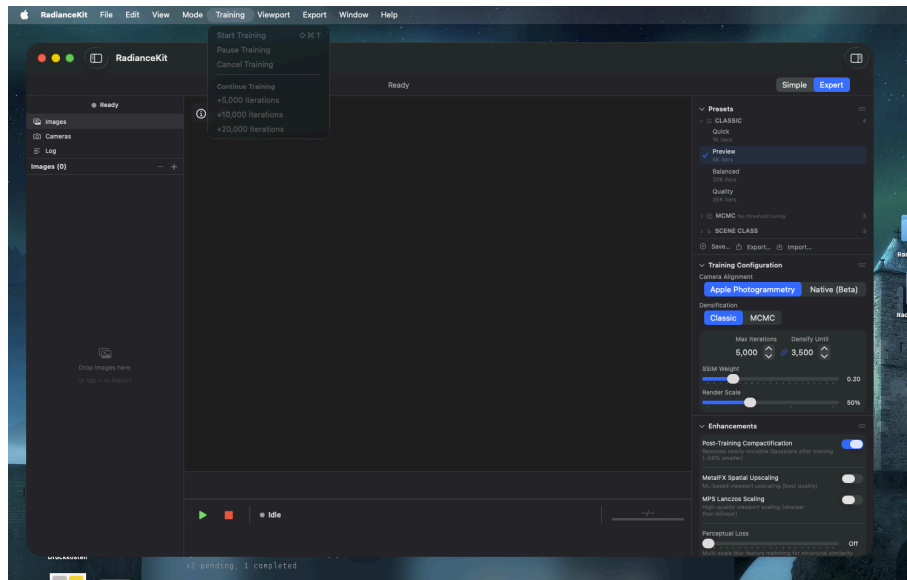


Рисунок 3. Меню Training с подменю Continue — пункты с M9 по M14

Четыре действия вокруг прогона обучения: старт, пауза, отмена, продление на фиксированное количество итераций. Все три пункта Continue ограничены IAP (не кликабельны в free trial-версии).

### M9 Training > Start Training



ГДЕ

Строка меню → Training → Start Training (⇧⌘T).



ТЕХНИЧЕСКИ

Запускает конвейер обучения асинхронно. Предусловие: существует результат SfM, и в настоящее время не запущен никакой другой конвейер. Оба условия блокируют пункт, если не выполнены. При запуске считываются текущие значения конфигурации, создаётся новый JSONL-лог под `~/Documents/RadianceKit/Logs/training_YYYY-MM-DD_HHmmss.jsonl`, и в зависимости от выбора стратегии берётся классический или MCMC-путь. Состояние обучения переключается с «idle» на «training».

#### ПРОСТЫМИ СЛОВАМИ

Большая зелёная кнопка — после того как вы импортировали фотографии и сделана реконструкция камер, запускается собственно обучение Gaussian Splatting. Дайте приложению поработать; в зависимости от предустановки от 1 минуты (Quick) до нескольких часов (MCMC Quality). Пункт остаётся серым, пока нет результата SfM или пока запущен другой конвейер. Каждый прогон также пишет лог в `~/Documents/RadianceKit/Logs/`, который можно позднее проанализировать через Pareto Dashboard (M40).


**M10 Training > Pause Training**

ГДЕ

Строка меню → Training → Pause Training.

 **ТЕХНИЧЕСКИ**

Приостанавливает запущенное обучение. Активно только тогда, когда состояние обучения — «training». Пауза останавливает цикл итераций на следующей безопасной точке синхронизации, сохраняет полное состояние GPU (буферы Gaussian, моменты оптимизатора, позицию scheduler) и переключается в «paused». Возобновление — повторным нажатием (заголовок пункта статичен, но приложение переключает между Pause/Resume во внутренней логике). Поставленные на паузу тренировки не переживают выход из приложения; в этом случае вместо этого сохраните сцену и позднее продлите её через Continue Training (M12–M14).

 **ПРОСТЫМИ СЛОВАМИ**

Кратко останавливает обучение без потери прогресса. Удобно, когда компьютер на короткое время нужен для чего-то более важного. Щелкните снова — продолжит. Не работает между перезапусками приложения — если действительно хотите продолжить позже, завершите обучение через Cancel (M11), сохраните сцену через Save Scene (M2), а затем используйте Continue Training (M12–M14). Во время паузы GPU полностью простаивает; однако память остаётся выделенной.

**M11 Training > Cancel Training**

ГДЕ

Строка меню → Training → Cancel Training.

 **ТЕХНИЧЕСКИ**

Отменяет запущенное обучение. Активно, когда состояние обучения не «idle». Устанавливает cancel-флаг в training-движке, который чисто завершает цикл итераций на следующей точке синхронизации, пишет итоговую summary-запись в JSONL-лог и сбрасывает состояние в «idle». Обученное к этому моменту облако сохраняется (можно сохранить или экспортировать), но помечается как «cancelled».

 **ПРОСТЫМИ СЛОВАМИ**


Окончательно отменяет запущенное обучение. Текущее состояние сохраняется — так что если после нескольких тысяч итераций у вас уже есть презентабельный результат, его всё ещё можно экспортировать. Если хотите только кратко прервать, используйте вместо этого Pause (M10). В журнале обучения прогон помечается как «cancelled», а итоговое значение loss всё равно записывается. Отменённую сцену можно позднее также продолжить через Continue Training (M12–M14), пока приложение тем временем не было закрыто.

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

Строка меню → Training → Continue Training → +5,000 iterations.



Продолжает обучение ещё на 5 000 итераций. Активно, если завершённый прогон обучения возобновляем и полная версия разблокирована. Возможность возобновления держится, если существует завершённый прогон обучения и полное состояние оптимизатора всё ещё в памяти. При продолжении переносятся моменты Adam и LR-scheduler, так что продолжение ведёт себя как непрерывный прогон 25K/45K/60K, а не как рестарт. JSONL-лог получает новую config-запись с инкрементной настройкой. Доступно только в полной версии.

 **ПРОСТЫМИ СЛОВАМИ**


Добавляет ещё 5 000 шагов обучения. Используйте это, если результат после первого прогона близок, но ещё не достаточно резкий. Работает только в платной полной версии. В отличие от полностью нового прогона, состояние оптимизатора сохраняется, так что продолжение ощущается как непрерывный прогон. Если нужно больше 5 000 шагов, сразу переходите к M13 (+10 000) или M14 (+20 000).

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

Строка меню → Training → Continue Training → +10,000 iterations.



Идентично M12, но с 10 000 дополнительных итераций. Те же предусловия, тот же путь LR-scheduler. Рекомендуется, когда первоначальное обучение прогонялось с предустановкой среднего уровня, и хочется увидеть значительное улучшение качества, не перезапуская прогон с нуля.

 **ПРОСТЫМИ СЛОВАМИ**

Продлевает обучение на 10 000 шагов — среднее из трёх доступных значений Continue. Хороший выбор, когда первый прогон был нормальным, но хочется получить заметно лучше. Как у M12 и M14, расписание learning-rate продолжается бесшовно вместо рестарта. Доступно только в полной версии.

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



ГДЕ

Строка меню → Training → Continue Training → +20,000 iterations.



ТЕХНИЧЕСКИ

Идентично M12 / M13, но с 20 000 дополнительных итераций. Самый крупный preset-continue-скачок. Для МСМС-обучения это часто то, что делает разницу между «нормально» и «benchmark-grade»; для Classic за пределами 35–40К опыт показывает, что мало что добавляется.

### ПРОСТЫМИ СЛОВАМИ

Добавляет 20 000 дополнительных шагов обучения, максимальное значение Continue. Используйте это, когда действительно хотите выжать последний процент качества. Для классического обучения за пределами 40 000 шагов это часто не приносит много большего — тогда как для МСМС это часто оправдано, потому что сходимость там устанавливается медленнее. Ожидайте значительного дополнительного времени выполнения в зависимости от сцены. Как и M12 и M13, этот пункт также доступен только в полной версии.

## Меню Viewport

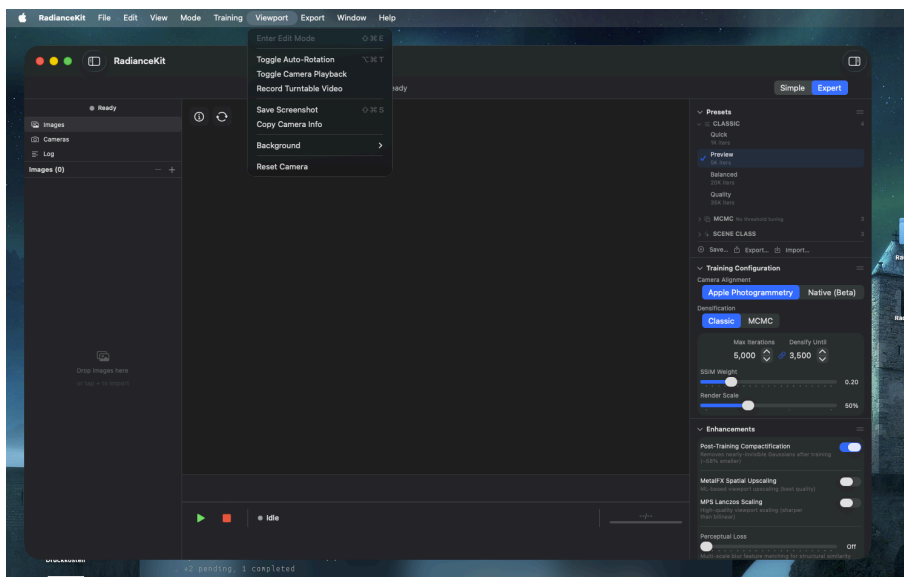


Рисунок 4. Меню Viewport с Edit Mode, элементами управления камерой и подменю Background

Управляет 3D-просмотром: Edit Mode для выбора и очистки Gaussian, элементы управления камерой (авто-вращение, проигрывание, запись), скриншот, цвет фона и сброс.

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

ГДЕ

Строка меню → Viewport → Enter Edit Mode (или «Exit Edit Mode», в зависимости от состояния).  
⌘E.

**ТЕХНИЧЕСКИ**

Заголовок пункта динамический и показывает «Exit Edit Mode» или «Enter Edit Mode» в зависимости от состояния. При нажатии Edit Mode переключается на рендере viewport. При выходе из Edit Mode текущий выбор также сбрасывается. Edit Mode активирует выбор Gaussians щелчком, выбор рамкой и удаление помеченных Gaussians (см. редакторскую область UI). Деактивирован, пока к viewport не подключён рендер.

**ПРОСТЫМИ СЛОВАМИ**

Переключает между обычным 3D-просмотром и режимом редактирования, в котором можно помечать и удалять отдельные Gaussians (напр., floaters или выбросы на фоне). При выходе выбор автоматически сбрасывается. Пункт остаётся серым, пока в viewport не видно сцены. Метка меняется между «Enter Edit Mode» и «Exit Edit Mode» в зависимости от состояния — так всегда понятно, в каком режиме вы сейчас находитесь.

**M16 Viewport > Toggle Auto-Rotation**

ГДЕ

Строка меню → Viewport → Toggle Auto-Rotation  
(⌘⌘T).

**ТЕХНИЧЕСКИ**

Включает или выключает непрерывное вращение камеры viewport вокруг вертикальной оси через центр сцены. Ось и скорость берутся из конфигурации управления камерой. Авто-вращение — чисто viewport-эффект и не влияет ни на обучение, ни на запись — если параллельно используется turntable- видеорекордер (M18), авто-вращение даёт точно ту траекторию, которую захватывает рекордер.

**ПРОСТЫМИ СЛОВАМИ**

Непрерывно медленно вращает камеру вокруг вашей сцены, чтобы можно было видеть её со всех сторон, не таская мышью. Щёлкните снова, чтобы остановить вращение. Удобно при просмотре завершённо обученных сцен или как фоновая анимация для живой демонстрации. Если параллельно записываете видео (M18), авто-вращение даёт ровно то движение, которое захватывает рекордер.

**M17 Viewport > Toggle Camera Playback**

Строка меню → Viewport → Toggle Camera Playback.



Переключает воспроизведение пути камеры. Если существует записанный путь камеры (напр., из предыдущей записи или потому что был загружен `transforms.json`), путь воспроизводится — камера viewport больше не движется в ответ на ввод мыши/трекпада, а воспроизводит траекторию покадрово. Повторное нажатие приостанавливает воспроизведение.

**ПРОСТЫМИ СЛОВАМИ**

Воспроизводит ранее записанный или импортированный путь камеры. Так можно повторить оригинальный путь, который использовался для съёмки сцены, или проверить запланированное орбитальное движение перед экспортом видео. Пока идёт воспроизведение, ввод мыши и трекпада деактивирован — камера строго следует пути. Повторный щелчок ставит воспроизведение на паузу. Если вы не загружали и не записывали путь камеры, ничего не происходит.

**M18 Viewport > Record Turntable Video**

Строка меню → Viewport → Record Turntable Video.



Переключает запись viewport. При первом нажатии запускается запись кадров во временный путь; при втором нажатии запись завершается, кодируется и пишется в MP4-путь (путь запрашивается через save-диалог). В отличие от Export → Media → Orbit Video (M31), который выдаёт фиксированную траекторию 360° с настраиваемой длительностью, turntable-рекордер захватывает *вживую* то, что вы видите в viewport — так что можно также записать ручное движение камеры.

**ПРОСТЫМИ СЛОВАМИ**

Записывает видео напрямую в viewport. Вращается ли камера автоматически или вы сами двигаете её мышью — всё, что вы видите, сохраняется в MP4-файл. В отличие от экспорта «Orbit Video» (M31), движение камеры вы контролируете сами. Первый щелчок запускает запись, второй щелчок завершает её и спрашивает, куда сохранить. Удобно, когда, напр., нужно показать определённый детальный рап, который не был бы возможен при жёсткой орбитальной траектории.

**M19 Viewport > Save Screenshot**

Строка меню → Viewport → Save Screenshot (⇧⌘S).



Захватывает один кадр viewport на полном разрешении рендера (т.е. не пиксельную раскладку окна, а полное содержимое render-target) как PNG-файл. Путь запрашивается через save-диалог. Цвет фона (M21–M23) запекается. Настройки MetalFX/MPS upscaling из Enhancements (см. I27/I28) вступают в силу, если активны — скриншот тогда показывает увеличенный вывод.

**ПРОСТЫМИ СЛОВАМИ**

Сохраняет снимок вашего текущего 3D-вида как PNG-изображение. Удобно для маркетинговых материалов или быстрого сравнения. Примечание — фон — часть изображения — если нужна прозрачность, лучше экспортируйте файл сцены. Разрешение соответствует внутреннему render-target, не размеру вашего окна — поэтому изображение часто меньше, чем выглядит в окне. Также применяются любые настройки upscaling (Inspector → Enhancements).

**M20 Viewport > Copy Camera Info**

Строка меню → Viewport → Copy Camera Info.



Считывает текущую позу камеры viewport (позиция, look-at-точка, up-вектор) и значения FOV из управления камерой и записывает их как многострочный текст в буфер обмена. Формат человеко-читаемый (метка = значение на строку), не JSON. Удобно для воспроизведения конкретного вида в отладочных целях или передачи в support.

**ПРОСТЫМИ СЛОВАМИ**

Копирует текущую позицию камеры и направление взгляда как текст в буфер обмена. Если хочется, напр., показать коллеге-разработчику, где в сцене какое-то место выглядит странно, просто вставьте текст в письмо или окно чата. Формат человеко-читаемый (одна строка на значение), не JSON. В основном предназначено для bug-репортов или запросов в поддержку.

**M21 Viewport > Background > Dark Gray**

Строка меню → Viewport → Background → Dark Gray.



Устанавливает цвет фона viewport на тёмно-серый (RGB 0.1/0.1/0.1). Рендерер использует этот цвет как фон, против которого композитятся Gaussians. Цвет по умолчанию при старте приложения управляется опцией настроек S3 «Default Viewport Background».

**ПРОСТЫМИ СЛОВАМИ**

Окрашивает фон 3D viewport тёмно-серым. Стандартный выбор для большинства сцен — предлагает хороший контраст и для светлых, и для тёмных Gaussians, не приковывая глаз к чисто чёрной или белой области. Цвет также запекается в скриншоты (M19) и orbit-видео (M31). Если Dark Gray ощущается слишком неэффектно, попробуйте для сравнения также Black (M22) или White (M23). Какой цвет активен при старте приложения, задаётся в настройках (S3).

**M22 Viewport > Background > Black**

Строка меню → Viewport → Background → Black.



Устанавливает цвет фона viewport на чистый чёрный (RGB 0/0/0). Помогает, когда сцена имеет много светлых floaters и хочется их выявить, или для маркетинговых материалов с тёмным look-and-feel.

**ПРОСТЫМИ СЛОВАМИ**

Чёрный фон. Хорош для очень светлых сцен или когда хочется заглянуть в Edit Mode и искать маленькие светлые Gaussians (floaters), которые теряются на сером. Также идеален для маркетинговых материалов с тёмным, драматическим видом. Цвет запекается в скриншоты и orbit-видео — если нужна прозрачность для последующего композита, чёрный — худший выбор. Для тёмных floaters переключитесь в обратную сторону на White (M23).

**M23 Viewport > Background > White**

Строка меню → Viewport → Background → White.



Устанавливает цвет фона viewport на чистый белый (RGB 1/1/1). Полезен, когда сцена имеет преимущественно тёмный контент и хочется видеть тёмные floaters (типичные уличные фоновые шумы).

**ПРОСТЫМИ СЛОВАМИ**

Белый фон. Удобен, когда субъект лучше виден light- on-dark, или для нахождения тёмных выбросов, которые потом можно удалить в Edit Mode (M15). Для уличных сцен белый часто полезнее чёрного, потому что типичные уличные floaters обычно тёмные. Как и с другими фоновыми опциями, цвет запекается в скриншоты и видео.

**M24 Viewport > Reset Camera**

Строка меню → Viewport → Reset Camera.



Сбрасывает камеру viewport, покидает вид тренировочной камеры и останавливает авто-вращение. Камера возвращается в начальную позицию (обычно: перед сценой, чуть сверху), авто-вращение выключено, и если рендерер показывал тренировочную камеру (одну из поз SfM), возвращается к свободной камере.

**ПРОСТЫМИ СЛОВАМИ**

Возвращает камеру viewport в начальную позицию. Если вы потерялись при движении или вытолкнули сцену из кадра — щёлкните здесь один раз, и снова видно то, что вы должны видеть. Одновременно выключает авто-вращение, если оно сейчас работает, и возвращает от замороженной тренировочной камеры к свободному виду. Так в любом случае получается чистый рестарт обзора.

## Меню Export

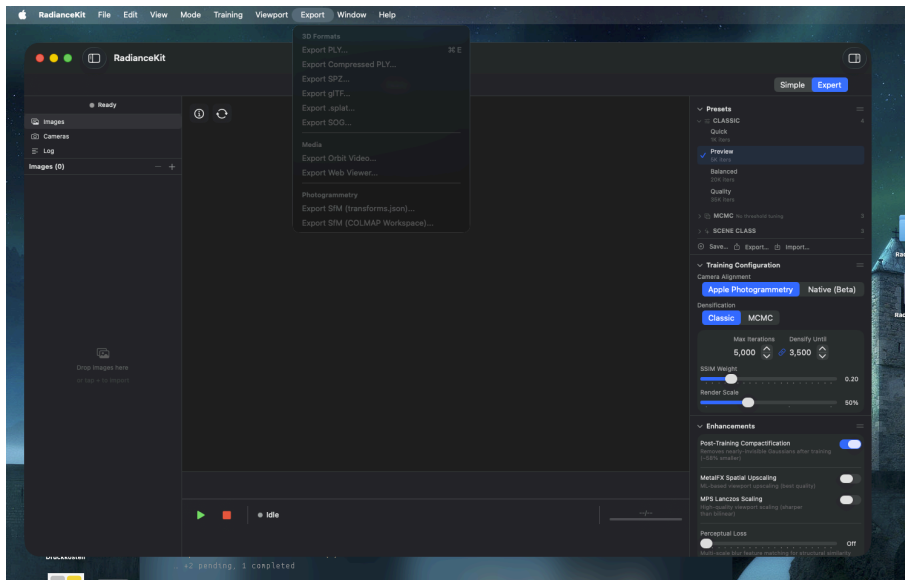


Рисунок 5. Меню Export с тремя группами подменю — 3D Formats, Media и Photogrammetry

Восемь целей экспорта плюс два photogrammetry-экспорта, сгруппированные в три секции (3D Formats, Media, Photogrammetry). Первые шесть строятся через общую вспомогательную процедуру, каждая из которых открывает save-диалог и регистрирует экспорт в каталоге форматов. Photogrammetry-пункты имеют индивидуальную логику. Все photogrammetry-экспорты и некоторые 3D-экспорты доступны только в полной версии.

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



ГДЕ

Строка меню → Export → 3D Formats → PLY (⌘E).



ТЕХНИЧЕСКИ

Открывает save-диалог с именем по умолчанию `gaussians.ply`. По ОК текущее облако Gaussian пишется в стандартизированный ASCII/binary PLY-формат — совместим с SuperSplat, PolyCam, PlayCanvas и всеми распространёнными 3DGS-просмотрщиками. Полные SH-коэффициенты, полная точность (Float32 на поле). Размер файла часто несколько сотен МБ при  $\geq 500K$  Gaussians.

### ПРОСТЫМИ СЛОВАМИ

Сохраняет вашу 3D-сцену как стандартный PLY-файл. Это самый универсальный формат — почти любое ПО умеет его загружать, от SuperSplat до PolyCam и PlayCanvas. Файлы, однако, становятся большими, часто несколько сотен мегабайт. Используйте PLY, когда хотите продолжать работу в полном качестве или архивировать. Если хотите поделиться сценой в сети, посмотрите вместо этого на SPZ (M27) или Compressed PLY (M26) — они существенно меньше.

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

ГДЕ

Строка меню → Export → 3D Formats → Compressed PLY.

**ТЕХНИЧЕСКИ**

Пишет облако Gaussian в формат compressed-PLY с собственным квантованием полей позиции, масштаба, поворота и SH. Файлы в 5–10 раз меньше несжатого PLY (M25) с минимальной визуальной потерей. Совместим с SuperSplat (который читает стандарт compressed-PLY) и PlayCanvas. Имя по умолчанию `gaussians_compressed.ply`.

**ПРОСТЫМИ СЛОВАМИ**

Как обычный PLY, но в 5–10 раз меньше. Качество остаётся почти таким же. Используйте это, когда хотите поделиться файлом онлайн или отправить по почте. Работает напрямую с SuperSplat и PlayCanvas. Если целевая система требует ещё меньших файлов (мобильные, браузерные демо), используйте вместо этого SPZ (M27) — он сжат ещё более агрессивно. Для полного качества редактирования используйте несжатый PLY (M25).

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

ГДЕ

Строка меню → Export → 3D Formats → SPZ.

**ТЕХНИЧЕСКИ**

Пишет облако Gaussian в SPZ-формат — сжатый splat-формат, опубликованный Niantic, с агрессивным квантованием (~90 % меньше, чем несжатый PLY). В основном оптимизирован для веб-просмотрщиков и мобильных приложений. Совместим с Niantic Splatt3R, `gsplat.js` и браузерным просмотрщиком Niantic.

**ПРОСТЫМИ СЛОВАМИ**

Один из самых маленьких форматов. Примерно в 10x меньше обычного PLY. Используйте это в основном, когда хотите показать сцену в браузере или просматривать в приложении телефона. Для максимального качества PLY — лучший выбор. SPZ был разработан Niantic и работает напрямую с `gsplat.js`, Splatt3R и веб-просмотрщиком Niantic. Из-за сильного сжатия SPZ-файлы больше нельзя легко продолжать обучать — для редактирования используйте PLY.

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

ГДЕ

Строка меню → Export → 3D Formats → glTF.



ТЕХНИЧЕСКИ

Пишет `.glb`-файл (binary glTF) с расширением `KHR_gaussian_splatting`. Стандарт-конформен, подходит для конвейеров, использующих glTF-движки вроде Babylon.js или Three.js и реализующих расширение `KHR_gaussian_splatting`.



## ПРОСТЫМИ СЛОВАМИ

Сохраняет сцену в формате glTF, который понимают многие 3D-программы и веб-движки — при условии, что они поддерживают расширение Gaussian-Splatting. Если у вас есть специфический 3D-конвейер (напр., Three.js или Babylon.js), который его понимает, это ваш формат. Файл выходит как двоичный `.glb` — единый пакет, содержащий всё. Для классических splatting-рабочих процессов обычно лучший выбор — PLY или SPZ, поскольку их понимают больше инструментов.

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

ГДЕ

Строка меню → Export → 3D Formats → .splat.



ТЕХНИЧЕСКИ

Пишет формат Antimatter15 `.splat` — фиксированные 32 байта на Gaussian (позиция как 3× Float32, масштаб как 3× Float32, поворот как 4× Uint8-нормализованный кватернион, RGB+opacity как 4× Uint8). Никаких SH-коэффициентов выше DC. Наименьший файл с прямой совместимостью с браузером. Для `gsplat.js` и онлайн-демо-просмотрщика `antimatter15`.



## ПРОСТЫМИ СЛОВАМИ

Простейший формат веб-просмотрщика. Маленький и сразу отображаемый в любом браузере. Но теряет детальное освещение (высшие SH-коэффициенты теряются — splat выглядит одинаково с любого угла, вместо того чтобы реагировать на свет). Хорош для максимальной производительности в вебе, но для фотореализма лучше SPZ или PLY. Работает с онлайн-просмотрщиком `antimatter15` и `gsplat.js`. Каждый Gaussian занимает фиксированные 32 байта, что делает формат простым и совместимым — но ценой глубины детализации.

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

ГДЕ

Строка меню → Export → 3D Formats → SOG.

**ТЕХНИЧЕСКИ**

Пишет облако Gaussian в формат SOG. SOG («Self- Organizing Gaussian») — формат PlayCanvas с texture-atlas- раскладкой и WebP-сжатием квантованных данных. Масштабируется с соотношением размеров в 15–20х лучше, чем PLY. Экспорт внутри вызывает `cwebp` как внешний инструмент — так что в sandbox-варианте (App Store) потенциально ограничен.

**ПРОСТЫМИ СЛОВАМИ**

Очень маленький формат для PlayCanvas-рабочих процессов. Примерно в 15–20 раз меньше PLY, потому что данные упакованы в texture-atlas-раскладку и сжаты WebP. Если у вас нет PlayCanvas-рабочего процесса, обычно лучший выбор — SPZ или Compressed PLY. Экспорт внутри вызывает `cwebp` как внешний инструмент — в версии для App Store (sandbox) этот шаг может быть ограничен.

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

ГДЕ

Строка меню → Export → Media → Orbit Video.

**ТЕХНИЧЕСКИ**

Рендерит орбиту 360° вокруг центра сцены и кодирует её как MP4 (H.264) или MOV (HEVC, в зависимости от системного дефолта). В отличие от M18 (живая запись), траектория здесь фиксированная — длительность выбирается в настройках или на шаге экспорта Simple Mode.

**ПРОСТЫМИ СЛОВАМИ**

Автоматически производит видео разворота вашей сцены. Не нужно ручного движения. Хорошо для соцсетей или быстрого демо. Если хочется контролировать камеру самому, используйте вместо этого Record Turntable Video (M18). Траектория фиксирована: полная орбита 360° вокруг центра сцены; длительность выбирается в настройках или на шаге экспорта Simple Mode. Видео выводится как H.264 MP4 или HEVC MOV в зависимости от системы.

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

ГДЕ

Строка меню → Export → Media → Web Viewer.

**ТЕХНИЧЕСКИ**

Упаковывает автономный HTML-просмотрщик (на основе gsplat.js) плюс данные Gaussian, закодированные в Base64, в единый `.html`-файл. Этот файл работает оффлайн в любом современном браузере — никаких серверных зависимостей, никаких внешних URL. Размер файла примерно в 1.3 раза больше SPZ-варианта (из-за overhead Base64).

**ПРОСТЫМИ СЛОВАМИ**

Сохраняет вашу сцену как самозапускающуюся веб-страницу. Двойной щелчок по HTML-файлу → браузер открывается → готовая интерактивная 3D-сцена. Работает без интернета, можно отправить по почте, простейший способ поделиться результатом с друзьями или клиентами. Файл содержит полный просмотрщик gsplat.js и данные Gaussian в одном документе — ничего не загружается из сети. Размер файла примерно на треть больше, чем у SPZ-экспорта, но взамен получателю не нужно дополнительное ПО.

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

ГДЕ

Строка меню → Export → Photogrammetry → Export SfM (transforms.json).

**ТЕХНИЧЕСКИ**

Выделенный путь экспорта (не через общую вспомогательную процедуру), потому что экспортируется не облако Gaussian, а результат SfM. Открывает save-диалог с `transforms.json` как значением по умолчанию и типом контента `json`. По ОК пишется совместимый с nerfstudio `transforms.json` с intrinsics камер, позами (как матрица 4x4 в соглашении NeRF) и путями кадров. Help-текст в UI указывает, что тренировочные изображения должны быть скопированы рядом как соседняя папка `images/`. Активно только если существует результат SfM и полная версия разблокирована.

**ПРОСТЫМИ СЛОВАМИ**

Если хотите использовать результат SfM в другой программе вроде nerfstudio, Brush, gsplat или OpenSplat, здесь вы экспортируете позиции камер. Поместите ваши тренировочные изображения дополнительно в папку `images/` рядом с файлом `transforms.json` — иначе целевая программа не сможет сопоставить изображения. Пункт серый, пока нет результата SfM, и заблокирован в free trial-версии. Для рабочего процесса COLMAP-workspace используйте вместо этого M34.

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



ГДЕ

Строка меню → Export → Photogrammetry → Export SfM (COLMAP Workspace).



ТЕХНИЧЕСКИ

Открывает save-диалог с именем по умолчанию `colmap-workspace` (без расширения, потому что это папка). Пишет стандартную рабочую область COLMAP с `sparse/0/cameras.bin`, `images.bin`, `points3D.bin`. Позволяет открыть SfM-реконструкцию, рассчитанную или импортированную в RadianceKit, в других инструментах вроде Postshot, Nerfstudio или Meshroom, либо для A/B-герана перезагрузить её как уже рассчитанный ввод в самом RadianceKit (через M5) — экономит время вычислений. Активно только если существует результат SfM и полная версия разблокирована.

#### ПРОСТЫМИ СЛОВАМИ

Как M33, но в формате COLMAP вместо nerfstudio. Если вы используете Postshot, Meshroom, Nerfstudio или другой инструмент с COLMAP-рабочим процессом, это ваш экспорт. Удобный побочный эффект: вы можете позднее перезагрузить эту папку через M5 в RadianceKit и сэкономить время вычислений SfM на следующем прогоне — особенно на больших сценах экономия часов. Как и M33, доступно только если существует результат SfM, и заблокировано в free trial-версии.

## Меню Help

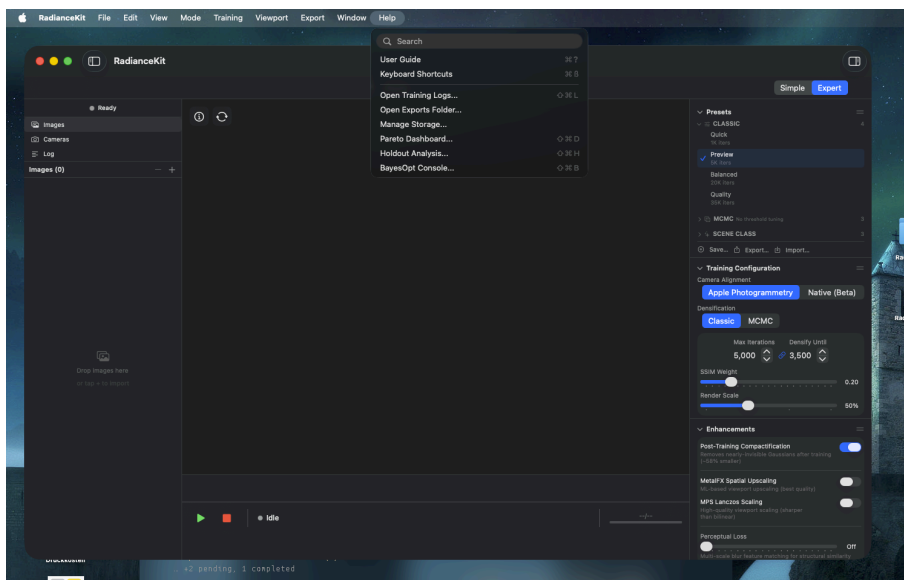


Рисунок 6. Меню Help с пунктами документации, папок и анализа

Семь пунктов: два окна документации (User Guide, Keyboard Shortcuts), три ярлыка папок (Training Logs, Exports, Storage) и три окна анализа (Pareto Dashboard, Holdout Analysis, BayesOpt Console). В стиле Apple меню Help появляется в самом

правом углу. Стандартное меню Help полностью заменено собственным вариантом RadianceKit.

### M35 Help > User Guide



Строка меню → Help → User Guide (⌘?).



Открывает окно User Guide. Оно показывает навигацию с боковой панелью тем и прокручиваемую детальную область при размере по умолчанию 860×640. Содержимое статически встроено (не парсится из Markdown).

#### ПРОСТЫМИ СЛОВАМИ

Открывает внутриприложенческое руководство. Если не хочется читать всё в этом мануале, ключевые шаги вы найдёте прямо в программе. Руководство построено как отдельное окно с боковой панелью тем — так что можно прыгать прямо к отдельным темам. Содержимое короче, чем этот мануал, и сфокусировано на самых распространённых рабочих процессах.

### M36 Help > Keyboard Shortcuts



Строка меню → Help → Keyboard Shortcuts (⌘/).



Открывает окно Keyboard Shortcuts — простая прокручиваемая раскладка со всеми сокращениями приложения, сгруппированными по родительскому меню. Размер по умолчанию 440×560. Содержимое также статически встроено.

#### ПРОСТЫМИ СЛОВАМИ

Открывает окно с полным списком всех клавиатурных сокращений. Если не можете вспомнить, напр., какая клавиша запускает обучение, посмотрите там. Сводка также есть в конце этой главы. Список сгруппирован по родительскому меню, так что вы быстро прыгаете в нужную область. Полезно при переходе со стиля работы мышью на стиль работы клавиатурой.

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

Строка меню → Help → Open Training Logs...  
(⇧⌘L).



Вычисляет папку логов как ~/Documents/RadianceKit/Logs, создаёт её при необходимости и открывает в Finder. Каждый прогон обучения пишет туда свой JSONL-файл `training_YYYY-MM-DD_HHmms.jsonl`.

**ПРОСТЫМИ СЛОВАМИ**

Открывает в Finder папку со всеми предыдущими протоколами обучения. Если что-то пошло не так или хочется посмотреть, когда именно обучение сошлось к какому значению, вы найдёте это здесь в JSONL-файлах. На каждый прогон обучения создаётся ровно один файл с timestamp — его также можно скормить другим инструментам или отправить в support по почте. Если хочется графический анализ, лучшая стартовая точка — Pareto Dashboard (M40).

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

Строка меню → Help → Open Exports Folder...



Аналогично M37, но с ~/Documents/RadianceKit/Exports. Создаётся при первом auto-test-прогоне или при первом щелчке; после этого туда попадают пути по умолчанию всех auto-test-экспортов (напр., `autotest_<timestamp>.ply`). Экспорты, выбранные вручную через save-диалог, НЕ обязательно идут сюда, а туда, куда сохраняет пользователь — так что эта папка интересна в основном для auto-тестов.

**ПРОСТЫМИ СЛОВАМИ**

Открывает папку, куда приложение кладёт собственные экспорты (особенно auto-test-прогоны). Если вы вручную через save-диалог разместили экспорт где-то ещё, он там, а не в этой папке. Удобно для уборки или для проверки, сколько места занимают предыдущие test-экспорты. Если хотите полный обзор, включая логи и scene-bundles, используйте вместо этого Manage Storage (M39).

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

Строка меню → Help → Manage Storage...



Открывает storage browser (см. главу 4 Вспомогательные окна, ID W7–W12). Перечисляет все сохранённые сцены, training-логи, экспорты и кеши в папке ~/Documents/RadianceKit/ с размером, позволяет reveal в Finder и перемещение в корзину на запись.

**ПРОСТЫМИ СЛОВАМИ**

Открывает оконный браузер, который показывает, сколько дискового пространства занимает RadianceKit — на сцену, лог и экспорт. Можно удалять отдельные элементы напрямую, не заходя в Finder. Удобно после продолжительного использования, когда диск заполняется — предыдущие логи и auto-test-экспорты могут складываться в несколько гигабайт. Через reveal в Finder вы в любое время попадаете и в классический вид.

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

Строка меню → Help → Pareto Dashboard...  
(⇧⌘D).



Открывает Pareto Dashboard (см. главу 4, ID W13–W22). Dashboard загружает все JSONL training-логи из ~/Documents/RadianceKit/Logs/, организует их по сценам и предустановкам и рисует Pareto-точечный график (по умолчанию: loss против Gaussians, опционально loss против wallclock или PSNR против итераций).

**ПРОСТЫМИ СЛОВАМИ**

Открывает обзор всех предыдущих прогонов обучения как диаграмму. Сразу видно, какой прогон выдал лучший баланс качества и размера. Удобно, когда хочется сравнить разные предустановки. По умолчанию график показывает loss против числа Gaussians — но можно также переключиться на wallclock- время или PSNR. Данные приходят из JSONL training-логов (M37); чем больше у вас прогонов, тем содержательнее анализ.

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

ГДЕ

Строка меню → Help → Holdout Analysis... (⇧⌘H).

**ТЕХНИЧЕСКИ**

Открывает окно Holdout Analysis (см. главу 4, ID W23–W29). Загружает `transforms.json`, рисует камеры как 3D-глобус и позволяет разбиения `train/test` по фолдам (угловые или линейные, 2–8 фолдов). Вывод — `fold-assignment.json`, который обучение может использовать как тестовый набор в соответствующих конфигурациях обучения.

**ПРОСТЫМИ СЛОВАМИ**

Помогает разделить ваши захваты камер на тренировочный и тестовый наборы — чтобы объективно измерить, насколько хороша ваша сцена (на изображениях, которые обучение никогда не видело). Скорее исследовательский и бенчмарк-инструмент. Камеры показываются как 3D-глобус; можно выбирать между 2 и 8 фолдами, равномерно по углу или линейно по порядку. Результат — маленький JSON-файл, который обучение затем использует как тестовый набор.

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

ГДЕ

Строка меню → Help → BayesOpt Console... (⇧⌘B).

**ТЕХНИЧЕСКИ**

Открывает BayesOpt Console (см. главу 4, ID W30–W39). Загружает предопределённые поисковые пространства (напр., «MCMC scale-reg + opacity-reg + ssim»), запускает `trials` байесовской оптимизации асинхронно и показывает кривую сходимости и `trial`-лог вживую.

**ПРОСТЫМИ СЛОВАМИ**

Встроенная консоль авто-тюнинга. Вместо того чтобы вручную пробовать разные параметры, приложение может работать само на ночь и в конце предложить лучшие значения для вашей сцены. Очень продвинутый инструмент — для большинства рабочих процессов достаточно хорошей предустановки (см. главу 7). Вы выбираете предопределённое поисковое пространство (напр., «MCMC scale-reg + opacity-reg + ssim») и видите вживую кривую сходимости и `trial`-лог. Планируйте от нескольких часов до дней в зависимости от настройки.

**Примечание: Cmd-Z в меню Edit**

С мая 2026 года Project Navigator в Expert Mode поддерживает удаление импортированных изображений через кнопку минус или клавишу `Backspace` и отмену через `Cmd-Z`. Это действие `Cmd-Z` появляется в системном меню `Edit macOS`

(предоставленном SwiftUI) как «Undo Remove Image», пока удалённое изображение всё ещё восстановимо. Оно зарегистрировано через стандарт-конформную систему, не в ; поэтому в инвентаре нет выделенной записи с M-ID.

## Обзор клавиатурных сокращений

Пункт меню	Сокращение
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

Меню Edit (системно-предоставленное, в Expert Mode при активном выборе в Project Navigator):

Действие	Сокращение
Undo Remove Image	⌘Z
Remove Selected Image	Backspace / Delete

## ГЛАВА

## Глава 2 — Инспектор (Expert View)

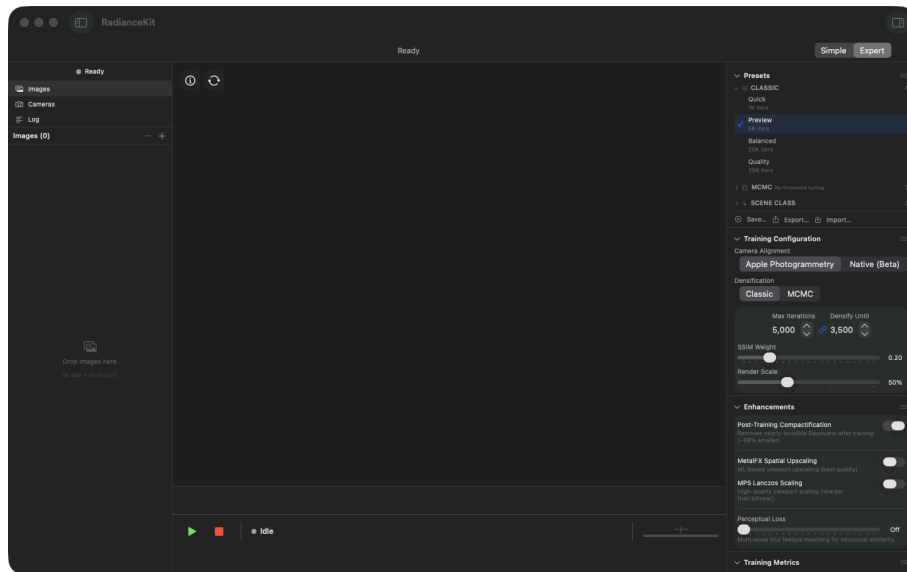


Рисунок 7. Пустой Expert Mode — Project Navigator слева (Images 0, Cameras, Log), пустой viewport посередине, Инспектор справа с секциями Presets/Training Configuration/Enhancements/Training Metrics

**Пустой Инспектор до импорта:** Левая боковая панель показывает счётчик Images 0 и подсказку дропа «Drop images here / or tap + to import». Инспектор справа полностью функционален, но предустановки только информативны (нет активного обучения). Предустановка по умолчанию «Preview» (5К итераций) помечена. Camera Alignment установлен на Apple Photogrammetry, Densification — на Classic, SSIM Weight — 0.20, Render Scale — 50 %. В Training Metrics и Loss History показаны пустые состояния («Start training to see live metrics», «Loss curve will appear during training»).

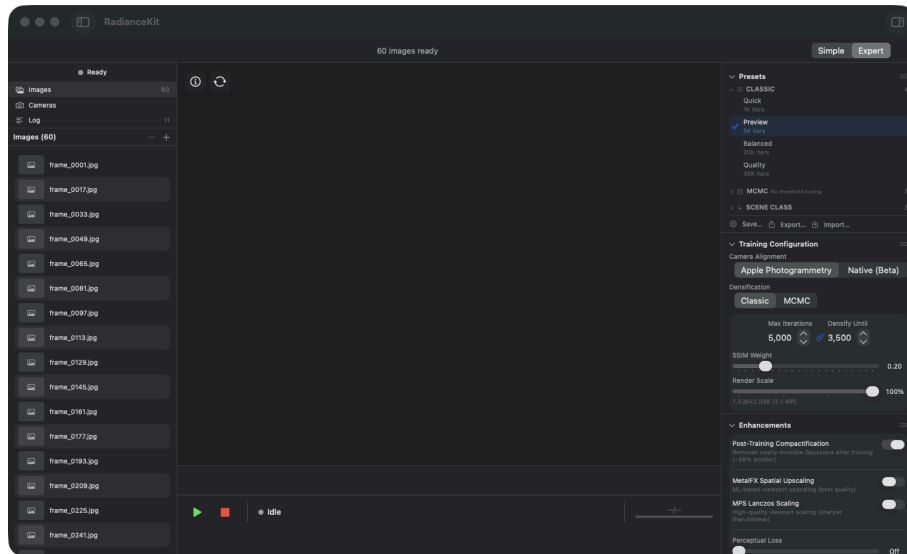


Рисунок 8. Инспектор с загруженными 60 изображениями flowers — боковая панель изображений показывает первые имена файлов frame\_0001.jpg и далее, заголовок гласит «60 images ready»

**Инспектор после импорта:** Статус заголовка гласит «60 images ready». Боковая панель изображений перечисляет все 60 импортированных кадров ( frame\_0001.jpg – frame\_0945.jpg , каждый 16-й кадр из набора Bouquet на 960 камер как подмножество для быстрых итераций). Логика автоматического render scale проверяет разрешение изображений ( $1536 \times 2048 = 3.1 \text{ MP}$ ) и подстраивает Render Scale соответствующим образом. Кнопка Play (зелёная, внизу слева) теперь активна и запускает обучение с активной предустановкой.

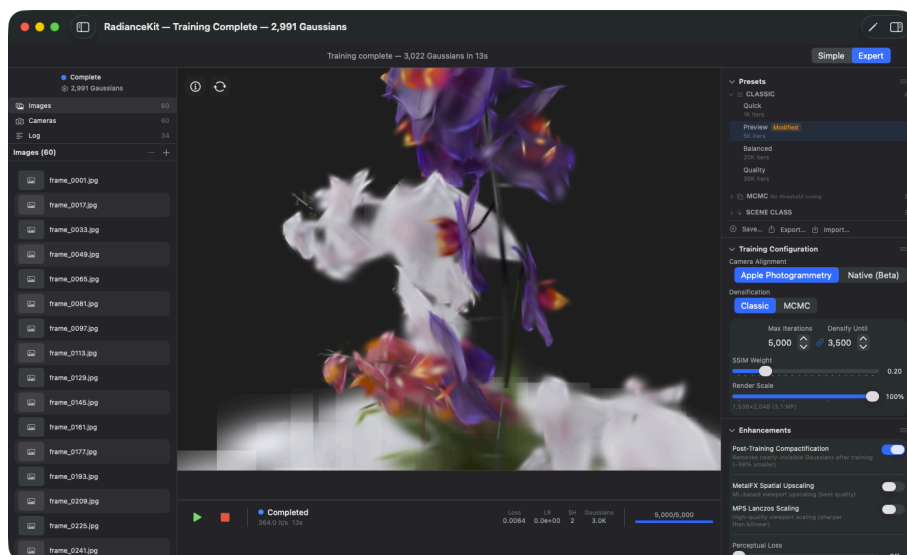


Рисунок 9. Инспектор во время обучения — живой viewport показывает реконструкцию flowers Bouquet, нижняя строка метрик (Loss / LR / Gaussian count / iterations), карточка предустановки «Preview» с пометкой «Modified», если параметры были изменены

**Инспектор во время обучения:** Строка заголовка показывает глобальный прогресс «RadianceKit — Training NN %». Viewport рендерит текущую реконструкцию Gaussian в реальном времени (обновляется каждые 50 итераций — интервал

живого предпросмотра задаётся в Settings → General → Training → Live Preview). Строка метрик под viewport: текущий Loss, Learning Rate, число Gaussians и счётчик итераций (напр. 1 600/5 000 с предустановкой Preview). Карточка пред-установки «Preview» получает пометку «Modified», как только любой параметр отклоняется от встроенного значения по умолчанию. Боковая панель «Log» собирает события этапов SfM и обучения.

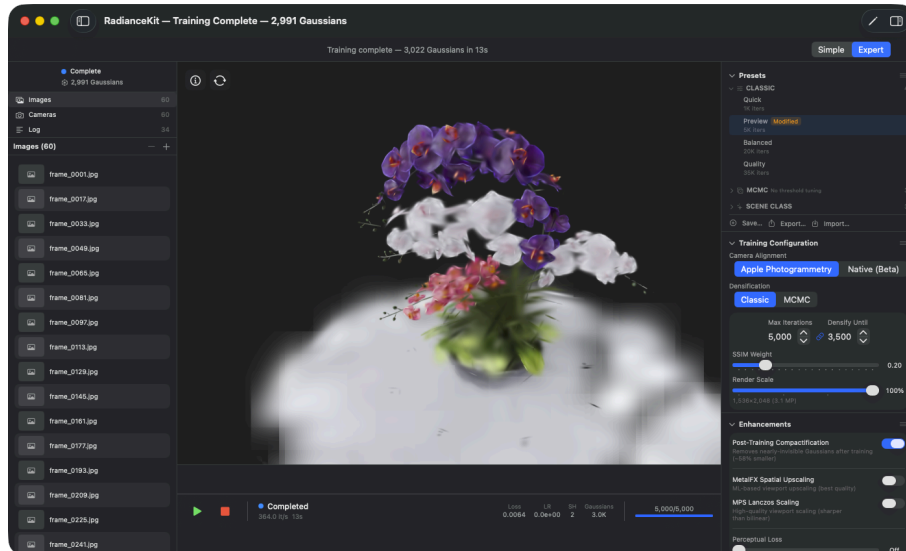


Рисунок 10. Инспектор после завершения обучения — viewport показывает законченную реконструкцию flowers Bouquet (2 991 Gaussians после 5K итераций за 13 с), строка заголовка «Training Complete — 2,991 Gaussians»

**Инспектор после обучения:** Строка заголовка показывает финальное число Gaussians (здесь 2 991 — очень компактно, поскольку синтетическая сцена Bouquet из Blender имеет простую геометрию на светлом фоне). Viewport показывает законченное облако точек — активна orbital drag-навигация (вращает вокруг центра сцены). Секция Training Metrics теперь заполнена финальными значениями, график Loss History показывает ход всех 5 000 итераций. Секция Export внизу теперь активна (все кнопки форматов разблокированы).

Инспектор — это правая боковая панель в Expert Mode (§2). Он объединяет все параметры, относящиеся к обучению, в семь сворачиваемых секций. Порядок по умолчанию сверху вниз при первом запуске такой: Look, Presets, Training Configuration, Metrics, Loss Chart, Enhancements и Export. Секция «Look» (пост-тренировочные коррекции изображения) — это реальное UI-переименование прежней секции «Finishing»: её внутренний enum-rawValue по причинам персистентности остаётся «Finishing», а отображаемый заголовок гласит «Look». Каждая секция может быть свёрнута щелчком по заголовку, а порядок можно менять через drag-and-drop (InspectorView.swift:81–97). **При первом запуске все семь секций свёрнуты** (inspectorCollapsedSections по умолчанию равно Set(InspectorSection.allCases)); затем состояние приложения запоминает предпочтения свёртывания и порядка между запусками.

Ряд элементов управления Инспектора также появляются почти в идентичной форме в Settings (глава 3) — обычно SfM-бэкэнд, sky masking и аналогичные значения по умолчанию. Разделение сделано сознательно: Settings предоставляет

глобальный для приложения шаблон для вновь создаваемых проектов, Инспектор переопределяет эти значения для текущего открытого проекта. Стоит освоить логику работы одной стороны — и другую можно использовать вслепую.

Левая колонка в Expert Mode — Project Navigator — не входит в Инспектор, но является его прямым соседом. Там можно выбирать импортированные изображения щелчком, предпросматривать их по пробелу через Quick Look и удалять через кнопку минус или клавишу Delete (с Cmd-Z для отмены). Инспектор следует за текущим выбором в боковой панели контекстно-зависимой детальной информацией, но семь основных секций всегда остаются доступными.

## Секция Look (L1–L5)

Секция Look (внутренний `rawValue` по-прежнему «Finishing») — это самая верхняя секция Инспектора, собирающая в одном месте **пост-тренировочные** коррекции изображения. Все регуляторы работают **неразрушающе**: каждый слайдер заново применяет `FinishingPass` к неизменному pristine-снимку (оригинальный DC-цвет, -непрозрачность, -масштаб) — так что коррекция **идемпотентна**, а не накопительна. Результат появляется **вживую в viewport** (WYSIWYG, точно так же, как последующий экспорт) и **запекается в каждый экспорт**. Секция доступна лишь **после завершения тренировочного прогона** (до этого стоит «Available after a training run completes.»); её значения **сбрасываются при каждом новом обучении**. Пока идёт экспорт, все регуляторы **заблокированы** — появляется подсказка-замок «Locked while exporting — the file uses the current settings.», а `GroupBox` деактивирован.

## L1 Слайдер Saturation



ГДЕ

Инспектор → секция Look → GroupBox → Saturation.



ТЕХНИЧЕСКИ

Слайдер 0.5–1.2, отображение двузначное (напр. «1.00»). Масштабирует SH-DC-хромю каждого Splat вокруг значения яркости: 1.0 = без изменений, < 1.0 = обесцвечено (цвет тянется к градации серого), > 1.0 = насыщеннее. Математически DC-цвет пересчитывается из pristine-снимка ( `desaturateDC` ), так что повторное перетаскивание не суммируется. Проверено на материале дронов DJI (виадук Pensford), который склонен к перенасыщению — дефолт для дронов лежит на 0.82. Действует только на цветовую базу (SH-степень 0), более высокие SH-коэффициенты остаются нетронутыми.

### ПРОСТЫМИ СЛОВАМИ

Насколько насыщены цвета готового Splat. 1.00 оставляет всё как обучено, значения ниже тянут цвет к серому — хорошо для материала дронов или видео, который часто выходит перенасыщенным. Значения выше 1.0 делают его насыщеннее. Можно сколько угодно гонять туда-сюда, ничего не «раскачивается», потому что приложение всегда считает заново от неизменного оригинала. Видно вживую в viewport и точно так же в экспорте.

## L2 Слайдер Splat length



ГДЕ

Инспектор → секция Look → GroupBox → Splat length.



ТЕХНИЧЕСКИ

Слайдер 0.3–1.0, отображение двузначное. Стягивает три оси масштаба каждого Gaussian в лог-пространстве к их среднему ( `shortenScale` , фактор `alpha` ): 1.0 = без изменений, меньшие значения делают вытянутые «игольчатые» Splat-ы круглее, 0 были бы чистыми сферами. Воздействует на игольчатые, переразвёрнутые Splat-ы, не меняя общий размер, и за счёт этого снижает типичные «конфетти»-артефакты. Применяется от pristine-снимка (оригинальный лог-масштаб), поэтому идемпотентно. Коммутирует со Splat size (L3), потому что оба работают в лог-пространстве.

### ПРОСТЫМИ СЛОВАМИ

Делает слишком длинные, осколочные Splat-ы круглее. 1.00 оставляет форму как обучено, более низкие значения сминают вытянутые «иглы» в более округлые кляксы — это успокаивает зернистые, поражённые конфетти-артефактами реконструкции. Общий размер остаётся прежним, речь только о вытянутости. Безопасно комбинируется со Splat size (L3).

### L3 Слайдер Splat size



ГДЕ

Инспектор → секция Look → GroupBox → Splat size.



ТЕХНИЧЕСКИ

Слайдер 0.5–2.0, отображение двузначное. Масштабирует каждый Gaussian равномерно по **всем** трём осям ( `sizeScale` ): 1.0 = без изменений, < 1.0 = меньше/плотнее/резче, > 1.0 = больше/«пушистее» (заполняет промежутки между Splat-ами). Поскольку масштабы лежат в лог-пространстве, умножение реализуется как аддитивный  $\log(\text{factor})$  -сдвиг — это коммутирует со Splat length (L2), потому что постоянный сдвиг оставляет отклонение-от-среднего нетронутым. От pristine-снимка, то есть идемпотентно. Новое в этой версии.

#### ПРОСТЫМИ СЛОВАМИ

Масштабирует все Splat-ы равномерно больше или меньше. 1.00 — это обученное состояние, значения ниже делают облако точек теснее и резче, значения выше прикрывают промежутки между Splat-ами (выглядит мягче/«пушистее»). Удобно, чтобы визуально закрыть дырявую реконструкцию или наоборот высвободить больше детализации. Без проблем уживается со Splat length (L2) — оба регулятора не влияют друг на друга.

## L4 Fade far region (с суб-слайдерами)



ГДЕ

Инспектор → секция Look → GroupBox → переключатель «Fade far region» плюс суб-слайдеры «Fade start xradius» и «Fade floor».



ТЕХНИЧЕСКИ

Переключатель, активирующий радиальный спад непрозрачности с расстоянием от центроида камер — слабо наблюдаемые «дальние конфетти» на фоне скрываются. **Только для orbit-съёмки:** переключатель деактивирован, когда `finishingContext.fadeEligible` равно `false` (линейные пролёты, слишком мало или вырожденные камеры); тогда вместо суб-слайдеров появляется подсказка «Far-fade applies only to orbit captures (not this scene).» Пригодность определяется по азимутальному покрытию позиций камер (orbit окружает центроид и заполняет многие сектора компаса, линейный пролёт — лишь ~2). Два суб-слайдера управляют геометрией: **Fade start xradius** (1.0–3.0) задаёт внутренний радиус как кратное orbit-радиуса, внутри которого действует полная непрозрачность; **Fade floor** (0.0–1.0) — это фактор непрозрачности далеко за fade-радиусом. Важно: fade **пропускает зону sky-dome** (замороженные Gaussians индексов [0, frozenCount]), чтобы намеренная фоновая купола не затемнялась заодно.

### ПРОСТЫМИ СЛОВАМИ

Скрывает рыхлые остатки на внешнем крае сцены — именно те «дальние конфетти»-комочки, что висят далеко позади при круговых съёмках. Работает только при настоящих orbit-/круговых съёмках; при прямых пролётах дрона или слишком малом числе камер переключатель серый, а подсказка объясняет почему. Если он активен, добавляются два тонких регулятора: «Fade start xradius» задаёт, с какого расстояния (как кратное радиуса облёта) начинается скрывание, «Fade floor» — насколько сильно дальние Splat-ы в итоге остаются видны (0 = совсем прочь, 1 = без изменений). Намеренно реконструированный sky-dome (I44) при этом никогда не трогается — небо сохраняется.

## L5 Кнопка Reset finishing



ГДЕ

Инспектор → секция Look → GroupBox → «Reset finishing» (внизу, маленькая кнопка).



ТЕХНИЧЕСКИ

Сбрасывает все настройки Look на значения по умолчанию (`FinishingPass.Settings()` = Saturation 1.0, fade выключен, Splat length 1.0, Splat size 1.0) и сразу запускает повторный finishing, так что viewport возвращается к неизменному обученному состоянию. `controlSize(.small)`. Поскольку весь Look-стек считает идемпотентно от pristine-снимка, «назад к дефолту» — это в точности исходный вывод обучения, без потери качества от многократного хождения туда-сюда. Как и все регуляторы секции, заблокирована во время идущего экспорта.

### ПРОСТЫМИ СЛОВАМИ

Одним щелчком возвращает все регуляторы Look на стандарт (Saturation 1.00, fade выключен, оба Splat-слайдера на 1.00) — viewport после этого снова показывает точно свежий обученный результат. Удобно, если заигрались и хотите начать чисто с начала. Поскольку приложение всегда считает от оригинала, потери качества при этом нет. Пока идёт экспорт, кнопка (как и слайдеры) заблокирована.

## Секция Presets (I1–I11)

Секция Presets — самый быстрый способ применить протестированную конфигурацию. Встроенные предустановки (Capture Class, Classic, MCMC, Hybrid) дают воспроизводимые отправные точки из 560+ задокументированных экспериментов; собственные предустановки можно сохранять, экспортировать, импортировать и передавать. Список сгруппирован по категориям (Capture Class, Classic, MCMC, Hybrid, Custom), и одновременно может быть развёрнуто несколько категорий. Контекстное меню (правый клик по строке) делает доступными Export, Duplicate, а для пользовательских предустановок — Delete.

## I1 Кнопка Save...



ГДЕ

Инспектор → секция Presets → кнопка Save... (панель действий внизу).



ТЕХНИЧЕСКИ

Открывает popover с текстовым полем и кнопками Save/Cancel. Текущее состояние TrainingConfig сохраняется как новая пользовательская предустановка (JSON-encoded, хранится для всего приложения). Процесс сохранения копирует все 81 параметр обучения плюс текущую стратегию densification. Предустановка автоматически попадает в категорию Custom, независимо от того, из какой встроенной предустановки была получена. Пустые имена и ввод из одних пробелов отклоняются. Уже существующие имена не отклоняются — у каждой предустановки свой внутренний ID, дубли имён технически разрешены, но практически путают.

### ПРОСТЫМИ СЛОВАМИ

Сохраняет вашу текущую конфигурацию как многократно предустановку. Щёлкните по кнопке, введите имя в popover и щёлкните Save — все 81 параметр, включая стратегию densification, попадают под выбранным именем в категорию Custom. Это нужно, когда вы вложили усилия и не хотите возиться с настройками заново в следующем проекте. Особенно удобно для повторяющихся настроек вроде «Drone 4K» или «Indoor fast». Дубли имён технически разрешены, но практически путают — лучше выбрать что-то описательное.

## I2 Текстовое поле Preset Name



ГДЕ

Save-popover → текстовое поле «Preset Name».



ТЕХНИЧЕСКИ

Простое текстовое поле со скруглённой обводкой, широкое. При щелчке по кнопке Save значение берётся как имя предустановки. Лимита длины в UI нет, но сохраняемое имя должно быть JSON-кодируемым и отображаемым в списках UI — эмодзи и умяюты работают. Содержимое автоматически сбрасывается в пустую строку при открытии popover. Кнопка Save остаётся деактивированной, пока поле пусто после trim. Auto-suggest и pre-fill именем текущей активной предустановки нет.

### ПРОСТЫМИ СЛОВАМИ

Сюда вы вводите имя для своей предустановки. Выберите что-то описательное вроде «Drone 4K 30fps» или «Indoor fast» — так позднее проще найти её в категории Custom. Эмодзи и умяюты разрешены, жёсткого лимита длины нет. Пока поле пусто или содержит только пробелы, кнопка Save остаётся серой. Если popover открыт заново, поле снова пусто — pre-fill именем активной предустановки отсутствует.

### I3 Кнопка Cancel (диалог Save)



ГДЕ

Save-porover → кнопка Cancel (слева).

#### ТЕХНИЧЕСКИ

Закрывает porover без сохранения. Отбрасывает содержимое текстового поля — при следующем открытии оно будет сброшено в пустое логикой кнопки Save... (I1). Стандартный стиль кнопки, без диалогов подтверждения и без hotkeys. Текущая TrainingConfig остаётся без изменений, поскольку путь сохранения вообще не выполнялся.

#### ПРОСТЫМИ СЛОВАМИ

Закрывает Save-porover без сохранения чего-либо. Если передумали, опечатались или случайно открыли диалог — просто щёлкните Cancel. Текущая конфигурация обучения остаётся без изменений, поскольку ничего не записано. При повторном открытии porover поле имени снова пусто. Никаких safety-запросов, никаких hotkey — просто щёлкаете, и его нет.

### I4 Кнопка Save (диалог Save)



ГДЕ

Save-porover → кнопка Save (справа, заметный стиль).

#### ТЕХНИЧЕСКИ

Триггерит собственно сохранение. Ещё раз валидирует, что имя непустое (defensive check), и затем пишет текущую TrainingConfig как JSON в app-storage. Затем закрывает porover. Выделено синим, серым пока текстовое поле пусто. Если сохранение не удаётся (напр. из-за полного app-storage — очень маловероятно), видимого диалогов об ошибке пока нет; предустановка просто не появится при следующем запуске приложения.

#### ПРОСТЫМИ СЛОВАМИ

Щелчком по Save вы принимаете имя и записываете вашу текущую настройку как новую предустановку. Porover закрывается, предустановка немедленно появляется в категории Custom списка предустановок, и её затем можно активировать щелчком. Кнопка выделена синим (borderedProminent) и остаётся серой, пока поле имени пусто. Если сохранение не удаётся (напр. UserDefaults полон), видимого диалогов об ошибке нет — предустановка просто будет отсутствовать при следующем запуске приложения.

## I5 Кнопка Export...



ГДЕ

Инспектор → секция Presets → панель действий → кнопка Export....

### ТЕХНИЧЕСКИ

Экспортирует текущую выбранную предустановку в файл `.radiancepreset` (внутри — JSON). Деактивирована, если не выбрана ни одна предустановка. По щелчку приложение открывает save-диалог с именем файла предустановки (имя предустановки

1. расширение `.radiancepreset`). Сохранённый формат содержит

полную TrainingConfig плюс метаданные (имя, категория, ID, built-in-флаг). Двойной щелчок в Finder открывает приложение, но **не** автоматически импорт; пользователь должен воспользоваться кнопкой Import (I6).

### ПРОСТЫМИ СЛОВАМИ

Выберите предустановку в списке и щёлкните Export — затем её можно сохранить как файл `.radiancepreset` и отправить коллеге, например, или перенести на второй Mac. Получатель её снова кнопкой Import... (I6). Работает одинаково хорошо для встроенных и собственных пользовательских предустановок. Кнопка серая, пока в списке ничего не нажато. Совет: через контекстное меню (I8) ещё быстрее — там не нужно сначала выбирать предустановку.

## I6 Кнопка Import...



ГДЕ

Инспектор → секция Presets → панель действий → кнопка Import....

### ТЕХНИЧЕСКИ

Открывает файловый диалог, разрешающий только файлы `.radiancepreset` (multiselect отключён). При выборе JSON-файл загружается, валидируется и вставляется в категорию Custom — с новым внутренним ID, чтобы не было коллизий со встроенными. Импорт автоматически устанавливает категорию в Custom, даже если экспортированная предустановка изначально была, напр., built-in. Повреждённые или несовместимые со старой версией схемы файлы тихо отклоняются, без диалога ошибки (консольный лог информацию даёт).

### ПРОСТЫМИ СЛОВАМИ

Читает файл `.radiancepreset` с диска. Полезно, когда кто-то присылает вам проверенную настройку или вы сами хотите синхронизировать любимые предустановки между несколькими Mac. Импортированные предустановки всегда попадают в категорию Custom — даже если они изначально были экспортированы из встроенных. Повреждённые или устаревшие файлы тихо игнорируются; консольный лог показывает причину. Multiselect в диалоге отключён, так что только один файл на щелчок.

## I7 Строка предустановки (активация щелчком)



ГДЕ

Инспектор → секция Presets → каждая строка предустановки в каждой категории.



ТЕХНИЧЕСКИ

Щелчок по строке предустановки заменяет все поля TrainingConfig значениями из предустановки, запоминает ID активной предустановки и сбрасывает статус Modified. Активная галочка перед строкой появляется только когда предустановка выбрана и не изменена. Как только значение в TrainingConfig меняется (слайдер, степпер, переключатель в других секциях Инспектора), после имени появляется оранжевая пометка «Modified». Встроенные предустановки нельзя перезаписать — при модификации нужно создать копию кнопкой Save (I1).

### ПРОСТЫМИ СЛОВАМИ

Щелчок по строке активирует предустановку и принимает все сохранённые в ней значения в текущие настройки обучения. Галочка перед именем показывает, какая предустановка сейчас активна. Как только после этого вы подкрутите любой слайдер, степпер или переключатель в других секциях, после имени появится оранжевая пометка «Modified» — потому что ваша настройка теперь отклоняется от предустановки. Встроенные предустановки перезаписать нельзя; если хотите сохранить изменения, создайте копию кнопкой Save... (I1) или продублируйте предустановку (I9).

## I8 Контекстное меню «Export...»



ГДЕ

Правый клик на любой строке предустановки → первая запись «Export...».



ТЕХНИЧЕСКИ

Идентичная функциональность с I5 (кнопка Export...), но удобнее доступна — без необходимости предварительно выбирать предустановку. Экспортирует напрямую предустановку из строки, по которой щёлкнули. Работает одинаково для всех категорий (built-in или custom), без ограничений. Экспорт содержит built-in-флаг и оригинальную категорию, но при повторном импорте категория мапится в Custom, как описано под I6.

### ПРОСТЫМИ СЛОВАМИ

Быстрый способ экспорта — правый клик по желаемой предустановке и выбор «Export...». Экономит обход через предварительный щелчок и затем нажатие кнопки Export.... Работает одинаково для всех категорий, в том числе для встроенных. Создаваемый файл `.radiancepreset` идентичен тому, что из I5; при последующем повторном импорте он автоматически попадает в категорию Custom.

## I9 Контекстное меню «Duplicate»



ГДЕ

Правый клик на любой строке предустановки → вторая запись «Duplicate».



ТЕХНИЧЕСКИ

Клонирует предустановку в категорию Custom. Создаёт новый внутренний ID, добавляет «Сору» к имени и сохраняет копию. Работает и для встроенных предустановок — клон тогда редактируем. Оригинал остаётся нетронутым. TrainingConfig копируется значение-за-значением (JSON round-trip), так что нет reference-bindings между оригиналом и копией.

### ПРОСТЫМИ СЛОВАМИ

Создаёт редактируемую копию предустановки в категории Custom. Удобно, если, напр., вы хотите взять встроенную предустановку «Quality» как отправную точку и потом подкрутить только SSIM-слайдер. Рабочий процесс: duplicate, переименовать (контекстное меню или новый Save...), подстроить, готово. Оригинал остаётся нетронутым — к нему можно вернуться в любой момент. Работает и для встроенных, что — единственный способ использовать их значения как основу и одновременно сделать редактируемыми.

## I10 Контекстное меню «Delete»



ГДЕ

Правый клик на ваших собственных строках предустановок → последняя запись «Delete» (красная, destructive).



ТЕХНИЧЕСКИ

Видна только для пользовательских предустановок. Встроенные удалить нельзя. Запись помечена как destructive, появляется в контекстном меню красной и отделена разделителем, чтобы не щёлкнуть случайно. Диалога подтверждения **нет** — щелчок сразу удаляет предустановку. Удалённая предустановка не восстанавливается (Cmd-Z здесь не работает — Undo в текущей сборке существует только для списка изображений, не для операций с предустановками). Если удалённая предустановка только что была активной, текущая TrainingConfig остаётся без изменений, обнуляется только выбор активной предустановки.

### ПРОСТЫМИ СЛОВАМИ

Удалите свои предустановки. Для встроенных (Quick, Preview, Balanced, Quality, Ultra Detail, Drone / Aerial, 360° Walkaround, Photo / Object и т.д.) «Delete» вообще не виден — их случайно не убить. Внимание: ни safety-запроса, ни undo, один щелчок — и предустановки нет. Если не уверены, сначала вытяните backup на диск через Export... (I5/I8) — его можно повторно импортировать в любой момент. Если предустановка только что была активной, ваша TrainingConfig остаётся без изменений, исчезает только галочка.

## I11 Заголовок категории (expand/collapse)



ГДЕ

Инспектор → секция Presets → каждый заголовок категории (Capture Class, Classic, MCMC, Hybrid, Custom).

### ТЕХНИЧЕСКИ

Состояние свёртывания на категорию с разными значениями по умолчанию: курируемая группа Capture Class стартует **развёрнутой**, Classic, MCMC, Hybrid и Custom — **свёрнутыми**. Состояние не сохраняется — при перезапуске приложения все категории снова в значении по умолчанию. Стрелка-шеvron поворачивается с анимацией. Число справа в заголовке показывает количество предустановок в этой категории. Hit-area щелчка охватывает всю область заголовка.

### ПРОСТЫМИ СЛОВАМИ

Разворачивайте и сворачивайте категории, чтобы держать список предустановок аккуратным. При старте приложения группа Capture Class открыта, Classic, MCMC, Hybrid и Custom — закрыты. Щелчок по заголовку (вся область кликабельна), и список выезжает или сжимается с короткой анимацией шеврона. Маленькое число справа показывает, сколько предустановок в категории. После перезапуска приложения — снова состояние по умолчанию — приложение сознательно не сохраняет эту настройку свёртывания.

## Секция Training Configuration (I12–I22)

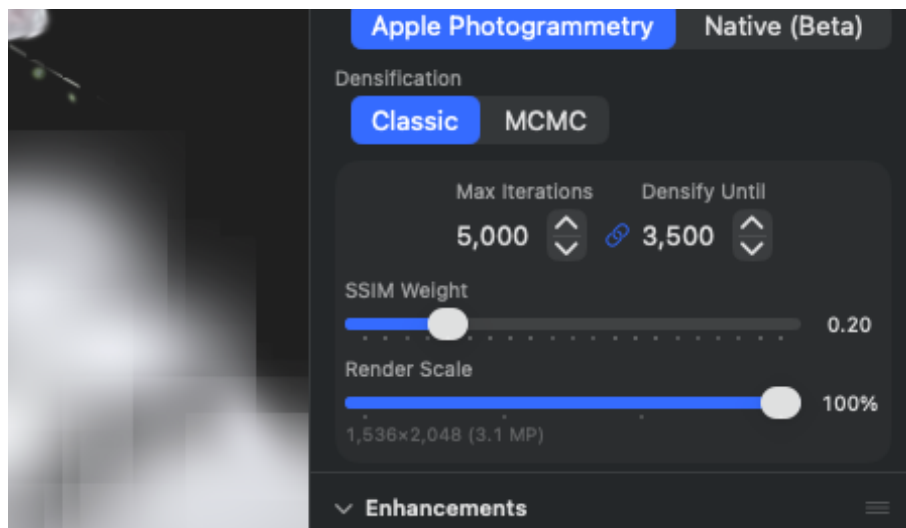


Рисунок 11. Вырезка только секции Training Configuration — Camera Alignment (Apple Photogrammetry активна, Native (Beta) неактивна), Densification (Classic активна), Max Iterations 5,000 / Densify Until 3,500 со значком link, слайдер SSIM Weight 0.20, слайдер Render Scale на 100 % (1,536×2,048 = 3.1 MP)

Здесь живут центральные рычаги: какой SfM-бэкенд должен считать, как работает densification, сколько итераций, сколько веса SSIM. При стратегии MCMC появляются два дополнительных переключателя («MCMC Quality» и «Auto-scale by

scene»), скрытые в Classic-режиме. При бэкенде Native SfM добавляется поле FOV override, нужное только для видеокадров без EXIF focal length.

## I12 Picker Camera Alignment



ГДЕ

Инспектор → Training Configuration → Camera Alignment (сегментированный picker наверху).



ТЕХНИЧЕСКИ

Сегментированный picker с двумя опциями: Apple Photogrammetry и Native (Beta). Выбор определяет SfM-бэкенд, используемый при следующей реконструкции камер. Одновременно он влияет, какие дополнительные элементы Инспектора видны: Native дополнительно показывает FOV override (I13), нужный только для видеокадров без EXIF. Примечание: для очень больших уличных съёмок можно через workspace import импортировать результат внешнего инструмента (Metashape или COLMAP) — см. главу 1 (M5) и главу 9 (Q3, Q6).

### ПРОСТЫМИ СЛОВАМИ

Здесь вы выбираете, как реконструируются позиции камер — важнейший переключатель для итогового качества. Apple Photogrammetry — быстрый стандарт и достаточен для большинства сканов объектов. Native (Beta) — совместимая с App Store собственная разработка, хороша для орбит и сцен с turntable и требует FOV override (I13) при видеокадрах без EXIF. При очень больших уличных наборах можно альтернативно рассчитать камеры в Metashape или COLMAP и загрузить результат через workspace import. Подробности и рекомендации по типу сцен — в главе 9.

## I13 Поле FOV Override (Native SfM)



ГДЕ

Инспектор → Training Configuration → FOV Override (видно только при Camera Alignment = Native).



ТЕХНИЧЕСКИ

Числовое текстовое поле (диапазон 0–170°), по умолчанию 0 = автоматическое определение из EXIF или эвристика. Ручной ввод нужен, когда входные изображения извлечены из видео, не содержащего метаданных focal length. Типичные значения: iPhone Wide ≈ 73°, DJI Mavic Wide-Crop ≈ 70°, дрон с full-frame-сенсором ≈ 84°. Значение ограничивается [0, 170] — выходящие значения сразу отбрасываются обратно. Влияет только на нативный SfM-конвейер (Q4/Q5); Apple Photogrammetry это значение полностью игнорирует.

### ПРОСТЫМИ СЛОВАМИ

Если у ваших изображений нет EXIF (типично для извлечённых видеокадров), введите здесь горизонтальное поле зрения камеры в градусах. Ориентировочные значения: iPhone Wide ≈ 73°, DJI Mavic Wide-Crop ≈ 70°, дрон с full-frame-сенсором ≈ 84°. 0 позволяет приложению угадывать самому — часто это срабатывает, но с редкими объективами может пойти не так. Значения свыше 170° автоматически ограничиваются обратно. Поле видно и эффективно, только если выбрано Native в Camera Alignment (I12) — Apple Photogrammetry полностью его игнорирует.

**I15 Picker Densification**

Инспектор → Training Configuration → Densification (сегментированный picker, всегда виден).

 **ТЕХНИЧЕСКИ**

Переключает между двумя стратегиями densification: Classic (оригинальная процедура 3DGS с clone/split/prune и gradient threshold) и MCMC (Stochastic Gradient Langevin Dynamics с Relocation, NeurIPS 2024). При переключении с Classic на MCMC приложение автоматически устанавливает MCMC-специфичные поля в проверенные значения по умолчанию (reg weights = 0, MCMC cap multiplier 3.0, sample/noise schedule). Без этой автоматической инициализации сессии со старыми предустановками страдали от 1.4.4 MCMC-collapse-бага (460K→5 Gaussians, watchdog kill). Выбор picker дополнительно определяет, какие элементы Инспектора видны — при MCMC появляются I16/I17. Подробный эффект полей — в главе 6, T11–T16 (Classic) и T61–T73 (MCMC).

 **ПРОСТЫМИ СЛОВАМИ**

Центральный выбор стратегии для роста числа Gaussian. Classic хорошо настроен по 459 экспериментам, даёт быстро и качественно результаты и не требует знания каких-либо MCMC-полей. MCMC — более новый подход (NeurIPS 2024), более воспроизводимый и обходящийся без ручной настройки порогов — но он считает примерно в 6 раз дольше для сравнимого качества. При переключении на MCMC приложение автоматически устанавливает безопасные значения по умолчанию, чтобы обучение не наткнулось на 1.4.4-collapse. Подробности о полях стратегии — в главе 6 (T11–T16 Classic, T61–T73 MCMC).

## I16 Переключатель MCMC Quality



ГДЕ

Инспектор → Training Configuration → MCMC Quality (только при `Densification = MCMC`).



ТЕХНИЧЕСКИ

Переключает `gradient accumulation` на 2 шага (актив) или 1 шаг (неактив). Накапливает градиенты из двух последовательных видов камеры перед выполнением шага оптимизатора. Эмпирически (Session 33, V544a) уменьшает финальную L1-ошибку примерно на 6 % (0.0246 с Quality против 0.0261 без, в среднем по 3 прогонам на Horse-Full- MCMC). Цена: удвоенное время обучения. При очень длинных тренировках (200K итераций) это приводит к 10+ минутам дополнительного ожидания — так что оправдано только если действительно нужны последние проценты качества. Влияет только на обучение, не на формат экспорта или отображение в viewport.

### ПРОСТЫМИ СЛОВАМИ

Режим качества для MCMC с `gradient accumulation` по двум видам. Эмпирически делает финальный результат примерно на 6 % лучше (L1 0.0246 вместо 0.0261 в Horse-тесте), но стоит вдвое дольше. Если у вас и так идёт 200K MCMC-обучение (легко 2 часа), добавляется ещё около часа. Оправдано для финальных `showcase`-рендеров или в конце `quality-sweep`-сессии, меньше — в ежедневном рабочем процессе. Видно только когда `Densification` установлен на MCMC (I15).

## I17 Переключатель Auto-scale by scene



ГДЕ

Инспектор → Training Configuration → Auto-scale by scene (только при MCMC).



ТЕХНИЧЕСКИ

Когда активен, масштабирует эффективный потолок `max-Gaussians` по числу SfM init-точек × MCMC `cap multiplier` (по умолчанию 3.0). Пример: SfM даёт 250K init-точек, `base cap = 150K`, `multiplier 3.0` → эффективный потолок =  $\max(150K, 750K) = 750K$ . При деактивации строго применяется только `base`. Был введён для v1.4.5, потому что большие уличные съёмки свыше 1000 кадров и соответственно высокой плотностью SfM-точек морили `densification` жёстким `cap 150K` по умолчанию — избыточные точки оставались, новые не могли появляться. По умолчанию OFF в пользовательских предустановках, ON во встроенных MCMC. Влияет только на время обучения, не на экспорт.

### ПРОСТЫМИ СЛОВАМИ

Позволяет максимальному числу Gaussians расти вместе с размером сцены (точнее: с числом SfM init-точек). На маленьких сценах разница едва заметна, на больших уличных часто решающая для качества — иначе обучение «задыхается», потому что потолок по умолчанию 150K слишком низок для сцены. Был введён для v1.4.5 после того, как очень большие уличные наборы (свыше 1000 кадров) заметно упорались в `cap`. Во встроенных MCMC-предустановках включён заранее; в ваших собственных предустановках выключен по умолчанию.

## I18 Стеллер Max Iterations



ГДЕ

Инспектор → Training Configuration → GroupBox → Max Iterations.



ТЕХНИЧЕСКИ

Стеллер с диапазоном 1 000–100 000, шаг 1 000. Определяет общее число итераций оптимизатора. Линейно коррелирует с временем обучения (половина = прим. 50 % времени). Эмпирические sweet spots: 20K (Classic Balanced,  $L1 \approx 0.028$ ), 40K (Classic Quality,  $L1 \approx 0.023$ ), 200K (MCMC Full,  $L1 \approx 0.0246$ ). Свыше 40K с Classic в среднем едва ли что-то даёт — diminishing returns. При изменении, если активна link-функция (I19), Densify Until пропорционально подтягивается (отношение по умолчанию: 0.5, т.е.  $\text{Densify-Until} = \text{Max}/2$ ).

### ПРОСТЫМИ СЛОВАМИ

Сколько шагов обучения выполняется — больше лучше, но и стоит линейно больше времени. Правило большого пальца: 20 000 для хорошего качества, 40 000 для оптимума со стратегией Classic (выше в среднем едва что-то даёт). MCMC требует существенно больше, 200 000 здесь — стандарт. Удвоение итераций примерно удваивает время обучения. С активной кнопкой link (I19) Densify Until подтягивается пропорционально — практически всегда то, что вы хотите.

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



ГДЕ

Инспектор → Training Configuration → GroupBox → маленькая link-кнопка между Max Iterations и Densify Until.



ТЕХНИЧЕСКИ

Кнопка-переключатель, замораживающая соотношение Densify Until к Max Iterations. Когда активна (link-значок выделен), при каждом изменении Max Iterations Densify Until подтягивается пропорционально. Когда unlinked (значок link-plus), значения остаются независимыми. По умолчанию linked, потому что это отражает типичную корреляцию — когда вы тянете обучение на удвоение итераций, обычно хотите, чтобы densification тоже шёл пропорционально дальше. Соотношение вычисляется из текущего значения при установке link-кнопки; типичное соотношение — 0.5 ( $\text{Densify-Until} = \text{половина числа итераций}$ ).

### ПРОСТЫМИ СЛОВАМИ


Маленькая кнопка-скрепка между Max Iterations и Densify Until. Когда активна (link-значок выделен), оба значения движутся вместе — если удвоить итерации, Densify Until тоже удваивается в том же отношении. Когда нет (значок `link.badge.plus`), их можно задавать независимо. По умолчанию linked, потому что это отражает типичную корреляцию — более долгое обучение обычно хочет более долгой densification-фазы. В 99 % случаев оставляйте заблокированным.

**I20** **Степпер Densify Until**

Инспектор → Training Configuration → GroupBox → Densify Until.



Степпер с диапазоном 500–50 000, шаг 500. Определяет индекс итерации, начиная с которого больше не добавляются новые Gaussians через clone/split (Classic) или relocation (MCMC). После достижения уточняются только позиция и цвет. Более высокие значения = больше Gaussians = больший файл, более долгое per-iteration-время (+30–60 % GPU-времени на шаг). Типичные значения: 15К (для 30К max-iter), 20К (для 40К), 100К (для 200К MCMC). При активном link (I19) автоматически масштабируется. Действует по-разному при Classic vs MCMC: Classic полностью останавливает рост, MCMC останавливает relocation-логику, но адаптация sample/noise продолжает работать.

 **ПРОСТЫМИ СЛОВАМИ**

До какой итерации могут добавляться новые Gaussians — при Classic через clone/split, при MCMC через relocation. После этого речь идёт только о цвете и уточнении формы существующих точек. Выше = больше детализации, но и больший файл и +30–60 % GPU-времени на шаг. Типичные значения: 15К (для 30К max-iter), 20К (для 40К), 100К (для 200К MCMC). Обычно висит через link (I19) на Max Iterations — редко имеет смысл разъединять это вручную.

**I21** Слайдер SSIM Weight

ГДЕ

Инспектор → Training Configuration → GroupBox → SSIM Weight.



ТЕХНИЧЕСКИ

Слайдер 0.0–1.0 с шагом 0.05, отображается как «0.20». Смешивает L1 loss (0.0) и SSIM loss (1.0). L1 ужесточает яркость на пиксель, SSIM ужесточает структурное сходство (края, локальные статистики). По умолчанию 0.2 — значение из оригинальной статьи 3DGS (Kerbl 2023) и обратно-инженерно определено как устойчивый компромисс в многочисленных сессиях. Более высокие значения (0.5+) способствуют сохранению деталей, но могут игнорировать локальные ошибки яркости. Более низкие значения ( $< 0.1$ ) ведут к потере деталей на резких краях. Вычисление SSIM работает в шейдере с гауссовым окном  $11 \times 11$ . Производительность: при 0.0 (только L1) обучение примерно на 8–12 % быстрее, поскольку вычисление SSIM в шейдере пропускается.

## ПРОСТЫМИ СЛОВАМИ

Насколько сильно структурное сходство изображений (края, локальные паттерны) взвешивается против чистого сравнения яркости. 0.2 — стандарт из оригинальной статьи 3DGS и достаточен почти для всех сцен. Выше (0.5+) для тонких структур вроде волос, меха или растительности — там больший структурный вес помогает. Ниже (0.0) делает обучение примерно на 8–12 % быстрее, потому что вычисление SSIM в шейдере пропускается, но стоит деталей на резких краях. Если у вас нет хорошей причины для изменения, оставьте 0.2 как есть.

**I22** Слайдер Render Scale

ГДЕ

Инспектор → Training Configuration → GroupBox → Render Scale.



ТЕХНИЧЕСКИ

Слайдер 0.25–1.0 с шагом 0.25, отображается как «100 %». Масштабирует разрешение рендеринга обучения относительно размера исходного изображения. Самый большой рычаг производительности: 50 % сокращает GPU-время прим. на 75 % (поскольку в 4x меньше пикселей), 25 % — прим. на 94 %. Gradient threshold автоматически масштабируется. Под слайдером появляется живое отображение разрешения в MP (напр. «2304x1296 (3.0 MP)»). Если текущее значение отклоняется от рекомендованного, оранжевым текстом показывается «— recommended: 50 %». Рекомендация целится в ~3 MP эффективного разрешения — диапазон, наиболее эффективно обрабатываемый GPU Apple Silicon. Для 4K-исходных изображений рекомендуется, напр., 25 %, для FullHD — 100 %. Изменение дополнительно триггерит buffer reallocation.

**ПРОСТЫМИ СЛОВАМИ**

При каком разрешении обучение рендерит — один из самых больших рычагов производительности. Full (100 %) даёт лучшее качество, но стоит много GPU-времени на больших изображениях. Half (50 %) экономит примерно 75 % GPU-времени, потому что считается вчетверо меньше пикселей — идеально для 4K-источников. Под слайдером видно эффективное разрешение в мегапикселях; приложение целится в около 3 MP, поскольку это эффективнее всего работает на Apple Silicon. Если ваше значение отклоняется, приложение показывает оранжевую подсказку «recommended» — обычно имеет смысл ей следовать.

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

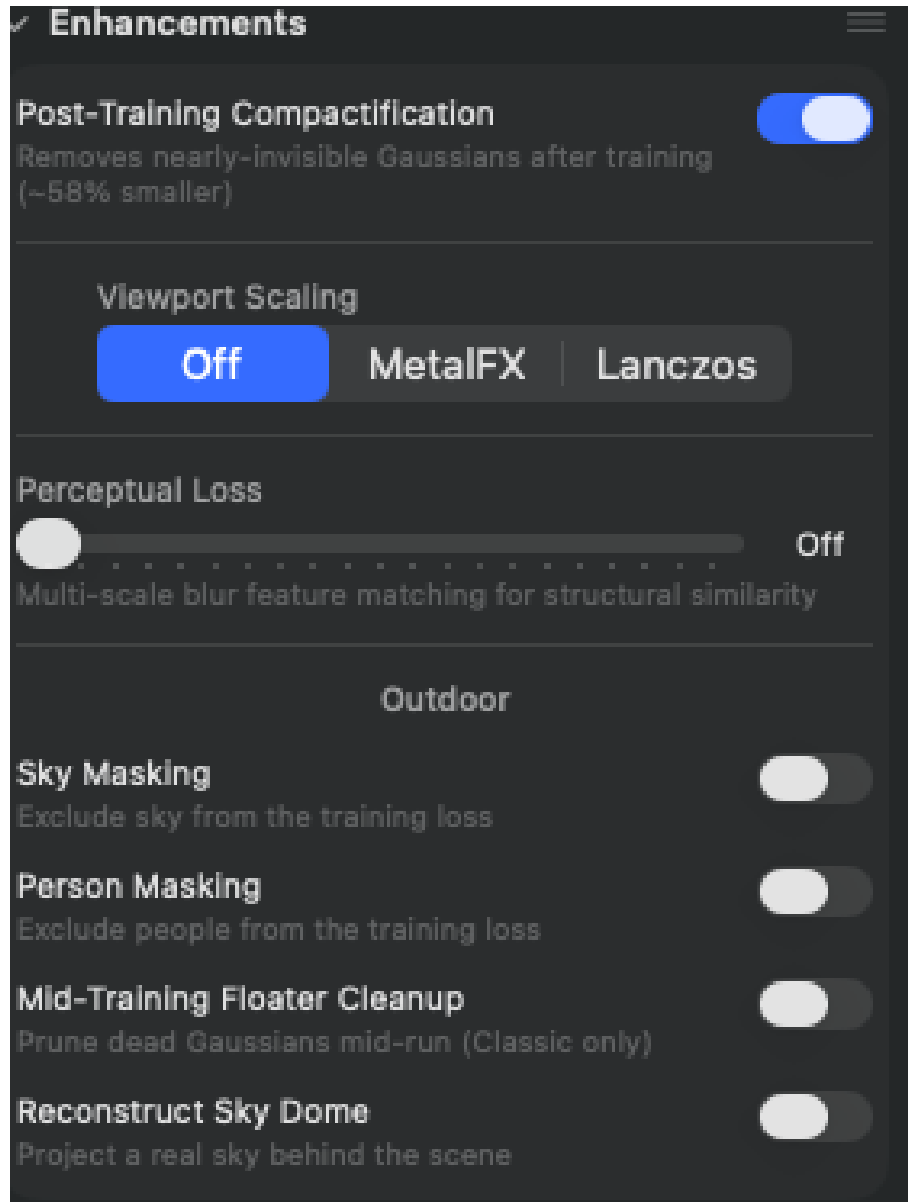


Рисунок 12. Вырезка только секции Enhancements — три строки: Post-Training Compactification (переключатель on), Viewport Scaling (сегментированный picker Off/MetalFX/Lanczos), Perceptual Loss (слайдер на «Off»). Каждая строка с подзаголовком объясняет свою функцию

Секция Enhancements группирует три функции, которые улучшают качество изображения, не меняя сам базовый цикл обучения. Первые две (I26–I27) — это **post-training** или **viewport-этапы**: Compactification наводит порядок после окончания обучения, а Viewport Scaling — чистый viewport-рендерер, не влияющий на идущее обучение. Perceptual Loss (I29), несмотря на принадлежность к секции, является компонентой обучения — активируется как дополнительный loss-член во время обучения, поэтому отделена от viewport-переключателей разделителем. Начиная с v1.6 секция дополнительно содержит группу Outdoor (I42–I44: Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome) — тренировочные опции против

floaters в небе, которые раньше жили в окне Settings, а теперь сидят здесь для каждого проекта.

## I26 Переключатель Post-Training Compactification



Инспектор → Enhancements → Post-Training Compactification.



Активирует V443 post-processing: после завершения итераций обучения Gaussians с непрозрачностью ниже 0.01 (1 % видимости) удаляются. Эмпирически это уменьшает размер файла на ~55–58 % при нулевой видимой потере качества — потому что эти Gaussians и так не вносят визуального вклада. Compactification работает как GPU compact pass и занимает от долей секунды до нескольких секунд в зависимости от числа Gaussians. Не влияет на производительность обучения. Если этот переключатель выключен, невидимые Gaussians тоже экспортируются — актуально только если хотите использовать формат для другого этапа обучения (Continue Training), иначе впустую потраченное хранилище.

### ПРОСТЫМИ СЛОВАМИ

Чистит после обучения Gaussians, которые всё равно не видны (непрозрачность ниже 1 %). Делает экспорт-файлы примерно вдвое меньше (55–58 % уменьшения размера) без видимой потери качества. Работает как короткий GPU-pass после последней итерации, занимает только доли секунды или несколько секунд. Практически всегда должно быть включено — единственная причина это выключить — если позднее хотите продолжить обучение через Continue Training и нужно сохранить невидимые Gaussians тоже. С обычными рабочими процессами экспорта просто оставьте включённым.

**I27 Picker Viewport Scaling**

ГДЕ

Инспектор → Enhancements → Viewport Scaling (сегментированный picker с тремя опциями: Off, MetalFX, Lanczos).

**ТЕХНИЧЕСКИ**

Единственный сегментированный picker, выбирающий viewport-upscaler — три опции **взаимоисключающие**. Когда разрешение обучения (через I22 Render Scale) ниже размера viewport, выбранный режим масштабирует отрендеренный кадр до размера дисплея. **Off** = простое билинейное растягивание. **MetalFX** = ML-based Spatial Upscaler от Apple, самая резкая опция (ML-модель оптимизирована на резкие края), overhead прим. 1–2 мс на кадр на GPU M3. **Lanczos** = Apple Metal Performance Shaders с 8-tap sinc-resampling, классически без ML, минимальный overhead (< 0.5 мс), качество ниже MetalFX, но без ML-типичного «заглаживания» тонких линейных структур. Конвейер рендерера перекоифигурируется вживую при переключении — видно немедленно, без перезапуска. **Предыстория:** раньше это были два отдельных переключателя (MetalFX + Lanczos), которые могли быть включены одновременно — противоречивое состояние, в котором MetalFX молча перекрывал Lanczos. Picker устраняет это состояние; возможно унаследованное от старых сессий состояние «оба включены» самоисцеляется при следующем переключении в MetalFX. Действует **только** на живой viewport, не на отрендеренные экспорты (orbit video, скриншоты) — те рендерятся в полном разрешении источника.

**ПРОСТЫМИ СЛОВАМИ**

Заостряет живое изображение в viewport — особенно полезно, если вы работаете с уменьшенным разрешением обучения (Render Scale 50 %, см. I22). Три ступени, из которых всегда активна только одна: «Off» просто растягивает пиксели, «MetalFX» использует машинное обучение Apple для самых резких краёв (практически всегда лучший выбор), «Lanczos» — классический фильтр без ML — возьмите его как fallback, если MetalFX в какой-то сцене заглаживает линии или показывает артефакты. Вступает в силу вживую, без перезапуска. Действует только на живой viewport, не на экспортированные orbit-видео или скриншоты — те рендерятся в полном разрешении источника. В отличие от прежнего поведения, вы больше не можете случайно выбрать два режима одновременно.

**I29** Слайдер Perceptual Loss

ГДЕ

Инспектор → Enhancements → Perceptual Loss.

## ТЕХНИЧЕСКИ

Слайдер 0.0–0.2 с шагом 0.01, при 0.0 отображается как «Off», иначе как «0.05» и т.д. Активирует дополнительный loss-член, сравнивающий многомасштабное гауссово размытие рендеринга с ground truth-изображением (3 blur scale). Захватывает структурные различия, которые L1+SSIM сами по себе не обнаруживают. Реализация V460. Эмпирически значение 0.05–0.1 улучшает L1-балл в сессиях на несколько процентов, но стоит ~5 % времени обучения (дополнительный forward pass через blur-kernel). Свыше 0.15 обучение становится нестабильным, и L1 снова ухудшается (loss-член доминирует в оптимизации). Действует **во время** обучения, не в post-processing — несмотря на позицию в секции «Enhancements» это не чистая post-обработка.

## ПРОСТЫМИ СЛОВАМИ

Дополнительный loss-член, проверяющий структурное сходство изображений по трём разным стадиям размытия. Особенно помогает на сценах с тонкими структурами вроде волос, тканей или растительности, потому что захватывает паттерны, которые L1+SSIM сами по себе не видят. Меньшие значения безопаснее — 0.05–0.1 — sweet spot, выше 0.15 обучение становится нестабильным, и loss снова ухудшается. При 0 (Off) функция полностью выключена и ничего не стоит; активная съедает примерно 5 % времени обучения на дополнительный forward pass через blur-kernel. Действует, несмотря на секцию «Enhancements», прямо во время обучения, не только в post-processing.

**I42 Маскирование неба**

ГДЕ

Инспектор → Enhancements (группа Outdoor) → переключатель «Sky Masking». Bound: `AppState.trainingConfig.skyMaskingEnabled` (для каждого проекта, `@DefaultFalse`). По умолчанию: `false`.



ТЕХНИЧЕСКИ

Активирует pre-training сегментацию пикселей неба на основе Apple Vision. Перед стартом обучения для каждой входной камеры через Apple Vision foreground mask (Sky = Background) извлекается область неба и назначается соответствующей камере как per-pixel-маска. Во время обучения вклад loss на пиксель умножается на дополнение к Sky Masking — пиксели неба вносят 0 в градиент, так что Gaussians, проецирующиеся в небо, не получают сигналов оптимизации и поэтому не «уплотняются» и не «ярчают». Существенно снижает floaters (тёмные сгустки в небе) в уличных/дроновых сценах. Стоит ~3 % регрессии L1 в классической 40K-тренировке (см. [memory/dev\\_outdoor-floater-reduction.md](#)). Полезно только для уличных сцен с явно распознаваемым небом; для комнатных сцен или белого фона сегментация неба определяет неправильные области и блокирует валидные сигналы loss. Значение теперь хранится для каждого проекта (больше не глобально для приложения) и следует за предустановкой / файлом сцены.

**ПРОСТЫМИ СЛОВАМИ**

На уличных снимках с небом в кадре в небе часто образуются чёрные или цветные сгустки — так называемые «floaters». Эта опция автоматически определяет, где находится небо, и говорит обучению: «Оставь небо в покое.» Очень хорошо работает для полётов дронов и пейзажных сцен. Для помещений или тёмного фона может ухудшать изображение — поэтому включайте, только когда видно настоящее небо. Подробности: [memory/dev\\_outdoor-floater-reduction.md](#).

### I43 Очистка floater в середине обучения



ГДЕ

Инспектор → Enhancements (группа Outdoor) → переключатель «Mid-Training Floater Cleanup». Bound: `AppState.trainingConfig.floaterCleanupEnabled` (для каждого проекта, `@DefaultFalse`). По умолчанию: `false`.

#### ТЕХНИЧЕСКИ

Включает два дополнительных прохода `density-control` во время Classic 40K-тренировки (предустановка «P4 Quality»): на итерации 20 000 и 30 000. Оба прохода сканируют все Gaussians по трём критериям: (а) очень низкая непрозрачность (по умолчанию 0.005), (б) крошечный `screen-space`-размер, (с) нет вкладов в `loss` за последние 1000 итераций. Gaussians, отвечающие всем трём условиям, удаляются. Эффект: ~5–15 % меньше Gaussians в конце обучения, заметно меньше тёмных сгустков в небе для дроновых/уличных сцен. Стоит ~1–3 % регрессии L1 на ближних комнатных сценах, поэтому не включено по умолчанию. Две итерации очистки (20К, 30К) зашиты жёстко и в настоящее время не могут быть изменены через UI; для более коротких тренировок (напр. P2 Preview 5K) переключатель не имеет эффекта, потому что никогда не достигает отметок итераций. **Новое:** переключатель доступен только когда активная предустановка использует **Classic-densifier** (`densificationStrategy == .classic`). При MCMC или Hybrid он **disabled**, и появляется inline-подсказка, потому что эти стратегии обрабатывают мёртвые Gaussians сами (MCMC через `relocation`, Hybrid через комбинированную `reloc-/noise-логику`) — ручные проходы очистки были бы там бесполезны или контрпродуктивны. Ссылка на код: `RadianceKitApp.swift`, General-вкладка. Подробности: `memory/dev_outdoor-floater-reduction.md`.

#### ПРОСТЫМИ СЛОВАМИ

Во время обучения иногда образуются «мёртвые» точки Gaussian, которые больше не вносят вклад в качество изображения, но занимают память. Эта опция дважды чистит длинное обучение (на 20К и 30К итераций) и удаляет эти трупы. Для уличных сцен с небом это особенно полезно, потому что там собирается большинство floaters. Для маленьких тренировок или снимков мебели крупным планом особо не нужно. Переключатель можно включить только когда ваша предустановка использует Classic-densifier — при MCMC или Hybrid-предустановках он серый (с коротким пояснением), потому что они убирают свои мёртвые точки сами.


**I44 Реконструировать купол неба**

ГДЕ

Инспектор → Enhancements (группа Outdoor)  
→ переключатель «Reconstruct Sky Dome».  
Bound: AppState.trainingConfig.skyDomeEnabled  
(для каждого проекта, @DefaultFalse ). По умолчанию: false .

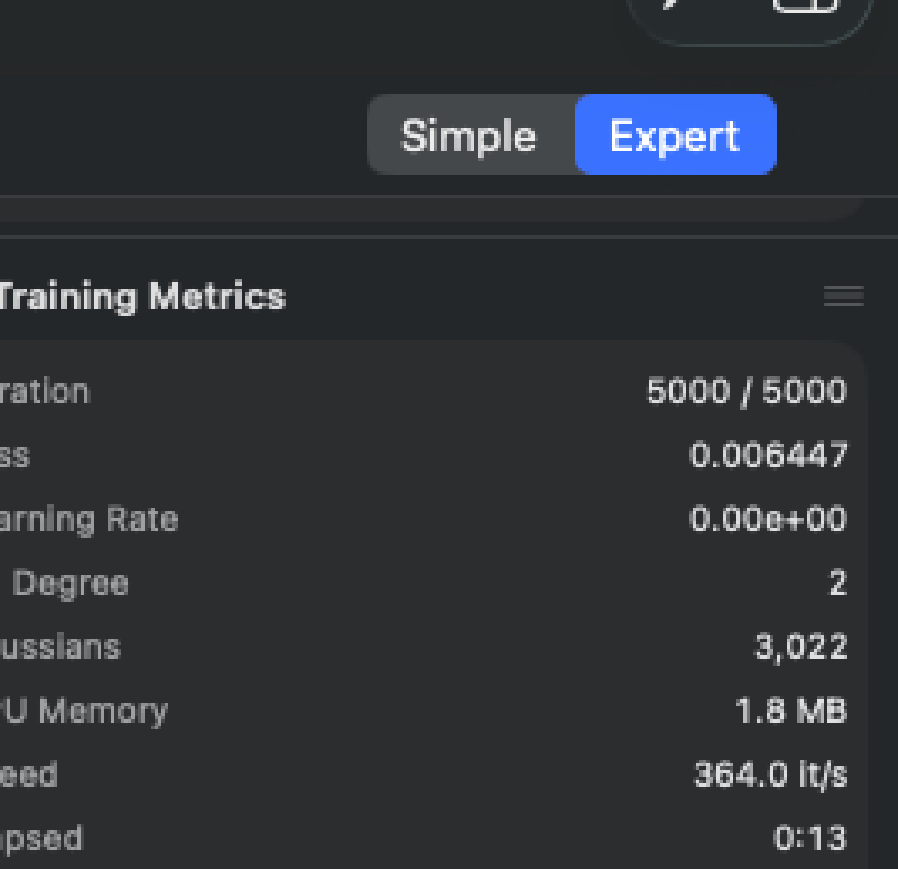
 **ТЕХНИЧЕСКИ**

Активирует pre-training проекцию Sky Dome (V549e MVP). После SfM и перед запуском обучения для каждой входной камеры извлекается shared с S7 маска неба Apple Vision из изображения, пиксели неба up-проецируются с помощью intrinsics камеры на виртуальную сферическую поверхность (радиус по умолчанию — 8x радиус сцены). На этой сфере инициализируется ~5000 новых Gaussians с цветовыми средними из спроецированных пикселей неба, очень большим масштабом (1.0 в единицах сцены) и начальной непрозрачностью 0.95. Эти 5000 Gaussians не являются маской неба в классическом смысле — они тренируются как все остальные, но высокая начальная непрозрачность удерживает их в тонкой оболочке. Результат: для 360°-novel-views в уличных/дроновых сценах появляются реальные цвет неба и облачные структуры вместо тёмных сгустков конфетти. Значение запоминается между перезапусками. Полезно только для уличных сцен с покрытием камер не менее 360°; для чисто объектных съёмки без обзора неба не имеет эффекта. Статус: экспериментальное, более широкая A/B-валидация на большем количестве уличных наборов ещё предстоит.

 **ПРОСТЫМИ СЛОВАМИ**

Вместо того чтобы обучение пыталось «угадать» небо по немногим видимым пикселям (что приводит к floaters), RadianceKit проецирует пиксели неба напрямую на виртуальную сферу вокруг сцены ещё до запуска обучения. Когда вы потом крутите готовую сцену в 360°, видно настоящее небо вместо чёрных сгустков. Работает только для уличных съёмки, где небо действительно попадает в кадр. Для сканов гостиной или студийных съёмки не даёт ничего.

## Секция Metrics (I30–I38)



The screenshot shows the 'Expert' view of the RadianceKit interface. At the top, there are two buttons: 'Simple' (disabled) and 'Expert' (active). Below this is a section titled 'Training Metrics' with a hamburger menu icon on the right. The metrics are displayed in a table format:

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

Рисунок 13. Вырезка только секции Training Metrics после завершённого обучения на Bouquet (5K итераций, финально 2 991 Gaussians) — таблица с метриками обучения (Iteration, Loss, SSIM Loss, Combined Loss, Gaussian Count, Learning Rate, Elapsed, ETA)

Пока идёт обучение, секция Metrics показывает девять живых значений из цикла обучения. До начала обучения секция пуста («Start training to see live metrics»). Все значения обновляются каждые ~30 итераций (частота обновления потока). Секция read-only — ни один элемент не кликабелен и не изменяем. Для более глубокого анализа смотрите JSONL- training-логи под `~/Documents/RadianceKit/Logs/` (скрипт `python3 scripts/analyze_logs.py best 5`).

### I30 Iteration



ГДЕ

Инспектор → Metrics → Iteration. Read-only.

#### ТЕХНИЧЕСКИ

Отображается в формате «4523 / 40000» — текущая итерация над общим планируемым числом итераций. Считает синхронно с циклом обучения, который отправляет значения каждые ~30 iter. Второе число соответствует значению Max Iterations на момент старта; оно больше не меняется, даже если пользователь подкручивает степпер позднее — идущий прогон использует собственную snapshot-копию. Если приложение через меню Training (Continue Training +5K/+10K/+20K) добавляет дополнительные итерации, знаменатель увеличивается.

#### ПРОСТЫМИ СЛОВАМИ

Где сейчас находится обучение. «4523 / 40000» означает: 4523 из 40 000 шагов пройдено, т.е. около 11 % готово. Левое число идёт в реальном времени; если оно зависает на минуты, обучение застряло — обычно намёк на GPU-throttling или конкурирующее приложение. Правое число соответствует значению Max Iterations (I18) на старте обучения и больше не меняется, даже если потом подкрутите степпер. С Continue Training (+5K/+10K/+20K) оно растёт на дополнительные шаги.

### I31 Loss



ГДЕ

Инспектор → Metrics → Loss. Read-only.

#### ТЕХНИЧЕСКИ

Float-значение с шестью знаками после запятой (напр. «0.024385»). Измеряет комбинированный L1+SSIM loss (смесь управляется через I21 SSIM Weight) плюс опциональный Perceptual Loss (I29) и другие регуляризаторы. Шкала не абсолютная, а зависит от сцены — для большинства сравнений требуется тот же датасет. Типичные конечные значения при хороших конфигурациях: - 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 (хуже из-за геометрии) - Indoor apartments: 0.018–0.025

Значения выше 0.10 после 5K итераций указывают на SfM- проблемы (плохие позы камер) — перервите и пересчитайте SfM.

#### ПРОСТЫМИ СЛОВАМИ

Насколько отрендеренное изображение всё ещё отклоняется от оригинала — комбинированно из L1, SSIM и, возможно, Perceptual Loss. Меньше — лучше. Ниже 0.03 — обычно действительно хорошо, ниже 0.05 — ещё нормально, уличные сцены лежат по геометрическим причинам скорее на 0.03–0.06. Выше 0.10 после нескольких тысяч итераций — сигнал тревоги, обычно дело в реконструкции камер (SfM не сработал чисто). Шкала не абсолютна, а зависит от сцены; сравнения — только внутри одного датасета. Если число вдруг скакнуло вверх, обычно произошло событие gradient explosion.

### I32 Learning Rate



ГДЕ

Инспектор → Metrics → Learning Rate. Read-only.

#### ТЕХНИЧЕСКИ

Отображение в научной нотации (напр. «1.60e-04»). Текущая скорость обучения для параметров позиции (у 3DGS шесть независимых LR для позиции, SH-DC, SH-Rest, непрозрачности, масштаба, поворота — здесь отображается позиция-LR как представительное значение). Начальное значение по умолчанию — 1.6e-4, которое спадает через exponential decay до ~1.6e-6 к концу обучения. Decay настраивается через поле LR schedule в training configuration (Т-поле в главе 6). Если LR остаётся необычно высоким (напр. 1e-3 или более после 10К итераций), это может указывать на неверно загруженную конфигурацию.

#### ПРОСТЫМИ СЛОВАМИ

Насколько большие сейчас шаги оптимизации — конкретно скорость обучения для позиций Gaussian. Стартует на 1.60e-04 и экспоненциально опускается к концу обучения до около 1.60e-06 («1.60e-06» = 0.0000016). Кривая идёт автоматически, тут ничего подкручивать не надо. Если значение после 10 000+ итераций всё ещё больше 1e-3, вероятно, загружена неверная конфигурация — прервите обучение и выберите новую предустановку. Внутри у 3DGS шесть независимых learning rate (позиция, SH-DC, SH-Rest, непрозрачность, масштаб, поворот); здесь вы видите только позицию-LR как представителя.

### I33 SH Degree



ГДЕ

Инспектор → Metrics → SH Degree. Read-only.

#### ТЕХНИЧЕСКИ

Integer 0–3. Степень spherical harmonics для цветового представления. Стартует на 0 (только DC-компонента, т.е. независимый от направления цвет на Gaussian — значит только одна RGB-константа) и прогрессивно растёт до 3 по ходу обучения. Стандартное расписание поднимает степень на итерациях 1000/2000/3000 на 1 каждый раз. SH-3 соответствует 48 цветовым коэффициентам на Gaussian (3 RGB-канала × 16 SH-базисных функций). Более высокая SH-степень = более зависимое от направления отражение (глянцевые поверхности выглядят корректно по-разному под разными углами обзора), но также больше памяти и более медленное обучение.

#### ПРОСТЫМИ СЛОВАМИ

Насколько сложно сейчас цветное представление на Gaussian. Стартует на 0 (только независимый от направления цвет на точку) и пошагово поднимается до 3 — типично на итерациях 1000, 2000 и 3000. Стадия 3 означает 48 цветовых коэффициентов на Gaussian и позволяет зависящие от направления отражения, т.е. что глянцевые поверхности выглядят корректно по-разному с разных углов обзора. Активно трогать не надо, расписание идёт автоматически. Более высокая степень стоит больше памяти и немного замедляет обучение — но это цена за реалистичные блики.

## I34 Gaussians



ГДЕ

Инспектор → Metrics → Gaussians. Read-only.

## ТЕХНИЧЕСКИ

Текущее число Gaussians в модели, отформатировано с locale-разделителем (напр. «524.318»). Рост: - Classic: стартует на SfM init-точках (обычно 50K–300K), растёт через clone/split до незадолго до Densify Until, затем статично до конца обучения (по модулю pruning) - MCMC: sample-точки добавляются до MCMC cap, затем только relocation

Здоровые конечные значения: - Classic Quality: 400K–700K (Horse 524K, Garden 800K) - MCMC Full: точно на cap (по умолчанию 150K, с auto-scale множителем × SfM count, в зависимости от сцены 500K–1.5M)

При MCMC, если число падает до < 60 % cap → аномалия (collapse-индикатор, указывает на слишком агрессивные регуляризаторы).

## ПРОСТЫМИ СЛОВАМИ

Сколько точек Gaussian сейчас имеет 3D-модель. Растёт во время обучения, пока не достигнут Densify Until (I20); после этого число остаётся практически постоянным. Больше точек = больше детализации, но также больший файл и более медленный рендеринг в viewport. 500 000 Gaussians — типичное среднее для Classic Quality на средней сцене; MCMC Full попадает в зависимости от auto-scale (I17) в 500K–1.5M. Если число при MCMC вдруг падает ниже 60 % cap, это индикатор коллапса — обычно слишком агрессивные регуляризаторы.

## I35 GPU Memory



ГДЕ

Инспектор → Metrics → GPU Memory. Read-only.

## ТЕХНИЧЕСКИ

Оценка потребления памяти Gaussian-буфера как число Gaussians × 616 байт (отформатировано в memory-стиле). 616 байт — эмпирический размер полностью оснащённого Gaussian (позиция, масштаб, поворот, непрозрачность, SH-коэффициенты степени 3, gradient accumulator). Отображение **не** учитывает overhead рендера (tile buffer, sort buffer, backward buffer) — реальная потребность в GPU-памяти обычно в 2–3 раза выше этого значения. При 500K Gaussians: отображается ~290 MB, реально ~700 MB. При 1.5M Gaussians: отображается ~880 MB, реально ~2.5 GB. На M3 Max с 64+ GB Unified Memory не критично, на M3 Pro с 18 GB уже предел.

## ПРОСТЫМИ СЛОВАМИ

Оценка того, сколько GPU-памяти занимают сами Gaussians — около 616 байт на точку. Фактическое потребление GPU в 2–3 раза выше отображаемого, поскольку рендерер добавляет свои собственные tile-, sort- и backward-буферы. На MacBook с 16–18 GB unified memory лучше оставаться ниже 500K Gaussians; с M3 Max или Studio (64+ GB) можно спокойно крутить 1.5M и больше. Если обучение вдруг падает или система свопит, обычно достигнут именно этот предел — уменьшите Render Scale (I22) или Densify Until (I20).

**I36 Speed**

ГДЕ

Инспектор → Metrics → Speed. Read-only.

**ТЕХНИЧЕСКИ**

Итерации в секунду с одним знаком после запятой («24.3 it/s»). Вычисляется тренером как скользящее среднее по последним ~100 итерациям. Типичные значения: - предустановка Quick (1K iter): 80–120 it/s (коротко, без устойчивого состояния) - Classic 20K @ 1.0 Render Scale (сцена Truck, M3 Max): 25–35 it/s - Classic 20K @ 0.5 Render Scale: 80–120 it/s - МСМС 200K @ 0.5 Render Scale: 25–50 it/s (медленнее из-за relocation) - При 1M+ Gaussians и полном разрешении: < 10 it/s

Уменьшение скорости по ходу обучения нормально — больше Gaussians = больше вычислений на итерацию. Внезапные провалы (напр. с 30 → 5 it/s) указывают на GPU thermal throttling или конкурирующие приложения.

**ПРОСТЫМИ СЛОВАМИ**

Насколько быстро идёт обучение, в итерациях в секунду. Обычно стоит на 20–50 it/s, с уменьшенным Render Scale (50 %) и маленькими сценами — и 80–120 it/s. Опускается по ходу обучения вполне естественно, поскольку больше Gaussians = больше работы на итерацию. Внезапные провалы (напр. 30 → 5 it/s) указывают на GPU thermal throttling или конкурирующие приложения — вкладки браузера с видео, Time Machine backup, индексацию Photos. Удержание приложения на переднем плане и закрытие фоновых программ часто помогает. При 1M+ Gaussians и полном разрешении ниже 10 it/s — норма.

**I37 Elapsed**

ГДЕ

Инспектор → Metrics → Elapsed. Read-only.

**ТЕХНИЧЕСКИ**

Уже прошедшее время как «4:23» (m:ss) или «1:23:45» (h:mm:ss). Смена формата от 1 часа. Измеряет только чистое время обучения, не предшествующие фазы (вычисление SfM, импорт изображений). С pause/resume часы продолжают идти — так что это wall-clock, не CPU time.

**ПРОСТЫМИ СЛОВАМИ**


Сколько времени уже работает обучение, как чисто секундомер (wall-clock time). Формат — «m:ss» до одного часа, затем «h:mm:ss». Не «CPU time», а «как долго мы уже ждём» — так что время пауз тоже считается. Измеряет только чистую фазу обучения, не предшествующее вычисление SfM или импорт изображений. Полезно для сравнения с ETA (I38) — если Elapsed явно превышает изначальный ETA, обучение где-то стало медленнее запланированного.

**I38** ETA

Инспектор → Metrics → ETA. Read-only.

 ТЕХНИЧЕСКИ

Оценочное оставшееся время как «17:42» или «1:12:35». Расчёт: (Max Iterations – текущая итерация) / итерации в секунду. Показывает «–», когда скорость сейчас нулевая (в самом начале или во время паузы). Оценка **не** подстраивается под типичное замедление к концу обучения — особенно при MCMC и Classic с большими значениями Densify-Until обучение склонно становиться медленнее, потому что появляется всё больше Gaussians. Реально обычно остаётся на 10–20 % выше изначального ETA.

 ПРОСТЫМИ СЛОВАМИ

Сколько времени ожидаемо осталось — считается из оставшихся итераций и текущей скорости (I36). Грубая оценка: если Mac вдруг становится медленнее (больше Gaussians из densify-фазы, thermal throttling, другие приложения), может длиться дольше отображённого. Приложение не учитывает типичное замедление к концу обучения, так что реальный конец обычно попадает на 10–20 % выше изначального ETA. Плюс 15 %, и обычно сходится. Показывает «–», когда скорость сейчас 0 (старт обучения или пауза).

## Секция Loss Chart (I39–I41)

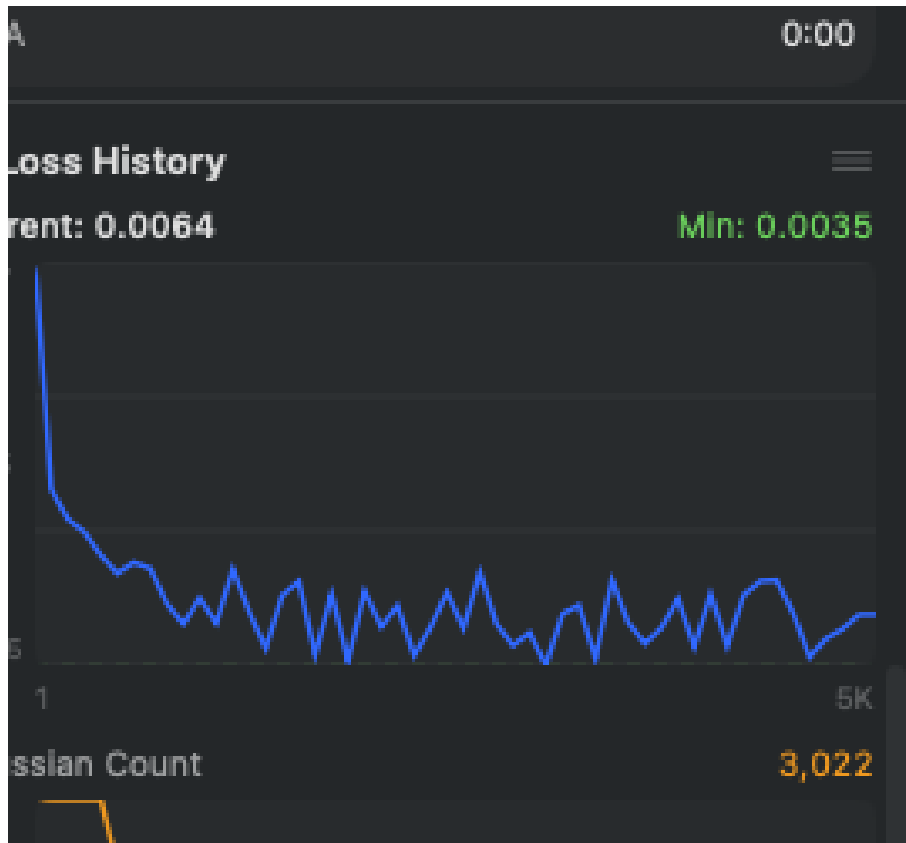


Рисунок 14. Вырезка только секции Loss History после завершённого обучения — Current 0.0064, Min 0.0035 (зелёный), синяя кривая от 0.027 (итерация 1) к 0.0035 (итерация 5K) с характерным изгибом около iter 200, ниже график Gaussian Count оранжевым

Секция Loss Chart визуализирует ход обучения со временем. Состоит из двух графиков: Loss Curve (большой, сверху, синий) и Gaussian Count (меньше, снизу, оранжевый). Оба строятся вживую во время обучения и сохраняются до следующего старта обучения. До первого обучения область пуста («Loss curve will appear during training»). Графики — чистые SwiftUI path-drawings (не Swift Charts framework), поэтому рендерятся плавно даже при 100K+ точек.

### I39 Current Loss (отображение)



ГДЕ

Инспектор → Loss Chart → левая label-область «Current: 0.0287». Read-only.

#### ТЕХНИЧЕСКИ

Float-значение последней точки loss-выборки, форматировано с четырьмя знаками после запятой. Идентично с I31 (Loss в секции Metrics), здесь только компактнее отформатировано. Источник — loss-history — список, получающий запись каждые ~30 итераций. В список добавляются только конечные значения — NaN/Infinity (очень редко, в случае gradient explosion bug) фильтруются.

#### ПРОСТЫМИ СЛОВАМИ

Текущее значение loss в более короткой нотации, чем в секции Metrics (четыре знака после запятой). Содержательно идентично с I31, но здесь отображение сидит прямо у loss-графика и даёт точное числовое значение при наблюдении за кривой. Как все живые метрики, обновляется каждые 30 итераций. NaN или Infinity (крайне редко при gradient-explosion-багах) приложение автоматически отфильтровывает. Полезно при взгляде на график, чтобы не прыгать в другую секцию.

### I40 Min Loss (отображение)



ГДЕ

Инспектор → Loss Chart → правая label-область «Min: 0.0245» (зелёный). Read-only.

#### ТЕХНИЧЕСКИ

Минимум всех когда-либо виденных значений loss текущего прогона обучения. Живо пересчитывается из loss-history — никакого отдельного сохранения. Отображается зелёным текстом, потому что «Min» = «лучший на данный момент». Пунктирная зелёная линия на нижнем краю графика визуально маркирует эту Y-позицию. С continue-training-сессиями отслеживание min рестартует — старая история заменяется в UI новой (не дописывается). Если текущее обучение идёт хуже предыдущего, отображение min может оказаться больше предыдущего конечного результата.

#### ПРОСТЫМИ СЛОВАМИ

Самое низкое значение loss, которое это обучение до сих пор видело — показано зелёным, потому что «лучший на данный момент». Пунктирная зелёная линия на нижнем краю графика визуально маркирует эту позицию. Если текущая кривая сейчас лежит явно выше неё, надеемся, ещё будет улучшение; обычно Min — индикатор конечного результата, который вас потом и интересует. С continue-training-сессиями отслеживание min рестартует, потому что старая история заменяется в UI новой — значение min может поэтому выглядеть хуже предыдущего конечного результата.

## I41 График Gaussian Count



ГДЕ

Инспектор → Loss Chart → второй график снизу (оранжевый). Read-only.



ТЕХНИЧЕСКИ

Линейный график числа Gaussians по итерациям обучения. Источник: история числа Gaussians (список пар (iter, count), заполняемый тренером каждые ~30 iter). Y- шкала динамическая между минимумом и максимумом истории. При стратегии Classic кривая обычно выглядит так: стабильно растёт до Densify Until, потом плоская (с маленькими pruning-колебаниями). При MCMC: крутой подъём до sар, затем горизонтальная линия (relocation удерживает число постоянным). Если кривая **падает** несмотря на активное обучение, densification прунит слишком агрессивно — индикация неверных дефолтов или известного MCMC-collapse- бага (тема hotfix v1.4.4).

### ПРОСТЫМИ СЛОВАМИ

Как развивается число Gaussians по времени обучения — меньший оранжевый график под кривой Loss. При стратегии Classic линия стабильно растёт, пока не достигнут Densify Until (I20), после этого остаётся плоской с маленькими pruning-колебаниями. При MCMC стрелой выстреливает до sар, затем остаётся горизонтальной, потому что relocation удерживает число постоянным. Если кривая несмотря на активное обучение вдруг ломается вниз, densification слишком агрессивен с pruning — классический признак MCMC-collapse-бага из v1.4.4. Тогда помогает обновление приложения или возврат к Classic.

### Как читать кривую loss?

График loss — важнейший диагностический инструмент в Инспекторе — ни один другой индикатор не показывает так напрямую, продвигается ли обучение полезно или застряло. Типичная здоровая форма — быстрый спад в первые 1000–3000 итераций (от ~0.15 до ~0.05), за которым следует медленный, устойчивый спад до конца обучения (до 0.020–0.030). Логарифмически кривая выглядит как гладкая диагональ.

**Что означает плато в loss?** Если кривая остаётся плоской несколько тысяч итераций, есть два возможных прочтения: (а) обучение «сошлось» — loss больше не может существенно падать, потому что модель так хороша, как может быть с данными данными и настройками. Это желаемое; это «готово». (b) Обучение «висит» — loss мог бы ещё падать, но оптимизация застаивается (локальный минимум, learning rate слишком мал, densification выключен). Различение: если значение loss лежит в типично хорошем диапазоне (0.020–0.030 при indoor/object, 0.040–0.060 при outdoor), и кривая плоская с 5K итераций, оно сошлось. Если значение явно выше, чем у сравнимых сцен (напр. 0.08), оно висит.

**Примечание: плато Gaussians ≠ плато Loss.** Плато по числу Gaussians **не** означает «обучение готово». Это только означает, что densification перестал добавлять новые точки — либо потому что достигнут (Classic), либо потому что MCMC сар полон. Обучение продолжается после этого и только уточняет существующие точки. Фактический сигнал «готово» вы читаете по кривой loss и по отображению Iteration (I30), не здесь.

**Правило большого пальца для отмены:** Если кривая loss после 5000+ итераций лежит выше 0.08 и едва ли снижается, с высокой вероятностью SfM-реконструкция перекошена. Прервите обучение, загляните в главу 9, подходит ли выбранный SfM-бэкенд к сцене, при необходимости переключитесь на COLMAP/Native, потом перезапустите. Лучше вложить 10 минут в лучший SfM, чем 2 часа обучения с плохим выравниванием камер.

## Когда обращаться к Инспектору?

Быстрая справка: какая секция + какие элементы управления для какого типичного варианта использования?

Распространённая задача	Секция	ID элементов
Обесцветить цвета готового Splat	Look	L1 (Saturation)
Округлить игольчатые/конфетти Splat-ы	Look	L2 (Splat length)
Заполнить дырявое облако / увеличить Splat-ы	Look	L3 (Splat size)
Притушить дальние «конфетти» на orbit-съёмках	Look	L4 (Fade far region)
Отбросить коррекции Look	Look	L5 (Reset finishing)
Загрузить готовую настройку	Presets	I7 (клик по строке)
Сохранить свою настройку	Presets	I1 → I2 → I4
Поделиться настройкой с коллегой	Presets	I5 (Export) или I6 (Import)
Сменить SfM-бэкенд (напр., потому что Apple-PG слишком нестабилен)	Training Configuration	I12 (см. гл. 9)
Обработать видеокadres без EXIF focal length	Training Configuration	I13 (FOV Override)
COLMAP-производительность: GLOMAP вместо Classic	Training Configuration	I14
Переключить с Classic на MCMC	Training Configuration	I15
Дать обучению дольше работать	Training Configuration	I18 (Max Iter) + I20 (Densify Until) — связаны через I19
Уполовинить GPU-время	Training Configuration	I22 (Render Scale на 50 %)
Качество обучения +6 % (MCMC)	Training Configuration	I16 (MCMC Quality)
Уличная сцена с большим числом SfM-точек	Training Configuration	I17 (Auto-scale by scene)
Настроить / сменить путь COLMAP	Training Configuration	I23 / I24 / I25
Сделать экспорт-файлы меньше	Enhancements	I26 (всегда оставлять включённым)
Более резкий viewport без увеличения времени обучения	Enhancements	I27 (Viewport Scaling → MetalFX)
MetalFX слишком сильно влияет на FPS при визуализации с тонкими структурами	Enhancements MetalFX Chart	I27 (Viewport Scaling → MetalFX) I28 (MetalFX Quality) I29 (MetalFX Scale) I30 (MetalFX Filter) I31 (MetalFX Blur) I32 (MetalFX Anti-aliasing) I33 (MetalFX Depth of Field) I34 (MetalFX Motion Blur) I35 (MetalFX Bloom) I36 (MetalFX Lens Flare) I37 (MetalFX Vignette) I38 (MetalFX Color Grading) I39 (MetalFX Contrast) I40 (MetalFX Saturation) I41 (MetalFX Brightness) I42 (MetalFX Gamma) I43 (MetalFX White Balance) I44 (MetalFX Color Temperature) I45 (MetalFX Color Balance) I46 (MetalFX Color Contrast) I47 (MetalFX Color Contrast) I48 (MetalFX Color Contrast) I49 (MetalFX Color Contrast) I50 (MetalFX Color Contrast)

## ГЛАВА

## Глава 3 — Настройки

---

Окно настроек открывается через RadianceKit → Settings... или стандартное клавиатурное сокращение **⌘, .** Оно содержит две вкладки: **General** и **AI Helpers**. В отличие от значений Инспектора из главы 2, настройки в этом окне действуют **глобально для приложения** (для всех проектов) — они сохраняются и переживают перезапуск приложения. Вкладка General группирует три содержательные секции: Interface, Viewport и Training. (Три переключателя Outdoor-Floater — Sky Masking, Mid-Training Floater Cleanup, Reconstruct Sky Dome — которые раньше жили здесь, начиная с v1.6 переехали в секцию Enhancements Expert-Инспектора, где теперь хранятся для каждого проекта; см. главу 2, 142–144.) Вкладка AI Helpers включает on-device ML-помощников (Vision, CoreML) для предобработки SfM и тренировки.

Прежних элементов управления для совместного включения или выключения всех AI Helpers в текущей версии больше нет — они здесь не документируются. Прежняя область «Coming Soon» для ещё не поставляемых помощников также удалена и здесь не упоминается.

## Вкладка General

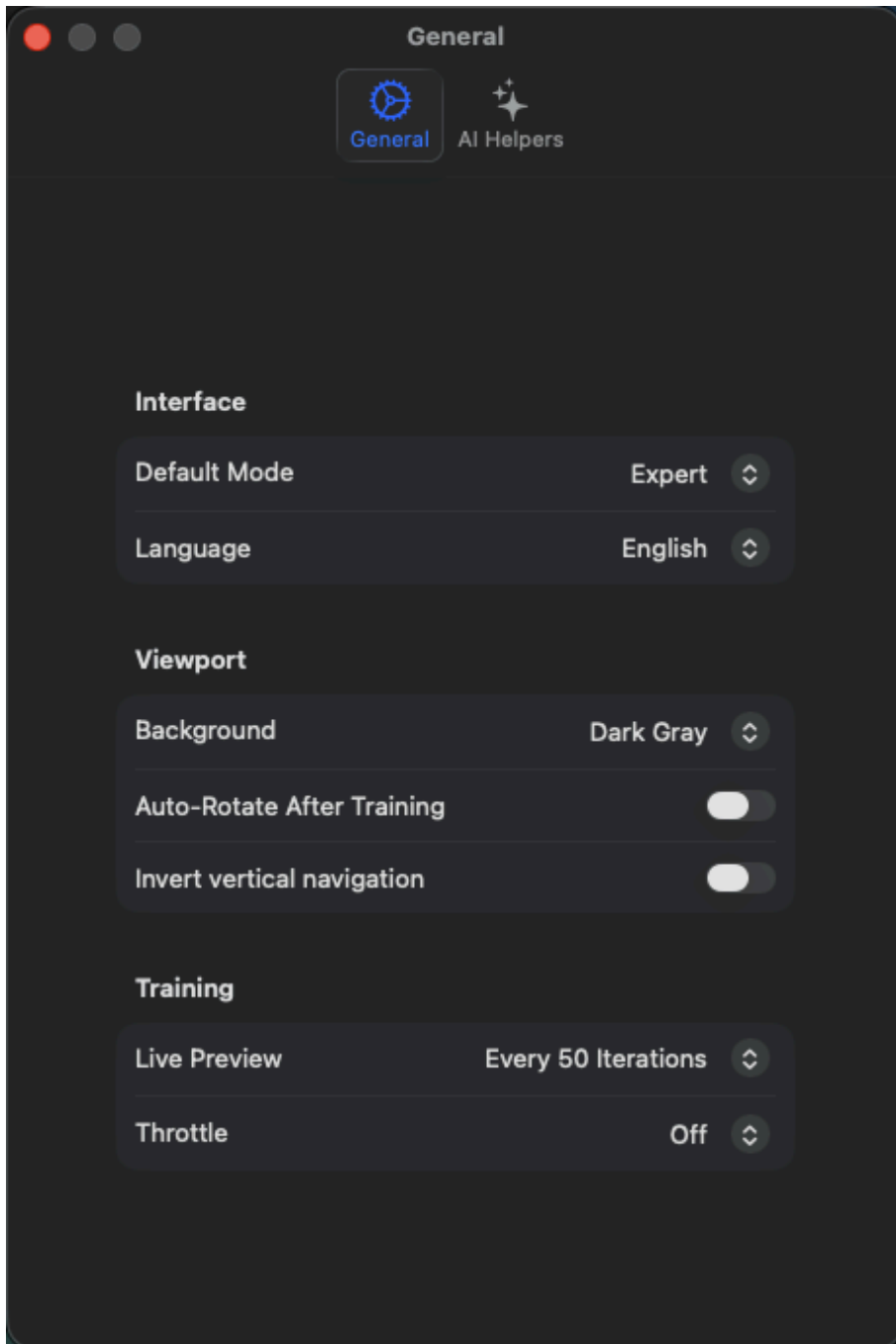


Рисунок 15. Settings → вкладка General с секциями Interface, Viewport, Training и Experimental

## S1 Режим по умолчанию



ГДЕ

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



ТЕХНИЧЕСКИ

Управляет тем, в каком из двух режимов UI приложение открывается при следующем запуске. «Simple Mode» — это направляемый рабочий процесс мастера из 4 шагов (Import → Processing → Preview → Export, документирован в главе 10 под Z1–Z4), «Expert Mode» — классическая трёхпанельная раскладка с Navigator, 3D Viewport и Expert Inspector из главы 2. Значение запоминается между перезапусками. Эффект идентичен меню Mode → Simple Mode (⌘1) / Mode → Expert Mode (⌘2), за исключением того, что меню переключает текущую сессию, а этот picker задаёт значение по умолчанию для будущих сессий. Оба режима обращаются к одному и тому же состоянию проекта — проекты, камеры и конфигурация обучения сохраняются при переключении режимов. Специфичные для режима кнопки панели инструментов перерендериваются немедленно.

### ПРОСТЫМИ СЛОВАМИ

Здесь вы выбираете, с каким интерфейсом RadianceKit запустится при следующем старте. «Simple Mode» — режим для новичков: четыре чётких шага, готовые предустановки, почти нет опций. «Expert Mode» — полная раскладка с инструментами и всеми элементами управления, которые вы видите в главе 2. Можно в любой момент переключаться через меню «Mode», не теряя изображений или прогресса тренировок.

## S2 Язык



ГДЕ

Settings → General → Interface → Language picker. Bound:.. По умолчанию: `.system` (следует языку macOS).



ТЕХНИЧЕСКИ

Выбирает язык отображения всего UI приложения независимо от системного языка macOS. RadianceKit локализован на 17 языков (`de`, `en`, `pl`, `en-AU`, `ar-SA`, плюс 12 других). При значении «System» приложение следует языку macOS. При явном выборе языковая настройка запоминается между перезапусками; полный эффект обычно требует перезапуска приложения, поскольку бандлы локализации загружаются только при старте. Все 298 задокументированных ключей локализации в проекте учитываются, включая все тексты в подвидах и подсказках помощи.

### ПРОСТЫМИ СЛОВАМИ

Если ваш Mac работает на английском, но вы предпочитаете немецкий интерфейс RadianceKit (или наоборот), задайте это здесь. Большинство текстов меняется немедленно. Некоторые диалоги появляются на новом языке только после перезапуска приложения.

## S3 Фон Viewport



ГДЕ

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



ТЕХНИЧЕСКИ

Задаёт цвет фона по умолчанию для 3D Viewport. Три варианта: «Dark Gray» (RGB 0.1, 0.1, 0.1 — по умолчанию), «Black» (0, 0, 0) и «White» (1, 1, 1). Настройка фиксирует значение по умолчанию для новых проектов и сессий между перезапусками и одновременно немедленно обновляет работающий рендерер Metal. Идентичные опции находятся в меню Viewport → Background (M21, M22, M23), но picker настроен задаёт значение по умолчанию, а меню переключает текущее отображение. Важно для скриншотов и демо-видео: белый фон сильнее выделяет зелёные/синие floaters, тёмный фон лучше для чистых render-захватов.

### ПРОСТЫМИ СЛОВАМИ

Цвет за вашими 3D-моделями в окне предпросмотра. Тёмно-серый — по умолчанию и подходит большинству сцен. Белый хорош для скриншотов, чёрный смотрится изящнее на gender-захватах. Цвет для текущей сцены можно в любой момент переключить через меню «Viewport → Background» — эта настройка только определяет, какой цвет активен снова при следующем открытии.

## S4 Авто-вращение после обучения

### ГДЕ

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

### ТЕХНИЧЕСКИ

Запускает непрерывное вращение камеры Viewport вокруг центра сцены сразу после окончания обучения (скорость вращения по умолчанию  $\sim 0.3$  rad/s). Практически полезно для демо-сессий, А/В-сравнений и для немедленной оценки с обзора 360°, образовались ли «floaters» на краю сцены. Эффект визуально идентичен меню Viewport → Toggle Auto-Rotation (M16, ⌘⌥T), за исключением того, что этот переключатель срабатывает автоматически после окончания обучения, а не вручную. Может быть прерван в любой момент через меню или щелчком в Viewport (пауза вращения). Не влияет на производительность обучения — вращение запускается только после его окончания.

### ПРОСТЫМИ СЛОВАМИ

Когда включено, 3D-сцена начинает автоматически вращаться сразу после завершения обучения — как карусель. Удобно, когда обучаете на ночь и хотите утром сразу увидеть результат в движении, не щёлкая мышкой. Для длинных сессий, где вы только наблюдаете за обучением, оставьте выключенным.

## S5 Интервал живого предпросмотра



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

### ТЕХНИЧЕСКИ

Определяет интервал итераций, на котором текущий снимок обучения рендерится в 3D Viewport. Четыре дискретных значения: 0 («Off»), 50, 250, 1000 итераций. С активным Live Preview тренер копирует буфер Gaussian с GPU в отдельный render-буфер и триггерит перерисовку Viewport. С «Off» Viewport обновляется только после завершения обучения. Цена производительности: каждые 50 итераций — на ~5–10 % медленнее на M3 Ultra, каждые 250 итераций — ~1–2 % медленнее, каждые 1000 итераций — неизмеримо. Накладные расходы памяти — постоянные ~2 ГБ для snapshot-буфера, независимо от интервала. Значение служит как значение по умолчанию для новых тренировок; после запуска обучения Training Inspector показывает фактическое живое значение запущенной тренировки. На интервале 50 визуальное впечатление — плавный «рост» облака точек, на 1000 выглядит рывками.

### ПРОСТЫМИ СЛОВАМИ

Пока идёт обучение, можно выбрать, как часто обновляется 3D-вид. «Off» означает: без обновления во время обучения (быстрее всего). «Каждые 50 итераций» показывает почти в реальном времени, как складывается ваша сцена (чуть медленнее). Для комфортного наблюдения за небольшим обучением «каждые 250» — хороший компромисс.

## S6 Задержка throttle



ГДЕ

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

### ТЕХНИЧЕСКИ

Вставляет искусственную задержку в миллисекундах между итерациями обучения. Четыре дискретных значения: 0 («Off»), 2 («Light»), 5 («Moderate»), 10 («Eco»). Назначение: при более долгих тренировках (несколько часов) GPU иначе был бы загружен на 100 %, что приводит к заметно более медленному системному UI (курсор мыши заедает, другие приложения становятся вялыми). Задержка throttle даёт GPU паузы, в которые могут выполняться другие задачи. Цена производительности значительна: при throttle 5 мс типичная 40К-тренировка длится примерно на 50–80 % дольше, чем без throttle. В режиме производительности «Eco» (10 мс) задержка на итерацию больше самой итерации — в 2–3 раза медленнее. При активном throttle под picker появляется примечание: «Throttle is on. Training will be slower than usual.» Само приложение не реагирует заметно лучше — выигрывают только другие приложения.

### ПРОСТЫМИ СЛОВАМИ

Если во время длительного обучения Mac слишком сильно греется или другие программы становятся слишком медленными, включите здесь тормоз. «Off» даёт GPU полный газ (быстрее всего). «Light» делает маленькую паузу между каждым шагом (чуть медленнее, но система реагирует лучше). «Eco» — самый сильный тормоз — хорошо для ночных тренировок на MacBook, который не должен слишком греться.

## Вкладка AI Helpers

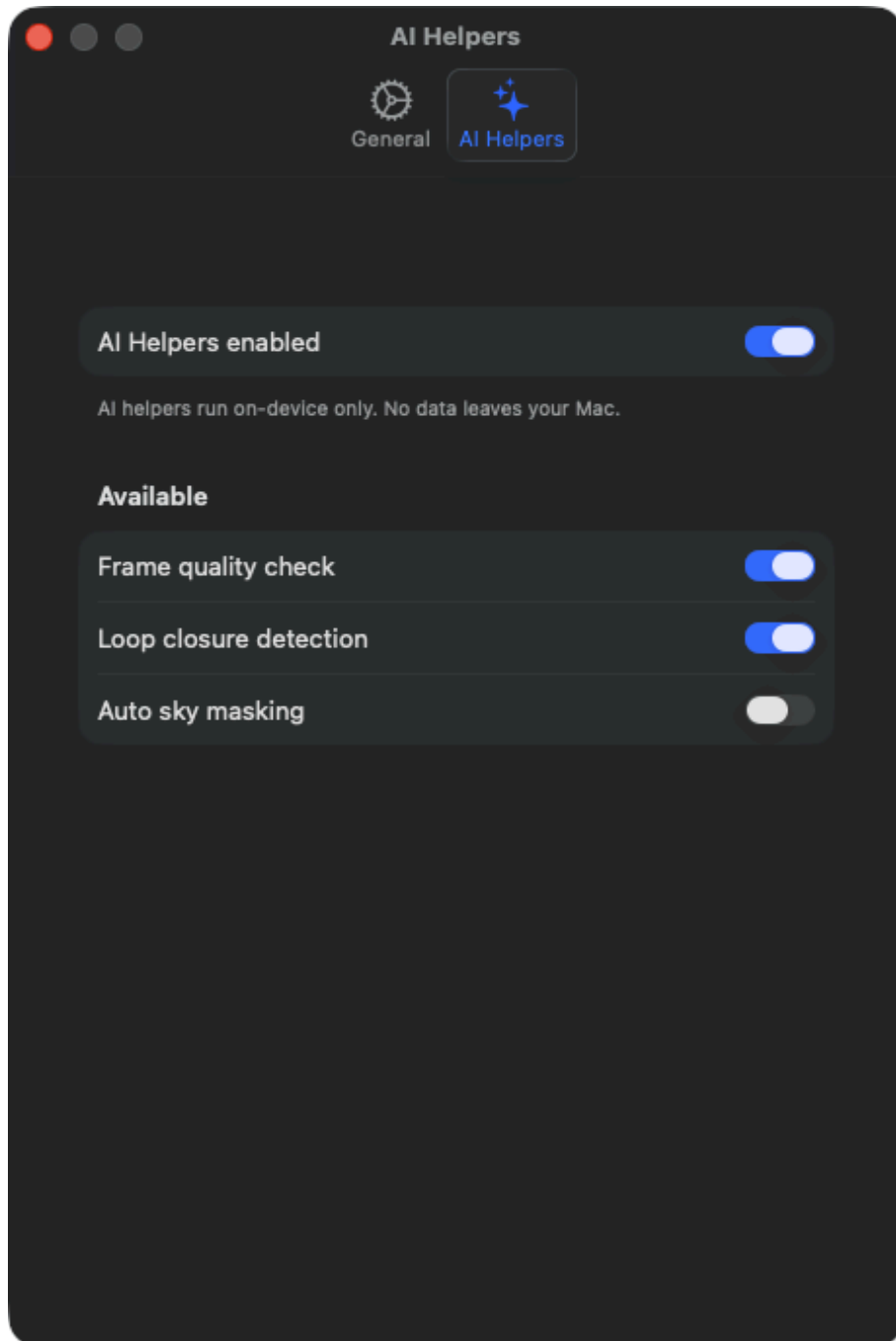



Рисунок 16. Settings → вкладка AI Helpers с главным переключателем и подпереключателями

**S11 AI Helpers включены (Master)**

Settings → AI Helpers → первая секция → переключатель «AI Helpers enabled». Bound: По умолчанию: `true`.



Главный переключатель над всеми функциями AI Helpers в конвейере. Когда выключен, конвейер импорта и SfM полностью пропускает все этапы ML-предобработки — никаких вызовов Apple Vision, никакой загрузки моделей CoreML, никакого пробуждения NPU. Когда включен, опрашиваются отдельные подпереключатели (S12–S13). Значение запоминается между перезапусками. Затрагивает следующие этапы: (a) предварительный контроль качества кадра перед SfM (S12), (b) обнаружение замыкания петли (S13). Важно: когда выключен, два подпереключателя деактивированы и визуально серые. Сноска в нижней части подчёркивает, что все AI Helpers работают строго on-device — никакой загрузки изображений, никакой облачной обработки. Гарантия приватности обеспечивается исключительным использованием фреймворка Apple Vision (локально на Neural Engine) и моделей CoreML, которые поставляются прямо в бандле приложения.

 **ПРОСТЫМИ СЛОВАМИ**

Главный переключатель всех функций, которые внутри используют AI/машинное обучение. По умолчанию — «вкл», потому что помощники экономят много времени, не давая вашим изображениям покидать Mac. Если вы хотите их полностью отключить (напр. для экономии энергии или потому что у вашего Mac нет NPU), отключите здесь — две подопции ниже тогда автоматически станут серыми и больше ничего не делают.


**S12** Контроль качества кадра

ГДЕ

Settings → AI Helpers → секция Available → переключатель «Frame quality check». Bound: По умолчанию: `true`.

 ТЕХНИЧЕСКИ

Активирует screener качества кадра (фаза 3.11), который анализирует каждый импортированный кадр перед вызовом SfM. Шаги конвейера на кадр: (a) Laplacian variance-фильтр из Apple Vision (обнаружение размытия — порог ~150), (b) гистограмма-based проверка над-/недо-экспозиции (порог: >5 % пикселей на 0 или 255), (c) blank frame detect (стандартное отклонение < 5 по всем пикселям). Кадры, проходящие все три проверки, пропускаются напрямую. Кадры, не прошедшие хотя бы одну проверку, триггерят модальный диалог подтверждения, который перечисляет каждый проблемный кадр с миниатюрой и причиной и спрашивает, нужно ли его удалить. Важно: никакого автоматического удаления — диалог всегда требуется, окончательное решение остаётся за пользователем. Производительность: ~50 мс на кадр на M3 Ultra, работает параллельно. Когда выключен, все кадры передаются в SfM без проверки. Если мастер (S11) деактивирован, этот переключатель визуально серый и не имеет эффекта. Статус поставки по memory: SHIPPED 2026-05-23.

 ПРОСТЫМИ СЛОВАМИ

Перед собственно обучением приложение смотрит на каждую фотографию: размыта ли? полностью тёмная или белая? пустая? Если да, оно спрашивает, не хотите ли вы выбросить изображение — оно никогда не удаляет ничего автоматически. Это потом экономит много часов, потому что одна полностью размытая фотография иногда может испортить всё обучение. По умолчанию — «вкл», потому что стоимость почти нулевая, а выгода большая.

**S13** Обнаружение замыкания петли

ГДЕ

Settings → AI Helpers → секция Available → переключатель «Loop closure detection». Bound: По умолчанию: `true`.

## ТЕХНИЧЕСКИ

Активирует обнаружение замыкания петли на основе feature-print Apple Vision. Для каждого импортированного кадра вычисляется вектор признаков размерностью ~768, представляющий нейронное embedding содержания изображения. Все feature-prints затем попарно сравниваются через cosine-similarity. Пары с similarity > 0.85 и расстоянием по индексам кадров > 50 (т.е. несоседние кадры) идентифицируются как «loop closure candidates» и записываются в sidecar JSONL-файл в папке проекта. Только информация — импортированная последовательность изображений не модифицируется. Назначение: даёт SfM-решателю (особенно COLMAP) подсказку, что эти кадры кластеризуются в 3D-пространстве. Для нативного SfM информация sidecar пока только документационная; COLMAP использует подсказки внутренне через custom matches-файл (ручная интеграция возможна, не подключается автоматически). Производительность: ~200 мс на кадр на M3 Ultra, работает параллельно. Когда выключен, feature prints не генерируются. Если мастер (S11) деактивирован, визуально серый.

## ПРОСТЫМИ СЛОВАМИ

Когда вы обходите объект во время съёмки и снова оказываетесь у точки старта, компьютеру это очень помогает знать. Эта опция автоматически определяет, какие из ваших фотографий сделаны «почти с того же места», и записывает это в маленький вспомогательный файл. SfM-инструменты (особенно COLMAP) могут использовать эту информацию для более чистой 3D-реконструкции. По умолчанию — «вкл», потому что работает без вашего вмешательства и ничего не меняет в ваших изображениях.

## Зеркальные настройки Инспектора

Остальные записи настроек (S17–S33) из инвентарной таблицы являются зеркалами из Expert Inspector и документируются в главе 2 (элементы управления Инспектора I12–I29). Они физически не появляются в окне настроек, а были перечислены в инвентаре только потому, что выполняются через свойства `TrainingConfig`, которые сохраняются через и поэтому формально имеют характер настроек. Содержательные объяснения — см. там.

## Когда что?

Настройка	Область	Сохранение
S1 Default Mode	Глобально для приложения	Перезапуск
S2 Language	Глобально для приложения	Перезапуск
S3 Viewport Background	Глобально (по умолч.) + Runtime	Перезапуск
S4 Auto-Rotate After Training	Глобально для приложения	Перезапуск
S5 Live Preview Interval	По умолчанию для новых тренировок	Перезапуск
S6 Throttle Delay	По умолчанию для новых тренировок	Перезапуск
S11 AI Helpers Master	Глобально для приложения	Перезапуск
S12 Frame quality check	Глобально для приложения	Перезапуск
S13 Loop closure detection	Глобально для приложения	Перезапуск

«Глобально для приложения» = затрагивает все проекты. «По умолчанию для новых тренировок» = затрагивает только следующую создаваемую тренировку; идущие сессии остаются без изменений. «Текущая тренировка» = вступает в силу немедленно для текущей конфигурации обучения, но не сохраняется без явного повторного импорта.

## ГЛАВА

## Глава 4 — Вспомогательные окна

---

Besides the main window (3D viewport plus Inspector), RadianceKit manages seven additional windows, all of which are opened via the Help menu. The list from top to bottom: User Guide (⌘?), Keyboard Shortcuts (⌘/), Open Training Logs... (does not open an app window, but the Finder; therefore not covered further here), Manage Storage..., Pareto Dashboard... (⇧⌘D), Holdout Analysis... (⇧⌘H), BayesOpt Console... (⇧⌘B). Three of these — Dashboard, Holdout, BayesOpt — are standalone analysis tools. They each have their own view-model stack, read or write JSON files on disk, and there is a CLI argument for each that lets you point the window at a specific file directly at app startup ( `--dashboard-dir` , `--holdout-file` , `--bayesopt-autorun` ).

The four simple windows (User Guide, Keyboard Shortcuts, Manage Storage, plus the submenu items Open Training Logs / Open Exports Folder) get a short entry per control. The three analysis windows are documented in more detail — each with an introduction explaining what you see in the window, when you should open it, and how to interpret the picture shown.

At the end of the chapter there is a cross-reference section to the main window's Inspector: what you can meaningfully read from the live loss chart and the Gaussian count display during a running training.

## User Guide (W1–W4)

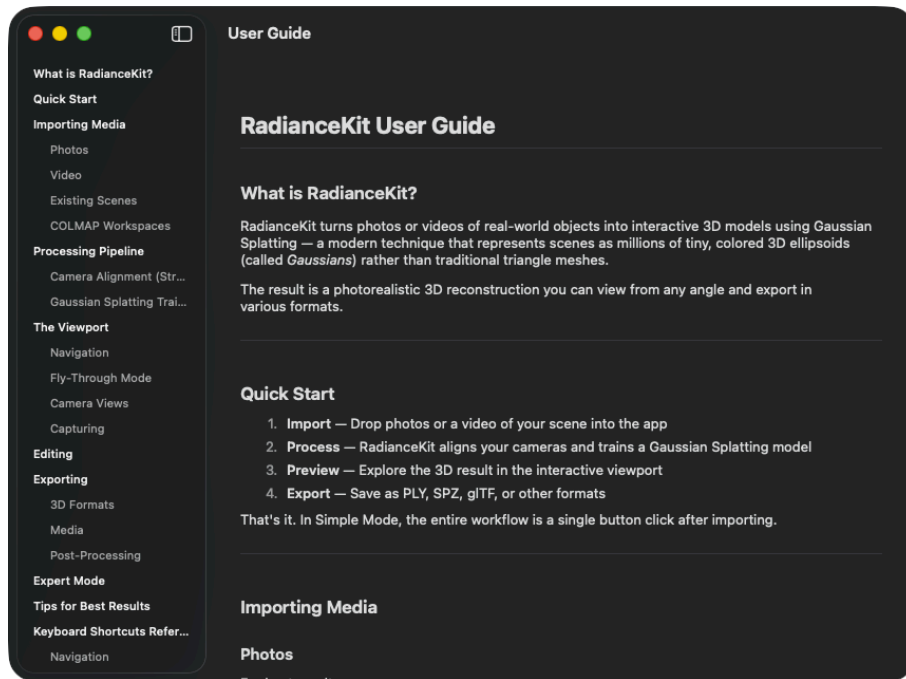


Рисунок 17. User Guide window with sidebar on the left and rendered Markdown content on the right

**What it is:** A built-in help window that renders the `guide_<language>.md` shipped with the app. The language is derived from the Settings (General tab → Language) or, if «System» is set there, from the macOS language preferences. Layout is classic: sidebar with all headings on the left, body text on the right.

**WHEN TO OPEN** When you need a quick reminder of a single point — so as a keyword substitute. The full reference is this manual; the built-in help window is more like what a `--help` would be on the command line. It is updated with every app release, but is intentionally kept more superficial in content.

## W1 NavigationSplitView (Sidebar + Detail)



Help → User Guide (⌘?)..



Two-column layout with a narrow sidebar (at least 180 pt wide) for the content tree and a scrollable detail area for the actual Markdown content. The window has a minimum size of 700 × 500 pt. On first open, the window loads the matching `guide_<lang>.md` from the app bundle (fallback `guide_en.md`), parses it into block records (headings H1–H4, paragraphs, lists, tables, separators) and separately extracts the heading structure for the sidebar. Inline formatting (bold, italic, code-span) is rendered via the built-in Markdown engine. The language is read from the app settings, with the special case of Chinese ( `zh-Hans` ) and Brazilian Portuguese ( `pt-BR` ), which are kept as full locale tags because these variants differ from `zh` and `pt` respectively.

### ПРОСТЫМИ СЛОВАМИ

The built-in help text, with the topic list on the left and the content on the right. The language adjusts automatically based on your system settings. Works offline, but is intentionally only a short version — the full reference is this manual.

## W2 List (Heading sidebar)



Left column in the User Guide window..



List of all H2 and H3 headings of the current Markdown document. H2 entries appear without indentation in Medium font weight, H3 entries with 16 pt left indentation and a reduced foreground style. H4 and deeper are ignored, because the depth would otherwise make the sidebar cluttered. Anchor IDs are generated from the heading text via slugification (lowercase + spaces to dashes + filtering for letters/numbers/dashes — the same algorithm GitHub uses for its Markdown anchors, so external URLs to the docs would potentially land on the same anchor as well). The list uses the native macOS style.

### ПРОСТЫМИ СЛОВАМИ

The navigation bar on the left side. Tap an entry and you jump to the section.

### W3 Button (Heading → anchor jump)



One button per sidebar row..



Each sidebar entry is a button that sets the current anchor, but visually looks like a list entry. An observer variable then triggers the scroll jump to the matching anchor with a smooth animation over 0.3 s. After the jump, the anchor value is reset so that the next click on the same anchor fires again (otherwise the observer wouldn't trigger again because the value hasn't changed).

#### ПРОСТЫМИ СЛОВАМИ

Clicking takes you to the corresponding spot in the text on the right.

### W4 ScrollView (detail content)



Right column..



Scrollable, vertically stacking content area with lazy rendering, because longer guides can easily have over 200 Markdown blocks — a non-lazy variant would instantiate them all at once. Each block gets its own ID, either the heading anchor (for jumpable H1–H3) or an index placeholder. Maximum width is 720 pt, padding 32 horizontal / 24 vertical, so long lines retain a well-readable layout. Tables are rendered cell-by-cell with horizontal stacks and separators; inline code via the built-in Markdown engine. Real code blocks are currently treated as paragraphs — a known limitation of the Help window.

#### ПРОСТЫМИ СЛОВАМИ

The actual help text. Scrollable, well-readable width, clear typography.

## Keyboard Shortcuts (W5–W6)

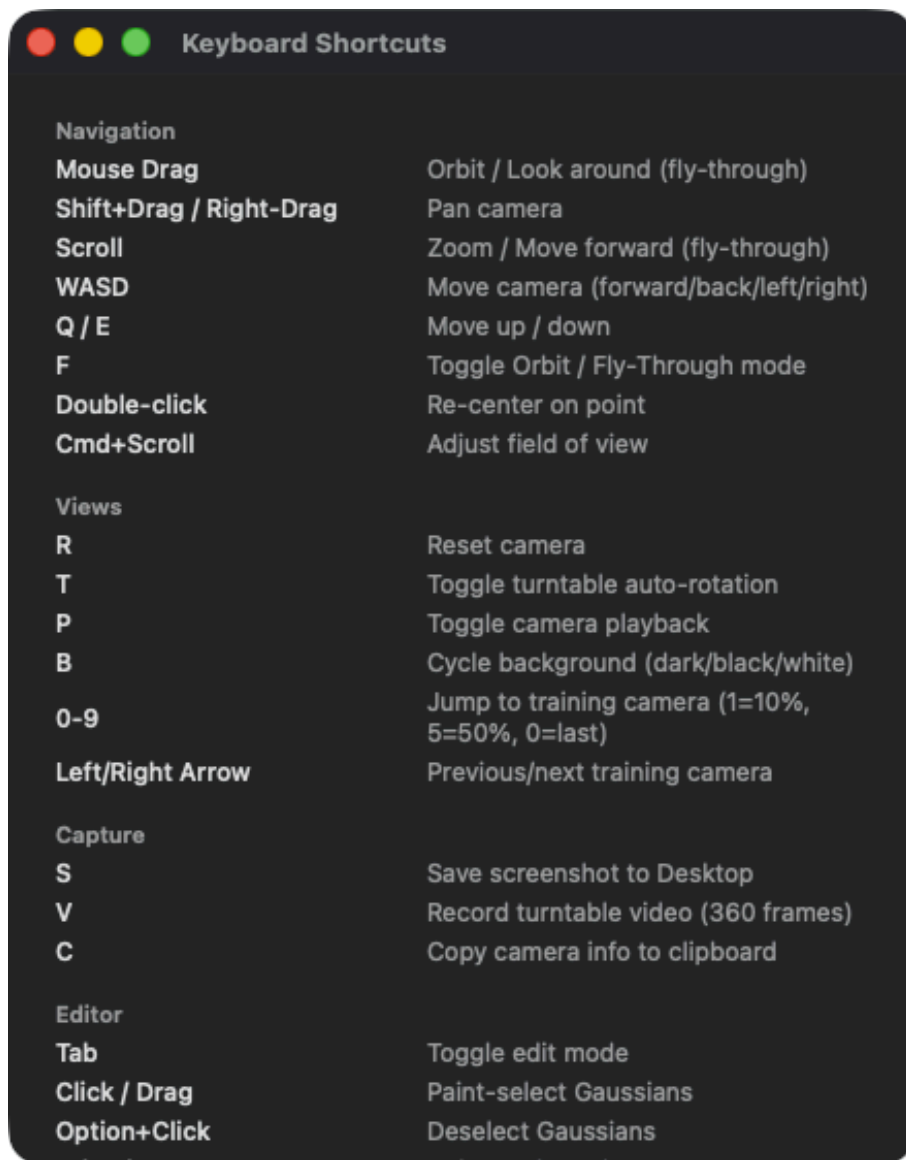


Рисунок 18. Keyboard Shortcuts window — five groups Navigation/Views/Capture/Editor/Training with hotkey column on the left and description on the right

**WHAT'S IN THE IMAGE** Static reference list in five sections. **Navigation:** Mouse Drag (Orbit/Fly), Shift+Drag/Right-Drag (Pan), Scroll (Zoom), WASD (fly-through movement), Q/E (Up/Down), F (Toggle Orbit/Fly), Double-click (Re-center), Cmd+Scroll (FoV adjust). **Views:** R (Reset Camera), T (Auto-Rotation), P (Camera Playback), B (Background cycle), 0–9 (jump to training cam 1=10%/5=50%/0=last), Left/Right Arrow (Prev/Next Cam). **Capture:** S (Screenshot to Desktop), V (Turntable Video), C (Copy Camera Info). **Editor:** Tab (Edit mode), Click/Drag (Paint-Select), Option+Click (Deselect), X / Delete (delete selection), Cmd-Z (undo last deletion), [ / ] (brush size smaller/larger), Esc (clear selection). **Training:** Start, Pause/Resume, Cancel, Continue +5K/+10K/+20K via the menu shortcuts in M9–M14.

**What it is:** A simple static overview of all keyboard shortcuts — Navigation, Views, Capture, Editor, Training. Content is hardcoded in, no Markdown loading.

**WHEN TO OPEN** When you're looking for the fastest way to do something in the viewport. WASD fly-through, R for camera reset, B for background cycling — they're all here.

### W5 ScrollView (content area)

 ГДЕ

Help → Keyboard Shortcuts (⌘/)..

 ТЕХНИЧЕСКИ

A simple scroll area with a vertical list inside. Padding 20 all around, no sidebar navigation tree (the list is short enough). Content is grouped into five sections (Navigation, Views, Capture, Editor, Training). One row per key combination with translatable text in both columns. Left column (key code) fixed to 180 pt width, so the descriptions on the right stay vertically aligned. No interaction except scrolling — clicking on a row doesn't trigger anything; the shortcuts are real keyboard modifiers in the menu and on the viewport.


 ПРОСТЫМИ СЛОВАМИ

Table of all shortcut keys. Static cheat sheet for quick lookup.

### W6 VStack (shortcut sections)

 ГДЕ

Inside the ScrollView..

 ТЕХНИЧЕСКИ

Left-aligned stacked sections with 16 pt spacing. Within the five sections, each contains a heading + row sequence. Headings use a secondary subheadline style — intentionally not a title format, because the sections don't need to be navigable. Content is intentionally flat (no disclosure, no search, no filter) so that the component runs unchanged on every macOS version and the file stays readable.

 ПРОСТЫМИ СЛОВАМИ

The grouping of keys by function (Navigation, Views, Editor, and so on).

## Manage Storage (W7–W12)

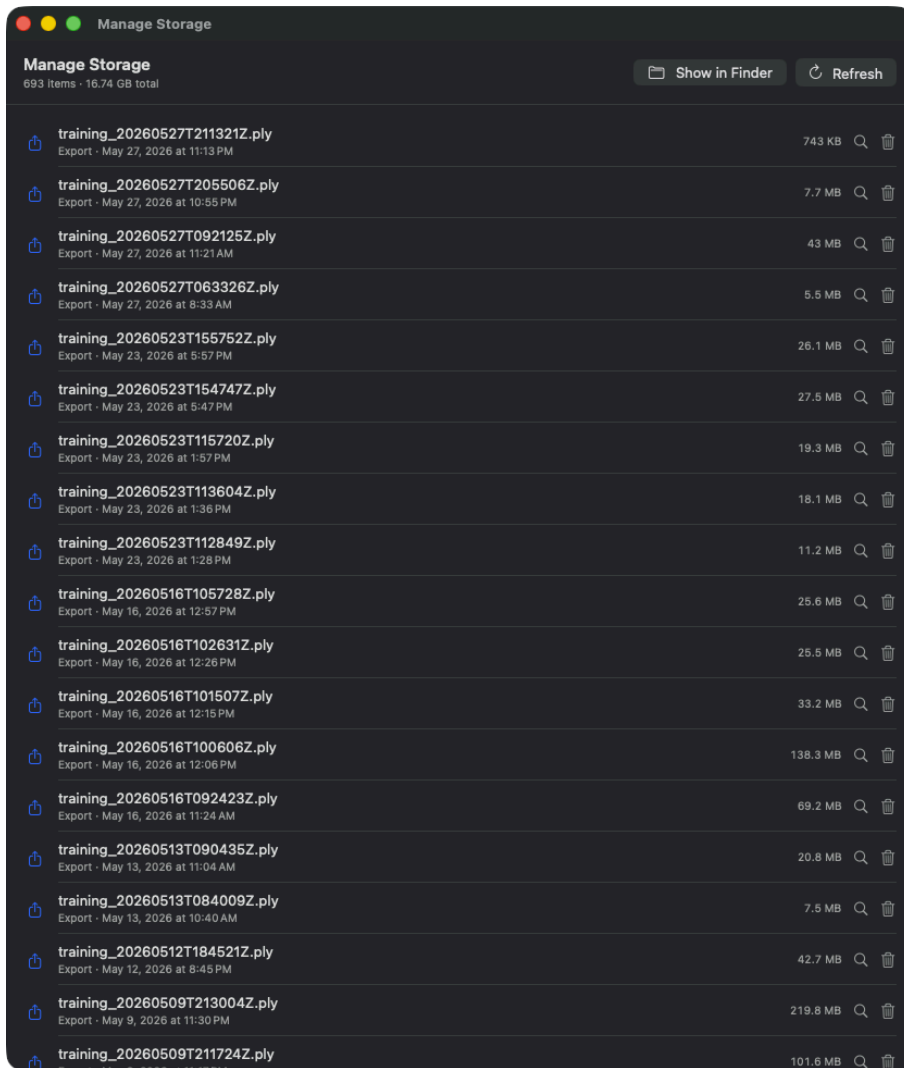


Рисунок 19. Manage Storage window — header shows „693 items · 16.74 GB total«, table with Export PLY files sorted by date, each with format pill + filename + size + date

**WHAT'S IN THE IMAGE** Table view of all files managed by RadianceKit. Header counts 693 items, 16.74 GB total size. Toolbar at top: «Show in Finder» + «Refresh». Each row: PLY icon, filename (e.g. training\_20260527T211321Z.ply), export date, size (varies 7 KB to 218 MB), magnifier icon (Reveal) and trash icon (Move to Trash). Files are sorted by date, newest at the top. In this demo recording PLY exports dominate because a lot of work was done with `--benchmark`.

**What it is:** A disk usage overview of everything RadianceKit stores under `~/Documents/RadianceKit/` — Logs, Exports, Scenes, Capture bundles (from the iOS companion), Imports (staging copies of the input images). One size in bytes per entry and two buttons: «Show in Finder» and «Move to Trash». This is NOT automatic cleanup — the app doesn't delete anything itself; you decide per entry.

**WHEN TO OPEN** When the disk is filling up. Logs in particular accumulate (one JSONL per training attempt, plus the `_qualityMetrics.json`); exports too of course (PLY 100% raw data, one per export). Also useful after a crash when the imports staging directory

still has old copies of the input images lying around (see «Disk pressure incident» in `dev_v549f-needle-reduction.md`).

### W7 Button «Show in Finder»



Top right of the header in the storage browser window..



Opens the entire RadianceKit directory (`~/Documents/RadianceKit/`) in the Finder, so you can see the folder structure directly and also manipulate it with the Finder itself. The action opens a new Finder window and does not switch into the app sandbox container — `~/Documents/RadianceKit/` is the regular Documents domain accessible to apps, not a sandboxed container path.

#### ПРОСТЫМИ СЛОВАМИ

Opens the directory in the Finder so you can work with the files yourself.

### W8 Button «Refresh»



Header, next to the Finder button..



Triggers a background scan that runs on a user-initiated asynchronous task so that scanning large directory trees doesn't block the UI. The actual walk goes through every known subfolder (Logs, Exports, Scenes, Captures, Imports) and produces one storage entry per direct child. Per entry the recursive size is determined — preferably the actual disk usage (including APFS hardlink sharing) with fallback to the logical file size.

#### ПРОСТЫМИ СЛОВАМИ

Re-reads the list in case you deleted or added something in the Finder in between.

**W9 List (storage entries)**

Main content below the header..



List with this layout per row: category- specific SF Symbols icon (document for Logs, upload arrow for Exports, cube for Scenes, tray for Imports), name + subtitle (Kind label + formatted modification date), bytes counter on the right (right-aligned, monospaced), Reveal button (magnifier icon), Trash button (trash can). Sorting: primarily by Kind (Scenes first, then Exports, Logs, Captures, Imports, Other), secondarily by modification date descending (newest at top). If the scan is still running, the area shows a «Scanning...» progress indicator instead. If nothing was found, an empty-state display with a tray icon.

**ПРОСТЫМИ СЛОВАМИ**

List of all your RadianceKit data, sorted by type and recency. Per entry you see the size and can delete directly.

**W10 Row button «Reveal in Finder»**

Per row, magnifier icon on the right..



Opens the Finder and selects the specific item (file or folder). Difference from W7: W7 opens the root directory; W10 highlights exactly this one entry. Practical workflow: identify a large entry, click on the magnifier, then copy it to an external volume for example.

**ПРОСТЫМИ СЛОВАМИ**


Jumps to this entry directly in the Finder so you can find it quickly.

**W11 Row button «Move to Trash»**

Per row, trash icon on the right next to the magnifier..



Triggers the confirmation dialog box (W12). Only after confirmation does the macOS standard «move to trash» operation run (so reversible, no direct deletion). After successful trash, the entry is removed from the list and the total byte counter is updated. On errors, a modal error dialog is displayed.

 ПРОСТЫМИ СЛОВАМИ


Moves the entry to the trash. A dialog asks first.

**W12 ConfirmationDialog (delete confirmation)**

Triggered by W11, displayed as a macOS sheet..



Standard confirmation dialog with a dynamic title «Delete <name>?» and a message line that explicitly points out that the entry goes to the trash and can be restored from there (until the trash is emptied). Two buttons: «Move to Trash» as destructive action (shown in red) and «Cancel» with automatic Esc binding. The dialog is non-modal in the sense that it only blocks this window, not the whole app — that's macOS standard for reversible deletions.

 ПРОСТЫМИ СЛОВАМИ

Safety prompt before deletion. «Move to Trash» is reversible — as long as the trash isn't emptied.

## Pareto Dashboard (W13–W22)

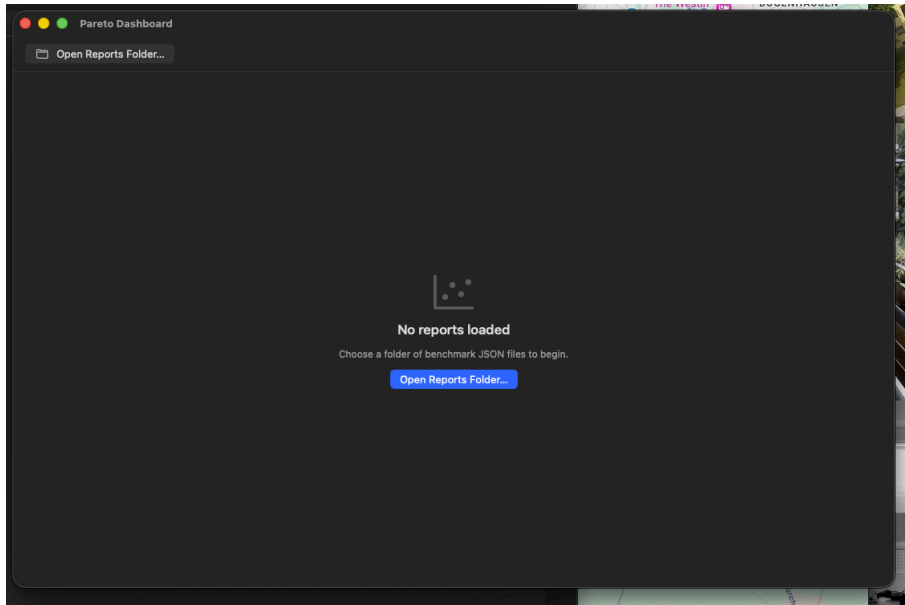


Рисунок 20. Pareto Dashboard — empty state before report import

Empty state (after first open) — empty state with call-to-action «Open Reports Folder...». The data points appear as soon as training reports are loaded, see next shot.

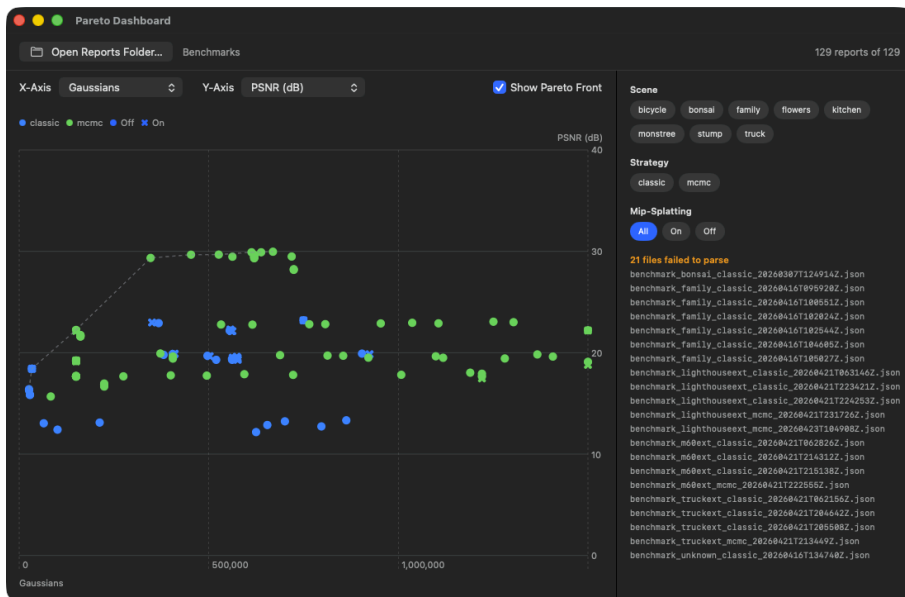


Рисунок 21. Pareto Dashboard with 129 loaded benchmark reports — Gaussians vs PSNR with Pareto front, Scene/Strategy/Mip filter

**WHAT'S IN THE IMAGE** Header toolbar shows «129 reports of 129» (all reports in the selected folder parsed successfully — 21 additional files could not be parsed due to older format, see hint list on the right). Axes: X-axis picker on Gaussians , Y-axis picker on PSNR (dB) . Scatter plot: green points = Classic strategy, blue points = MCMC. The dashed Pareto front line runs along the best-achieved PSNR values and plateaus around PSNR≈30 dB from about 500K Gaussians onwards. Filter chips on the right: 7 scenes


(bicycle, bonsai, family, flowers, kitchen, stump, truck), 2 strategies (classic, mcmc), 3 Mip-Splatting options (All, On, Off). Currently all filters are open, hence the dense point cluster.

**What it is:** A multi-run comparison tool. In the past you have trained several scenes, or the same scene with different presets — each of these training runs produces (if you passed `--benchmark` or called via the Benchmark function) a JSON report file containing among other things final PSNR, SSIM, LPIPS, Gaussian count and wallclock time. The Dashboard reads a whole folder of such reports at once and plots them as a 2D scatter with selectable axes. Additionally, the Pareto front (the set of non-dominated points) is drawn as a dashed line.

**WHEN TO OPEN** After you have produced at least three or four training reports. With fewer points the frontier line is not meaningful. Typical use case: you tried to reconstruct an outdoor scene and successively ran P3 Balanced (Classic), P4 Quality (Classic), P7 MCMC Quality and P9 Outdoor (tuned) — now you want to know which configuration delivers the best PSNR per second of training time, or which requires the fewest Gaussians for a given PSNR.

**HOW TO INTERPRET** Both axes are freely selectable (X-axis:,, `psnr`, `ssim`, `lpips`, ...; Y-axis the same). The Pareto front logic in `ParetoFront2D.indices` knows for each metric whether «smaller = better» (e.g. LPIPS, Loss, Time) or «larger = better» (PSNR, SSIM) — so depending on the axis choice, the line runs from bottom-left to top-right or from top-left to bottom-right, always along the best achieved combination. A point is Pareto-optimal if NO other point is at least as good in BOTH dimensions (so no other point dominates it). Pareto-optimal points lie on the line, other points to the right/above (depending on axis orientation). Points ON the line are the real candidates for «best preset»; points FAR from the line are wasted training time.

**FILTER CHIPS** You can restrict the selection to a particular scene (if you only want to compare outdoor runs, e.g.), to a particular strategy (Classic or MCMC), or to Mip-Splatting on/off (relevant after Phase Q1.5, where Mip remains as an opt-in advanced flag).

 You have three reports for the «truck» scene under `~/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). Set X-axis to `trainingTime`, Y-axis to `PSNR`. Run B lies upper right, Run C even further upper right, Run A lower left. The Pareto front connects A and C — both non-dominated. Run B is «lost» (C is better in Time AND PSNR). Insight: for «truck» the MCMC default isn't worth it; either fast+ok (A) or long+very good (C). Save the configuration from C as your own preset (Inspector → I1 Save Preset).

**Next action:** Save the best configuration as a preset. Concretely: look at the Pareto points (hover shows PSNR/SSIM/LPIPS/Gs/Time in the tooltip), decide which one suits you best in the time-vs-quality trade-off, open the corresponding report (filename contains run timestamp), copy its training configuration into a new run, or save it after the next training session as a preset via the Inspector.

**W13 Button «Open Reports Folder...»**

Toolbar top left..



Opens a folder picker dialog with the prompt «Select a folder containing benchmark .json reports». After confirmation, a background task runs that parses all `.json` files in the folder sequentially. Faulty reports (broken JSON, wrong schema) are collected and shown at the bottom of the sidebar as «N file failed to parse» — no crash. If a second click happens while a first load is still running, the previous task is canceled so that two results don't write into state simultaneously.

Also via CLI: `--dashboard-dir /path/to/reports` loads the folder directly at app startup.

 ПРОСТЫМИ СЛОВАМИ


Selects the folder where your benchmark reports live. Default path is `~/Documents/RadianceKit/Reports/`. Then loads all JSONs at once.

**W14 Picker «X-Axis»**

Above the chart, on the left..



Menu picker with all available metric axes of the dashboard module (PSNR, SSIM, LPIPS, Gaussian count, training time and so on). Default is Gaussian count. On change, the hovered point is reset because a previously highlighted position in the old axis coordinate system no longer makes sense after axis change. The picker is constrained to content width so it doesn't span the entire width.

 ПРОСТЫМИ СЛОВАМИ

Which metric should be on the horizontal axis. Usually «training time» or «Gaussian count» because those are the «costs» you want to compare.

**W15 Picker «Y-Axis»**

Above the chart, next to X-Axis..



Identical to W14, except the default is PSNR. The axis choice is stored independently, so the user can also pick nonsense combinations (X=PSNR, Y=PSNR — would throw all points onto a diagonal). Such combinations are not caught, however; a deliberate decision, because a comparison «SSIM vs PSNR» is quite interesting for seeing how consistently the metrics behave.

**ПРОСТЫМИ СЛОВАМИ**

What's on the vertical axis. Normally «PSNR» or «SSIM» as a quality measure.

**W16 Toggle «Show Pareto Front»**

To the right of the axis pickers..



Standard macOS toggle. If active, a line with the computed 2D Pareto front is drawn in the Pareto chart in addition to the point cloud. Style: dashed (dash pattern 4–4), gray semi-transparent, line width 1.5 pt. The Pareto computation runs on the main thread — with the typical number of reports ( $\leq \sim 50$ ) this is fast without issue. If the toggle is off, the line is omitted so only the bare points remain.

**ПРОСТЫМИ СЛОВАМИ**

Shows the line running through the «best-so-far» points. If the line is in the way (e.g. because you only want to compare individual trades), turn it off.

**W17 Chips «Scene» filter**

Right sidebar in the Dashboard window..



Filter chips for every scene appearing in the loaded reports. Custom flow layout that automatically wraps chips into multiple lines as soon as the width is exhausted. Active chips get the accent background, inactive ones a neutral standard material background. Multi-selection is possible (set semantics); if no chip is selected, all scenes count as «passed through» — i.e. the set logic is «empty selection = all», not «empty selection = nothing».

 ПРОСТЫМИ СЛОВАМИ


Clicking a scene name filters the points to only this scene. Multi-selection possible. Empty = all scenes.

**W18 Chips «Strategy» filter**

Below Scene filter in the sidebar..



Exactly like W17, but for training strategies — typically the two values «classic» and «mcmc», derived from the strategy field of the benchmark report JSONs. Helpful if you have mixed reports of both strategies and only want to see one kind (e.g. «only show MCMC runs because I've already excluded Classic»).

 ПРОСТЫМИ СЛОВАМИ

Filter by Classic or MCMC. By default both are active.

**W19 Chips «Mip-Splatting» filter**

Below Strategy filter in the sidebar..



Three-valued filter (instead of a set like W17/W18): «All» / «On» / «Off». Background: Mip-Splatting was evaluated in Phase Q1.5 as an experimental multi-scale improvement and the final verdict was «no nice win across the board; keep as opt-in flag». When you do Mip on/off comparisons you often want to separate very sharply. Hence the dedicated ternary filter with the states «let everything through», «only Mip on», «only Mip off». The sidebar section is only displayed if at least one Mip report AND at least one non-Mip report is in the data set (otherwise filtering makes no sense).

**ПРОСТЫМИ СЛОВАМИ**

If you want to compare Mip-Splatting on/off, here is a three-way filter. Otherwise ignore.

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

Helper component, used in W17/W18/W19..



Minimalist button wrapper. Content: label text with caption font size and padding 10 horizontal / 5 vertical. Background conditional: if active → app accent color with white text; otherwise neutral standard material background with black text. Shape is a capsule (pill-like). Plain button style so the capsule material isn't overlaid by a system border.

**ПРОСТЫМИ СЛОВАМИ**

The round filter buttons themselves. Visually like an iOS tag.

**W21 Chart (Pareto scatter)**

Center area of the Dashboard..



Swift Charts diagram with two layers: 1. one point per report — position from the chosen X and Y metrics, color by strategy, symbol by Mip status. Symbol size normal 80, highlighted 200 (if the ID matches the currently hovered report).  
2. a line for the Pareto front, only if the toggle is on.

Chart overlay: a transparent rectangle registers mouse motion; per frame the Euclidean nearest point position in the plot frame is determined and the hovered report is updated if the distance is under 24 px (otherwise reset). So you get the tooltip without clicking — hovering is enough.

 ПРОСТЫМИ СЛОВАМИ


The actual scatter plot. Each point is a training run. Hover for detail tooltip.

**W22 Tooltip (hover detail)**

Below the chart, displayed on hover..



Horizontal stack: scene name (headline), strategy tag (caption), separator, then PSNR/SSIM/LPIPS/ Gs/Time metrics each in a small vertical group (label + monospaced value). If Mip was activated, additionally a «Mip» capsule tag in accent color. Background semi-transparent blur, rounded rectangle with 8 pt radius. Only displayed when the mouse is actually over a point. Disappears automatically on leaving.

 ПРОСТЫМИ СЛОВАМИ

The detail card at the bottom when you hover over a point with the mouse. Shows all quality metrics and the run configuration at once.

## Holdout Analysis (W23–W29)

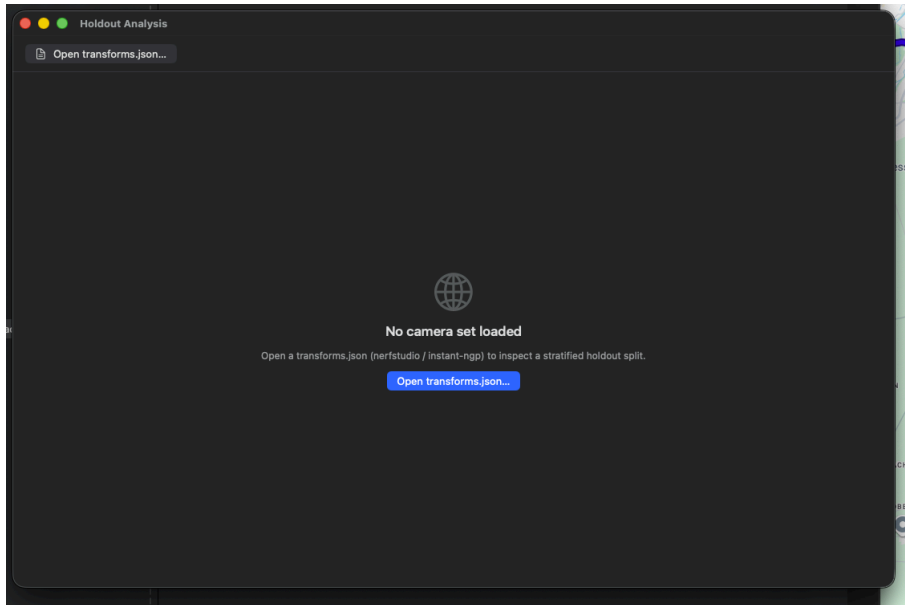


Рисунок 22. Holdout Analysis — empty state before loading a transforms.json

Empty state with empty state and call-to-action «Open transforms.json...». Accepts NeRF Studio and Instant-NGP format.

Empty state (after first open) — the camera markers appear as soon as a transforms.json is loaded, see next shot.

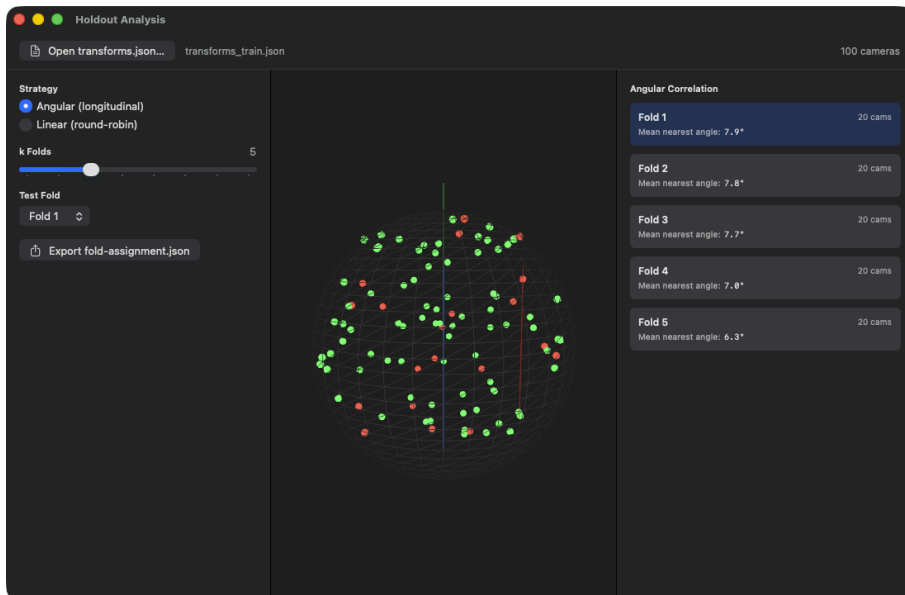


Рисунок 23. Holdout globe with 100 NeRF Blender mic cameras, 5 folds of 20 cameras each, angular strategy active

**WHAT'S IN THE IMAGE** Header shows loaded file (transforms\_train.json) and cam count («100 cameras»). Left sidebar: strategy picker with two options — Angular (longitudinal) active (aligns folds along longitudinal/latitudinal sectors on the sphere so each test fold is geometrically dense) vs Linear



(round-robin) (order-based, every k-th frame as the test set). k-folds slider is on 5, test-fold picker on Fold 1. Export button produces a `fold-assignment.json` for Nerfstudio/Instant-NGP. Middle panel: 3D globe projection of all 100 cameras — green points = Train, red points = current test fold (Fold 1 with 20 cameras). Right sidebar (Angular Correlation): per fold 20 cams + Mean Nearest Angle (Fold 1: 7.9°, Fold 2: 7.8°, Fold 3: 7.7°, Fold 4: 7.0°, Fold 5: 6.3°) — a smaller value means the cameras within this fold lie close together, so the Holdout split is spatially coherent.

**What it is:** A 3D visualizer for your camera arrangement with cross-validation logic. You load a `transforms.json` (the standard format from Nerfstudio / Instant-NGP for camera poses), the app reads all cameras, projects their view directions onto a unit sphere and shows them as small sphere markers on a virtual globe. Then it divides the cameras into `k` folds (per the selected strategy: angular or linear), marks the training portion green and the test portion red (Holdout), and computes per fold an Angular Correlation score that tells you how far away the test fold is from the training fold in view-angle space.

**WHEN TO OPEN** When you want to do Holdout evaluation — i.e. how well does your model generalize to unseen viewpoints? Default in training is «every-8th view as Holdout» (Mip-NeRF360 convention), but that is a very linear split. If your images are clustered in time for example (one side of the object first, then the other), then «every-8th» is not representative — a random sequence position ends up in the test, but all its neighbors are in training, which is too easy. With «angular» you stratify across the view-angle space instead: each fold contains cameras from all areas of the orbit so the test really probes generalization gaps.

**HOW TO INTERPRET** Angular vs Linear: - Angular (default): divides the cameras by longitudinal angle ( $\phi$  coordinate around the Y axis) into `k` equal sectors. Fold 0 contains cameras with  $\phi \in [0^\circ, 360/k^\circ)$ , Fold 1 the next ones, and so on. Advantage: each fold covers a portion of the orbit; the test fold is spatially compact but broadly distributed across the world dataset. Good for classic orbit recordings. - Linear (round-robin): Fold index = (image\_index modulo k). That is the simple «every k-th» split. Works if the image order has NO spatial bias (e.g. randomly sorted drone shots). Works poorly if the images cluster in time.

In the 3D globe you immediately see: green points (training) and red points (test). If the red points all cluster in one corner, the Holdout is poor (no good generalization test). If they lie evenly between the green ones, it's good. The Angular Correlation score per fold (right sidebar, in degrees) additionally says: smaller value = the test is close to training (each test camera has a nearby training camera, easy test); larger value = the test is far from training (harder generalization).

 You captured your truck scene with 251 images, export via menu item M33 (Export SfM transforms.json) a Nerfstudio file. Open the Holdout window ()H), load the JSON via «Open transforms.json...», look at the globe. `k=5` (default) gives you 5 folds. Click on «Fold 3» — check whether the red markers are reasonably even. If yes: «Export fold-assignment.json», put the exported file in the reports folder, and at the next training run with `--benchmark` (or the corresponding Inspector settings) exactly this fold assignment is used as the test Holdout — instead of the default «every-8th».


**W23** Button «Open transforms.json...»

Toolbar top left..



Opens a file picker dialog restricted to JSON files. After confirmation the Holdout module loads the file. The loader parses both the Nerfstudio format (camera intrinsics plus list of frames with image path and transform matrix) and the Instant-NGP format (same structure). For each frame the view direction is extracted from the transform matrix (z-axis of the camera local basis) and stored. If parsing fails, an error message is shown in the status area.

Also via CLI: `--holdout-file /path/to/transforms.json` opens the window directly with the file loaded.

 ПРОСТЫМИ СЛОВАМИ

Loads your camera poses JSON. Standard are Nerfstudio and Instant-NGP exports. RadianceKit itself can export transforms.json via Menu → Export → SfM.

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

Left sidebar, at the top..



Radio picker with two options: Angular and Linear. Strategy change automatically triggers a recomputation of the folds. The view directions are a list of 3D unit vectors on the sphere; the angular strategy projects them onto the longitudinal angle  $\phi$  and sorts, the linear strategy simply does a modulo split over the frame index.

 ПРОСТЫМИ СЛОВАМИ

Angular for evenly distributed orbit shots (default, safe), Linear only if your images don't cluster spatially.


**W25 Slider «k Folds»**

Left sidebar, in the middle..



Slider from 3 to 10, step size 1. On change, the fold computation is automatically restarted so that the folds list, the training/test indices and the per-fold score are immediately recomputed. The selected value is displayed as monospaced-digit text next to the label on the right.

Rule of thumb:  $k=5$  is standard (gives you 20% test per fold, which is common for cross-validation).  $k=10$  if you have a lot of data and need more folds for statistical significance.  $k=3$  if you have little data.

 ПРОСТЫМИ СЛОВАМИ


How many folds in the split. 5 is the default and fits almost always.

**W26 Picker «Test Fold»**

Left sidebar, below the k slider..



Menu picker. Options are dynamically  $0..<k$ , labels «Fold 1» through «Fold N» (so 1-indexed in the UI, 0-indexed internally). If the previously selected index is  $\geq k$  (e.g. because you reduced  $k$  from 10 to 5), it is automatically reset to 0. The selected test fold is shown in red on the globe, all others in green.

 ПРОСТЫМИ СЛОВАМИ

Which fold is currently the test fold. You can click through and see how each individual fold looks on the globe.

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

ГДЕ

Left sidebar, at the bottom..



ТЕХНИЧЕСКИ

Opens a save dialog with default filename `fold-assignment.json`. After confirmation the Holdout module encodes the current split into a JSON schema (per-frame fold assignment plus strategy meta block). This file can then be passed to the next training run with `--benchmark`, so the same Holdout is used for the final metric evaluation. Write errors are shown as error text; success in green text as «Saved to (filename)».

**ПРОСТЫМИ СЛОВАМИ**

Saves the current train/test split as JSON. You can then pass this file directly to the training so the same test set is used again.

**W28 SCNView (3D Camera Globe)**

ГДЕ

Center panel in the Holdout window..



ТЕХНИЧЕСКИ

SceneKit globe view. The scene consists of: a wireframe sphere (radius 1.0, 36 segments, dark gray), three colored axis stubs (red/green/blue for X/Y/Z, each 1.2 long), and per camera a small marker sphere (radius 0.03) at the corresponding view direction position on the unit sphere (slightly outside so it doesn't disappear INTO the wireframe sphere). The markers are NOT rebuilt on each fold change — rebuild is only needed when the frame list changes (i.e. a new JSON is loaded). Instead, per update an in-place update of the material colors runs: red for test indices, green for training, light gray if neither. So slider ticks stay performant even at  $N > 1000$  cameras.

Camera control is enabled — you can rotate, zoom and pan the globe with the mouse. Lighting makes sure the markers don't look flat. Background is dark gray.

**ПРОСТЫМИ СЛОВАМИ**

The 3D globe with the camera positions. Green = training, red = test, light gray = unassigned (doesn't occur, all cameras belong somewhere). With the mouse you can rotate and zoom the globe.

**W29 FoldCard (tap to select fold)**

Right sidebar, «Angular Correlation» section..



One card view per fold — rounded rectangle with 6 pt radius, padding 10, vertical layout with two rows (top «Fold N» + camera count, bottom «Mean nearest angle:» + value in degrees). Background color conditional: active fold = accent color semi-transparent, inactive = neutral standard material. Tapping selects the fold and the globe recolors live.

The «Mean nearest angle» score is the mean smallest angle per test camera to the nearest training camera (internally computed in radians, displayed in degrees in the UI).

**ПРОСТЫМИ СЛОВАМИ**

One small card per fold on the right with the number of cameras and the average distance to the nearest training camera. Clicking it selects this fold as test.

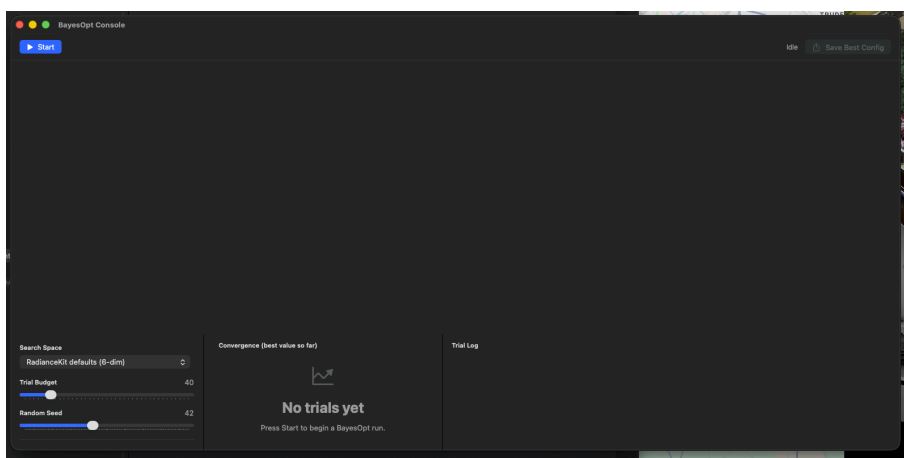
**BayesOpt Console (W30–W39)**

Рисунок 24. BayesOpt console — empty state before trial start

Empty state with search-space picker (RadianceKit defaults (6-dim)), trial budget slider (default 40), random seed (42) and three empty panels for convergence chart, trial log and search-space parameter list.

Empty state (after first open) — convergence chart and trial table fill up as soon as a run is started, see next shot.



Рисунок 25. BayesOpt console after 40 trials — convergence chart rises steeply until trial 15, best value 0.9943, trial log with init/bo/restart tags

**WHAT'S IN THE IMAGE** Status top right «Finished — best 0.9943 after 40 trials». Left sidebar: search-space picker on RadianceKit defaults (6-dim), trial budget 40, random seed 42. Parameter list shows the six hyperparameters to tune with their value ranges: 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]. Center: convergence chart (X = trial index 1–40, Y = objective value 0–1) — gray points = initial samples (LHS), blue points = BayesOpt acquisition, orange points = restart trials (#22 and #31). Best-value line rises steeply up to trial ~7, then only marginal improvement until trial 15, from there a flat plateau at 0.99+. Right sidebar: trial log #1–#34 with score + tag (init/bo/restart). Save Best Config button top right writes `bayesopt-best.json`.


**What it is:** A Bayesian-optimization console for hyperparameter search. Bayesian optimization is an automatic method that tries to find the optimum of an unknown function with as few experiments as possible — typically: «which combination of mcmcMaxGaussians, capMultiplier, ssimWeight and gradThreshold delivers the best PSNR for my scene class?» Instead of a grid of  $6^4 = 1296$  trials, Bayesian optimization tries about 40–100 informed trials and gets close to the optimum that way.

**Important:** The version currently shipped in the app does not run the optimization against real training runs (that would take days) but against a synthetic demo objective — a multi-modal landscape with hill-climbing character plus light noise. This is intentional: the window is meant to show you the behavior of the optimizer (convergence curve, sample points, best-so-far) and let you understand the search-space definitions. For real training-driven BayesOpt runs (as carried out in Phase Q7 for the scene-class presets), a separate offline CLI workflow is used; the window is the live UI variant.

**WHEN TO OPEN** Three use cases: 1. You want to understand how BayesOpt works — then start a demo run and observe the convergence chart. 2. You're planning a new scene class (such as «aquariums» or «antique furniture») for which the built-in 10 presets don't fit perfectly. Mentally define a search space, test it here with the «Bowl demo» or «Densify» preset, then export the best config as JSON and use it as a starting point for a real training run. 3. You want to inspect the default search spaces defined

in the RKBayesOpt package (Mip subset, RadianceKit defaults) — they are listed in the parameter panel of the left sidebar.

**HOW TO INTERPRET** - **Convergence chart** (middle column): Y = best objective function value achieved so far. X = trial index. Initially rises steeply (BayesOpt tries the initial samples randomly, some of them lucky), then flattens out increasingly because the near-optimum region is exhausted. If the line stays flat for 20+ trials, you can stop the run — additional trials won't add anything. The individual points in the chart are the individual trial values (so not «best so far»), colored by phase: gray = initial sample, blue = BayesOpt acquisition, orange = restart. - **Trial table** (right column): #1, #2, #3, ... each with value and phase tag. The best trial so far is marked with a yellow star. From the table you can identify the best trial and inspect its parameter values later during export. - **Search-space inspector** (left sidebar): shows for the selected preset all parameter names and their search ranges `[lo, hi]`. If you're on the preset «RadianceKit defaults (6-dim)», you see e.g. «densifyGradThreshold [5e-7, 5e-6]» — so log-uniform between these two values.

 Pick preset «RadianceKit defaults (6-dim)», trial budget 40, seed 42. Click «Start». Observe: the first 8 trials are gray (initial samples, LHS Latin hypercube), the following ones blue (BayesOpt-acquired). The convergence chart rises steeply up to trial ~15, after that it flattens out. At trial ~30–40 the best value stabilizes. Click «Save Best Config» — a `bayesopt-best.json` is saved with the preset name, trial index, value, and the decoded parameter values. You can then manually copy this JSON into your preset definition.

### Button «Start»

 ГДЕ

Toolbar on the left, in idle/finished state..

 ТЕХНИЧЕСКИ

Resets the trial list, switches into running state, generates a new run ID (for stale detection on multiple Start clicks) and creates a fresh pause gate. Then a background task starts that runs the optimizer as an asynchronous stream. Initial-samples size is  $\min(8, \text{budget} / 4 + 1)$  — so typically 8 Latin-hypercube samples at budget  $\geq 28$ , fewer at small budget. Trial updates are received incrementally and appended to the list. Stale-run protection: if in the meantime a second Start click sets a new run ID, updates from the old run are discarded.

Primary action style for the prominent button look.

#### ПРОСТЫМИ СЛОВАМИ

Starts a fresh optimization run with the current search space, budget and seed.

**W31 Button «Pause»**

Toolbar on the left, in running state..



Activates the pause gate and switches into paused state. The actual effect: the runner waits in a 50 ms polling loop before it evaluates the next objective function. This means a trial currently running is run to completion (it's synthetic and only takes microseconds), but no further trial is started. As soon as Resume runs, it continues where it left off.

 ПРОСТЫМИ СЛОВАМИ


Pauses the run. The current computation still runs to completion, then it pauses.

**W32 Button «Stop»**

Toolbar on the left, in running and paused state..



Cancels the runner task, nulls the reference, releases the pause gate (if still paused) and switches into finished state (if trials exist) or idle state (if not). The already computed trials remain visible in the list — Stop does not delete them. Destructive button role shows the button in red because it cancels the run.

 ПРОСТЫМИ СЛОВАМИ

Cancels the run permanently. Trials stay visible; you can still export the best config.

**W33 Button «Resume»**

Toolbar on the left, in paused state..



Releases the pause gate and switches back to running state. The runner task is already running (it's waiting in the polling loop); as soon as the loop notices that the pause has been lifted, it continues and starts the next trial.

 ПРОСТЫМИ СЛОВАМИ

Resumes a paused run.

**W34 Button «Save Best Config»**

ГДЕ

Toolbar on the right, always visible (but disabled if no bestTrial exists)..



ТЕХНИЧЕСКИ

Opens a save dialog with default filename `bayesopt-best.json`, restricted to JSON. After confirmation a payload dictionary is built: preset name, trial index, value (objective score), parameters (dictionary of decoded parameter names → values). The decoding projects the normalized search-space coordinates in  $[0,1]^d$  back into the original value range (with log-uniform/linear/integer scales accordingly). JSON output is pretty-printed and with sorted keys. On write errors (in the current demo version) is silently ignored — no error UI because it's a demo path.

The button stays gray as long as no trial has run.

**ПРОСТЫМИ СЛОВАМИ**

Saves the parameter values of the best trial so far as JSON. You can then manually copy these values into your preset configuration.

**W35 Picker «Search Space» preset**

ГДЕ

Left sidebar, at the top..



ТЕХНИЧЕСКИ

Menu picker with four preset options: - «RadianceKit defaults (6-dim)» — the full standard search space with all Q7 hyperparameters. - «Mip subset (2-dim)» — only `mipSmoothing3DScale` [0.05, 0.5] log-uniform and `mipFilter2DVariance` [0.1, 0.6] linear. Useful when you want to tune Mip-Splatting for a scene class. - «densify-until + ssim-weight + grad-thresh» — three Densify-relevant parameters (`densifyGradThreshold` log-uniform, `ssimWeight` linear, `densifyUntilIter` integer). - «Bowl demo (1-dim)» — pedagogical single-parameter search space for «this is how BayesOpt works» demos.

While a run is active, the search space cannot be switched (would confuse the optimizer).

**ПРОСТЫМИ СЛОВАМИ**

Which hyperparameter search space BayesOpt explores. Default is «RadianceKit defaults». For targeted Mip-tuning attempts «Mip subset». To understand how BayesOpt works «Bowl demo».

**W36 Slider «Trial Budget»**

Left sidebar, below the search-space picker..



Slider from 10 to 200, step size 5. Default 40. This means: BayesOpt may do a maximum of N trials. Of these the

first few are initial samples (Latin hypercube), the rest are real BayesOpt trials. Rules of thumb for practice: a search space with  $d$  dimensions needs about  $10d$  to  $20d$  trials for a good optimum. At 6-dim defaults that's 60–120, at 2-dim Mip subset 20–40, at 1-dim Bowl demo 10–20.

During the run the slider is disabled.

 ПРОСТЫМИ СЛОВАМИ

How many optimization attempts at most. More attempts = better solution, but costs more time. 40 is a good default for the demo objective.


**W37 Slider «Random Seed»**

Left sidebar, below the budget slider..



Slider from 1 to 100, step size 1. Default 42. The seed is passed both to the initial Latin-hypercube samples

and to the noise component of the demo objective. Reproducibility: same seed + same search space + same budget yields exactly the same trial sequence. Useful for «do all your colleagues get the same run when they rebuild the demo?». Disabled during the run.

 ПРОСТЫМИ СЛОВАМИ

Controls the random generator. Same seed = same run — for reproducing.

**W38 Chart (Convergence)**

ГДЕ

Middle column of the window..

**ТЕХНИЧЕСКИ**

Swift Charts diagram with two layers: 1. a line for «best-value-so-far» per trial — a monotonically rising or constant curve in accent color. 2. one point per trial with the individual objective value, colored by phase. Symbol size 40. Three phase labels: «init» (gray), «bo» (blue), «restart» (orange).

A small legend shows the phase colors at the top left. If the trial list is empty (before the first start), an empty-state display with a chart icon and the hint «Press Start to begin a BayesOpt run.» is displayed instead.

**ПРОСТЫМИ СЛОВАМИ**

The progress chart. The solid line is «best solution found so far»; the points are the individual attempts. If the line stays flat for a long time, BayesOpt has found the optimum.

**W39 Table (Trial Log)**

ГДЕ

Right column of the window..

**ТЕХНИЧЕСКИ**

Scroll area with lazily stacked trial rows. One horizontal stack per row: trial number (3-digit monospaced, on the left), value (monospaced, right-aligned, 70 pt wide), phase tag (capsule, filled with phase color at 25% opacity), optionally a yellow star if this trial is currently the best. An auto-scroll mechanism automatically jumps to the end as soon as a new trial is added — so you can follow the live progression at the bottom of the screen without scrolling yourself.

**ПРОСТЫМИ СЛОВАМИ**

The table of all attempts. Value, phase, star for the best. Auto-scrolls along, new trials appear at the bottom.

**Main Window: Loss Curve and Gaussian Count (I39–I41, cross-reference)**

Three of the Inspector displays in the main window deserve their own explanation because they are constantly visible during a running training and there are important rules of thumb for when the curve looks healthy. The displays are in the Inspector under the «Loss Chart» section (see Chapter 2 — Inspector) and complement the Holdout analysis from the auxiliary window above.

**When is the loss curve healthy?** A healthy loss curve shows three phases: (1) **Warmup** — the first 200–500 iterations the loss falls steeply from high (typically 0.15–0.25 for L1+SSIM combined depending on the scene) to about half. If the loss does NOT fall in this phase, the input is usually wrong (broken images, bad SfM poses, too few initial Gaussians). (2) **Densification** — between ~500 and `densifyUntilIteration` (classic 15K, MCMC up to 20K or 25K) the loss continues to fall, often with small jumps downward when densify operations insert new Gaussians and the optimizer exploits them. The Gaussian count rises in this phase. (3) **Refinement** — after that the loss runs into a tail that flattens out. Typical end values: Tanks-&-Temples Truck with P4 Quality lands at  $L1 \approx 0.023$ , Horse with Full Classic V546 at  $L1 \approx 0.0230$ , outdoor Mip-NeRF360 scenes often worse (0.04–0.07).

**What does a plateau mean?** A plateau (loss curve runs horizontally over several thousand iterations) has two interpretations: (a) the model has converged, more training won't add anything — the good case. (b) the model is stuck (local minimum, bad gradient information, a cap at the buffer limit) — the bad case. Both look identical in the chart. Distinction: look at the Gaussian count. If it's also flat AND close to the MCMC cap (e.g. 150K of 150K at `.fullMCMC`), you are at the limit — either raise the cap or accept the plateau. If the Gaussian count is still growing but the loss isn't falling, it's stuck.

**When to abort vs continue training?** Rule of thumb: 10K iterations long no improvement of the min loss → abort, further iterations are wasted. Before that: you can append an extension via `Cmd+T` (Training menu → Continue Training → +5K iterations), if you see marginal improvement. Watch out: with MCMC the plateau is often real — the cap is the natural limit.

**Gaussian count plateau is NOT a «done» signal.** It only means that MCMC has reached the cap or that Classic Densification is exhausted. The real «done» question is answered only by the Holdout analysis — PSNR/SSIM/LPIPS on an independent test set, evaluated in the Holdout window (W23–W29) or via the `--benchmark` flag.

**PSNR/Holdout is the truth, loss is only a proxy.** Loss is a relative metric: it falls as your model fits the training views. A low loss does not automatically mean a good model — if the model has memorized the training images (overfitting), the loss would be small, but PSNR on unseen views (Holdout) would be bad. Therefore: for final quality assessment always look at Holdout metrics, not end-loss alone.

## Rule-of-Thumb Box

- User Guide and Keyboard Shortcuts are static help — for keyword questions fast, for depth use this manual at hand.
- Open Manage Storage as soon as the disk falls below 10% free space. Logs and imports staging are the usual culprits.
- Pareto Dashboard only useful after at least three or four training reports. X-axis = costs (Time / Gs), Y-axis = quality (PSNR / SSIM). The Pareto front shows the efficient combinations.

- Use Holdout Analysis before publishing PSNR benchmarks with others — it assures you that your test set is really representative.
- BayesOpt Console is primarily a learning and inspection tool for search-space definitions. For real training-driven hyperparameter tuning use the offline CLI workflow.
- Loss plateau and Gaussian-count plateau are to be interpreted separately. Cap limit is not a «done» signal. Real quality is measured only by Holdout PSNR.
- 10K iterations without min-loss improvement → stop training.

## ГЛАВА

## Глава 6 — Конфигурация обучения

```
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
  }
}
```

Рисунок 26. Preview preset exported as JSON and displayed in TextEdit — fields `id/` `name/` `category/` `version/` `createdAt/` `description`, `trainingConfig` with all relevant parameters (`maxIterations` 5000, `densifyUntilIteration` 3500, `ssimWeight` 0.20, `renderScale` 0.50, `strategy` `classic`, `cameraAlignment` `applePhotogrammetry`, `densifyGradThreshold` `2.0e-06`, `opacityResetInterval` 3000, `minOpacity` 0.005, six boolean toggles)

**WHAT YOU SEE IN THE IMAGE** A typical preset JSON export. Top-level fields: `id` (UUID), `name`, (`classic` | `mcmc` | `sceneClass` | `custom`), (`schema version`), (`timestamp`), (`free text`). The nested object holds the parameters that are critical for reproducibility — on import the entire block is deserialized into the `TrainingConfig` struct, and defaults from the current app version fill in any fields missing from the JSON (e.g. after an app update). To hand a preset over to another Mac, you just ship this JSON file.

The `TrainingConfig` struct is the heart of every training run in RadianceKit. It collects every parameter that influences training — from maximum iteration count over the eight learning rates to the special fields for MCMC, Mip-Splatting, the curriculum and the scene-aware cap logic. You edit it in the sidebar in the Training Configuration section (Expert View), save it as a preset or hand it over as a JSON export to another Mac. At training start this very object is frozen and handed to the GPU backend.

This chapter is reference material for power users and script authors. It lists all 81 public fields, the 9 static presets and the one public method. The source file is `TrainingConfig.swift` — when in doubt the doc comment stored there and the initializer default are the source of truth.

**NOTE · UI VS. PRESET/CLI**

Only 12 of the 81 fields have a direct slider, toggle or picker in the Inspector (sandboxed App Store build): **T1, T2, T17, T20, T22, T38, T56–T58, T60, T61, T73**. The remaining 69 fields are set via the chosen **preset** (Chapter 7) and can only be overridden directly via a **CLI flag** (see Chapter 5). This separation is by design: defaults stay stable and production-tested, while power users still have an escape hatch. If a field really interests you, first check Chapter 2 (Inspector) and Chapter 5 (CLI) to see whether you can reach it without JSON tinkering.

**Table of contents:**

1. Iteration (T1–T2)
2. Learning Rates (T3–T10)
3. Densification — Classic (T11–T16)
4. Loss (T17–T20)
5. SH Degree Progression (T21)
6. Performance (T22–T25)
7. Diagnostics and Point Cloud Preparation (T26–T30)
8. Regularization (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. MCMC Densification (T61–T73)
14. Mip-Splatting (Q1.5) (T74–T76)
15. Adaptive Densification (Q5) (T77–T79)
16. Curriculum (Q6) (T80–T81)
17. Static Presets (TP1–TP9)
18. Method:
19. Which field for what? (Cheat Sheet)
20. Dangerous Fields

## Iteration (T1–T2)

### T1 maxIterations

#### ПОДРОБНОСТИ

**Default:** 30 000 (initializer), 35 000 ( `.full` ), 200 000 ( `.fullMCMC` ) **Range:** 1 000 – 500 000 (UI slider), no hard upper limit in the logic **Defined in:**

#### ТЕХНИЧЕСКИ

Total number of training iterations the backend runs through. One iteration means a forward render of a single training camera, one backward pass over all loss components (L1 + SSIM + optional regularizations + sky mask) and one Adam optimizer step. This number directly drives the other schedules: position learning rate follows a cosine annealing curve from 0 to either `T1` itself or to `T49 positionLRScheduleEndIteration`; densification stops at `T2 densifyUntilIteration`; MCMC noise decay ends at `T69 mcmcNoiseDecayEnd`; SH degree upgrades happen at the three marks in `T21`.

For classic densification the empirically determined sweet spot is at 20 000–35 000 iterations (Sessions 1–32, V546 tests), for MCMC at 60 000–200 000 (V534). Pushing well beyond the values stored in the preset rarely brings additional quality — Adam momentum saturates, and without an LR decay end the loss stagnates. Conversely, going below ~5 000 leads to incompletely converged geometry (density control has too little time to clone/split).

#### ПРОСТЫМИ СЛОВАМИ

How long the app trains. More iterations = better result, but eventually no longer noticeably better, just much longer. The presets are chosen so that without thinking you get a good value: Quick 1 000, Preview 5 000, Balanced 20 000, Quality 35 000, MCMC Quality 200 000. If you turn the knob yourself: with MCMC feel free to go high (100 000–200 000), with Classic not above 40 000 — doesn't help any more after that.

## **T2** densifyUntilIteration

### ПОДРОБНОСТИ

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

### ТЕХНИЧЕСКИ

Iteration at which densification stops. Up to this point Gaussians are cloned, split and pruned according to the rules parameterized in `T11–T16` (Classic) or `T67–T70` (MCMC); after that the Gaussian count stays constant and only positions, rotations, scales, opacities and SH coefficients are optimized (refinement phase). In the 3DGS original paper the value sits at 50 % of `T1`, in RadianceKit's `.full` preset at only ~14 % (5 000 of 35 000) — a consequence of the V310/V338 experiments which showed that after 5 000 iterations further densification makes the result worse (more floaters, higher memory use, no quality gain). MCMC, on the other hand, runs relocation up to 80 % of `T1` (V504b) because MCMC does not produce harmful floaters. If `T2` is chosen too small (< 1 000), too few Gaussians arise; too large under Classic (> 50 % of `T1`) leads to overgrowth and RGB saturation outliers (see Outdoor Overtraining Findings).

### ПРОСТЫМИ СЛОВАМИ

Up to when the app is allowed to create new Gaussians. After that, only what already exists is refined. With classic training at 35 000 iterations, 5 000 is the right value here — anything above makes the scene mushier. With MCMC it is 80 % of total iterations (so 160 000 for a 200 000 run). If you change the Quality preset, better leave this field alone.

## Learning Rates (T3–T10)

### T3 positionLearningRate

#### ПОДРОБНОСТИ

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

#### ТЕХНИЧЕСКИ

Adam learning rate for each Gaussian's XYZ position at the start of training (iteration 0). It follows a cosine annealing curve and decays over training down to T4 positionLearningRateFinal. The default 0.00016 comes from the 3DGS original paper (Kerbl et al. 2023) and in RadianceKit is not to be scaled even with higher image resolution — position moves in world coordinates, not in pixel space. A clear increase ( $> 0.0005$ ) makes Gaussians jump over long distances and the loss becomes unstable; values well below ( $< 0.00005$ ) mean badly initialized point clouds never find their place. V414 tested doubling the initial value  $\rightarrow$  16.8 % worse L1 loss; the V544a tunings confirmed the paper default as optimal. Note: under `.fullMCMC` we deliberately leave this at the default — MCMC needs constant learning rates for its relocation logic, so tuning here brings nothing.

#### ПРОСТЫМИ СЛОВАМИ

How fast the splat points are allowed to move through space. The default value is very well tuned and basically doesn't need changing. Only when splats «wobble» in the image or a whole corner is missing because nothing moves there would the learning rate be something to adjust — but then typically something else is wrong upstream (camera poses, initial point cloud).

## T4 positionLearningRateFinal

### ПОДРОБНОСТИ

**Default:** 0.0000016 (initializer + paper), 0.000016 (`.full`, `.fullMCMC` — 10× higher) **Range:** 0 – **Defined in:**

### ТЕХНИЧЕСКИ

End value of the position LR cosine annealing curve. It is reached either at `T1 maxIterations` or, if set, at `T49 positionLRScheduleEndIteration`. The RadianceKit `.full` preset uses 0.000016 — i.e. 10× higher than the paper default 0.0000016. V420 experiments showed that 0.5× of the final value (0.000008) makes loss 6.4 % worse; V414 showed that 2× the initial value makes it 16.8 % worse. The high final value is not a trade-off but a deliberate choice: with too aggressive a decay, Gaussians lose during the refinement phase the ability to react to newly arrived densification candidates. Through the V431/V433 extension the schedule phase can be shortened (`T49 < T1`), so that `T4` is reached before the end of training and the rest of training runs at the constant mini-LR — typical configuration: `T49 = 20 000`, `T1 = 35 000`, refinement thus at 0.000016 for 15 000 iterations.

### ПРОСТЫМИ СЛОВАМИ


How slow the position learning rate becomes at the end of training. We've intentionally set this less aggressively than the original paper — splats can still wiggle a bit until the very end, which makes them sharper. If you turn the knob: higher = more restless splats at the end, lower = splats can no longer adapt when new ones appear.

**T5** shDCLearningRate **ПОДРОБНОСТИ**

**Default:** 0.0025 (initializer + paper), 0.005 ( `.full` and all MCMC presets — 2×) **Range:** 0.0001 – 0.05

**Defined in:** **ТЕХНИЧЕСКИ**

Adam learning rate for the DC component (degree 0, i.e. constant albedo) of the spherical harmonic color. SH-DC corresponds to the direction-independent base tone of a Gaussian, essentially the «base color». V176 and V188 experiments found 2× higher than the paper default to be optimal — faster color convergence, especially because with short training (, 5 000 iterations) the SH-DC otherwise does not converge. Unlike the geometric LR, SH-DC has no decay; the learning rate stays constant over all iterations (or just follows the optional extended-phase decay from T51 ). V416 tested a quadrupling to 0.01 → 6.4 % worse loss with beta2=0.99 Adam.

 **ПРОСТЫМИ СЛОВАМИ**


How fast the base color of each splat adapts. You almost never change this value yourself — the presets have the right value. Higher would be faster but can lead to unstable colors.

**T6** shRestLearningRate **ПОДРОБНОСТИ**

**Default:** 0.000125 (initializer + paper), 0.00025 (.full and MCMC — 2×) **Range:** 0.000001 – 0.005 **Defined in:**

 **ТЕХНИЧЕСКИ**

Adam learning rate for the higher-order SH coefficients (degree 1, 2, 3 — i.e. the view-direction-dependent color components that produce highlights, reflections and soft shading). 20× smaller than T5 per paper convention, because these coefficients grow quadratically in count (3 for degree 1, 5 for degree 2, 7 for degree 3 → 15 floats total per Gaussian) and without a smaller learning rate would oversaturate the image. They are unlocked in two steps — until the first mark in T21 shDegreeUpgradeIterations only degree 0 is active (so only T5), after that 1, then 2, finally 3. Low values here are particularly important on scenes with lots of diffuse lighting; on very glossy surfaces (car paint, water) tuning doesn't help — the SH representation itself is limited.

 **ПРОСТЫМИ СЛОВАМИ**


How fast the view-dependent color effects (reflections, gloss) learn. By default very small, otherwise everything starts to shine. Better leave the value alone — if you want better highlights, you're better served by MCMC and longer training time than by this LR.

**T7** **opacityLearningRate** **ПОДРОБНОСТИ**

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

 **ТЕХНИЧЕСКИ**

Adam learning rate for the logit opacity of each Gaussian. The app stores opacity as an unbounded float value and transforms it with sigmoid into  $[0, 1]$ ; the LR acts in logit space. The paper default 0.05 was restored after V50 tests (best single-run L1 0.1664), V71 reverted V67's 0.025. The V188 doubling to 0.1 makes pruning more efficient — dead Gaussians fall faster below the T14 `pruneOpacityThreshold`. V418 showed: 0.05 with `beta2=0.99` Adam is 7.1 % worse than 0.1 — the interaction with the Adam configuration is non-trivial. Low values ( $< 0.01$ ) mean «dead» Gaussians linger forever and waste memory; too-high values ( $> 0.5$ ) can lead to opacity explosion, which is why the logit value is clamped in the optimizer to  $[-15, 3]$  (see the «Opacity Explosion Prevention» note in CLAUDE.md).

 **ПРОСТЫМИ СЛОВАМИ**

How fast splats become transparent or opaque. Important for cleanup — splats that contribute nothing have to vanish quickly so no haze appears. The default value fits, only pros change it.

**T8** `opacityLearningRateFinal` **ПОДРОБНОСТИ**

**Default:** 0.0 (= «no decay») **Range:** 0 or 0.001 –  
**Defined in:**

 **ТЕХНИЧЕСКИ**

Optional cosine decay end value for the opacity LR (V427). When 0.0, decay is disabled and the opacity LR stays constant at `T7` over the entire training. V427 tested a decay 0.1 → 0.01 — result was 11.5 % worse loss; reverted, hence the default «off». The hypothesis behind the field: in the refinement phase, a constant opacity LR could lead to oscillation, so that splats which had already reached the right level of transparency would be pushed around again by random gradient fluctuations. Empirically this is not the case — the logit clamping logic catches that anyway. The field stays available for future experiments; very long MCMC runs (> 500K iterations) might also benefit from it.

 **ПРОСТЫМИ СЛОВАМИ**

Whether the opacity learning rate should become smaller toward the end. Default: no. We tried it, it was worse, we leave it off. Stay at 0.

**T9** `scaleLearningRate` **ПОДРОБНОСТИ**

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

 **ТЕХНИЧЕСКИ**

Adam learning rate for the three scale components of each Gaussian in log space (RadianceKit stores  $\log(\text{scale})$  so that scales stay positive). The paper default 0.005, doubled in RadianceKit to 0.01 for better scale convergence with the optimized learning rate configurations. V423 experiment: 0.005 with  $\beta_2=0.99$  Adam → 18.7 % worse loss and visibly too few Gaussians (density control couldn't clone because scale updates were too sluggish). Scale controls the extent of each Gaussian — too fast learning leads to «needle» Gaussians (extremely long thin splats, see `T34 scaleRatioPruneThreshold`), too slow learning keeps splats too compact and density control has to split too often.

 **ПРОСТЫМИ СЛОВАМИ**


How fast the shape of the splats adapts. Default is good. If you push this up you get «needle» splats — extremely long thin droplets that make the image float.

**T10 rotationLearningRate** **ПОДРОБНОСТИ**

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

 **ТЕХНИЧЕСКИ**

Adam learning rate for the four quaternion components of each Gaussian. The quaternion is re-normalized after each Adam update (L2 norm = 1) — otherwise the covariance matrix would degenerate. RadianceKit doubles the paper default in the Quality presets because rotation has smaller absolute gradient magnitudes than scale/position (on the unit sphere each step stays short) and without 2× rotation would be clearly under-converged in the 35 000 iteration window. V188 documents this. On NeRF-Blender scenes (Lego, Chair) rotation matters particularly — the edges of objects only line up properly after 5 000–10 000 iterations.

 **ПРОСТЫМИ СЛОВАМИ**

How fast the splats learn to rotate — that is, to orient themselves correctly on the surface of an object. Default fits. Put another way: if splats look like slabs lying askew instead of hugging the surface, training time is too short rather than this learning rate too low.

## Densification — Classic (T11–T16)

### T11 densifyGradThreshold

#### ПОДРОБНОСТИ

**Default:** 0.000002 (initializer, calibrated for 0.5x resolution), 0.0000011 ( `.full` , calibrated for 1.0x), 0.000004 ( `.quickTest` , calibrated for 0.25x),  $2e-7$  ( `.fullClassicPaper` ) **Range:**  $1e-8 - 1e-3$  (resolution-dependent) **Defined in:**

#### ТЕХНИЧЕСКИ


Threshold for the L2 norm of the screen-space projected gradient `dMean2D` , above which a Gaussian is marked for cloning or splitting. The absolute value depends directly on the training resolution — `dMean2D` scales roughly as  $1/\text{resolution}^2$  (more pixels = smaller per-pixel gradients). Hence every T22 `trainingRenderScale` step needs a calibrated threshold:  $0.25x \rightarrow 4e-6$ ,  $0.5x \rightarrow 2e-6$ ,  $1.0x \rightarrow 5e-8 \dots 1.1e-6$  ( `.full` ). The paper default 0.0002 is NDC-normalized and not directly comparable to RadianceKit's world-space pipeline. With the V440-introduced T52 `adaptiveDensifyThreshold` flag the value can be derived at runtime from the p98 of the current gradient distribution — but V440 tested this on real scenes and produced 63 K Gaussians (catastrophic pruning loss); the flag stays off. Q5 (T77–T79) provides an alternative adaptive logic via rolling median. **This field is not harmless** — halving creates 2–4x more Gaussians (memory pressure, OOM risk); doubling can under-densify the scene.

#### ПРОСТЫМИ СЛОВАМИ

How sensitive the app is when deciding whether a splat is under-represented and should be duplicated. Lower value = more sensitive = more splats. Higher = fewer splats. This is one of the most dangerous values of all: too low and your Mac fills up with millions of splats and may crash. Leave the field alone, or change it only in steps of 10 %.

**T12** densifyFromIteration **ПОДРОБНОСТИ****Default:** 500 **Range:** 100 – 5 000 **Defined in:** **ТЕХНИЧЕСКИ**


First iteration at which densification becomes active. Before that only «bare» learning on the initial SfM point cloud happens, without new Gaussians being created. The default 500 comes from the 3DGS paper and gives initialization time to stabilize — if densification starts as early as iteration 0, mis-positioned SfM points are cloned many times before they even find their proper place. V349 tested 1000 → slightly worse loss; the default is optimal.

 **ПРОСТЫМИ СЛОВАМИ**

When the app first starts cloning splats. Before that it just learns the already existing points. 500 is the default — gives the app enough time to find its bearings first before multiplying.

**T13** densifyInterval **ПОДРОБНОСТИ****Default:** 100 (initializer, MCMC), 200 ( `.full` )**Range:** 50 – 1 000 **Defined in:** **ТЕХНИЧЕСКИ**

How many iterations sit between two densification steps. Paper default 100 — every 100 iterations the list of densify candidates is evaluated, cloned/split, and at the same time the list of prune candidates ( $\text{sigmoid}(\text{opacity}) < \text{T14 } \text{pruneOpacityThreshold}$ ) is removed. V112 tests found 200 to be optimal for `.full` — this relieves the GPU because fewer reorganization passes run, and gives each Gaussian more time to settle after a clone action. V417 tested 100 with  $\text{beta2}=0.99$  → 5.8 % worse (957 K Gaussians, over-densification). Under MCMC the same field is interpreted as the relocation interval; see T67 `mcmcRelocationInterval` for the MCMC-specific logic.

 **ПРОСТЫМИ СЛОВАМИ**


How often the app looks for new splats. 100 = often, 200 = medium. Higher means: each splat has more time to settle before being multiplied again. That's good. Reducing to 50 can keep the GPU busy permanently without it being noticeably better.

**T14** `pruneOpacityThreshold` **ПОДРОБНОСТИ**

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

 **ТЕХНИЧЕСКИ**

Sigmoid opacity threshold below which a Gaussian is deleted at the next densification step. Works together with `T7 opacityLearningRate` and the logit clamp logic in the optimizer. V393 lowered the default from 0.005 to 0.001 in `.full` — result: splats that only matter under exotic viewing angles are kept longer and contribute to SH detail. V394 tested 0.0001 → slightly worse (too little pruned, memory wasted). Important: density control MUST always prune, even if buffer capacity is already full due to other measures (see «Density Control Must Always Prune» in CLAUDE.md) — otherwise dead Gaussians accumulate and the count freezes.

 **ПРОСТЫМИ СЛОВАМИ**

When a splat counts as «transparent enough» to be deleted. 0.005 is the paper default, in Quality we have 0.001 — i.e. we give splats a chance for longer. That makes soft light and faint shadows easier to represent. Setting higher (above 0.01) makes the splat count drop quickly — can make sense with memory pressure but costs detail.

**T15** `opacityResetInterval` **ПОДРОБНОСТИ**


**Default:** 3 000 (initializer + paper), 100 000 (`.full` = effectively disabled), 200 000 (`.fullMCMC` = disabled) **Range:** 1 000 – 100 000+ **Defined in:**

 **ТЕХНИЧЕСКИ**

How many iterations between resetting the opacity of all Gaussians to a low value (~0.01) — a measure from the 3DGS paper to reassess «frozen» splats. V194 showed that with RadianceKit's warmup

1. stochastic training setup + 2× learning rates, opacity reset costs

5.5 % quality and that logit clamping already covers the reset function. Hence in `.full` practically disabled (100 000 > 35 000, so never triggered). V421 tested reset every 3 000 with `beta2=0.99` → 4.9 % worse; reverted. Under `.fullClassicPaper` (Q1.5-A, paper-true test) it is deliberately set back to 3 000 — that was one of the levers with which the paper-magnitude Gaussian budgets were to be reached.

 **ПРОСТЫМИ СЛОВАМИ**

How many iterations between the app resetting the visibility of all splats to «almost invisible» — a kind of reset button for opacity. With us disabled (value so high that it never fires) because other mechanisms make it unnecessary. Only enable for paper-faithful experiments.


**T16** maxScreenSize **ПОДРОБНОСТИ**

**Default:** 0.0 (= disabled) **Range:** 0 (off) or > 0

**Defined in:**

 **ТЕХНИЧЕСКИ**

Maximum screen-space size (in projected pixels) a Gaussian may reach before being forcibly split. The value is set to 0 (V48 tested and reverted) — RadianceKit's density control instead uses the world-space scale threshold from the `dMean2D` logic. Stays in the field catalog because future experiments with Mip-Splatting (T74–T76) or scene-specific splatting strategies could benefit from it. Enabling (value > 0, e.g. 20) would force splats that have grown very large on screen to subdivide — relevant with large smooth wall surfaces where a single giant splat offers too little detail.

 **ПРОСТЫМИ СЛОВАМИ**

Limit on how large a single splat is allowed to get on screen. With us off. Turning it on would cause huge flat splats (e.g. on a wall) to be forcibly broken up into several smaller ones. Leave off unless explicitly experimenting with it.

## Loss (T17–T20)

### T17 ssimWeight

#### ПОДРОБНОСТИ

**Default:** 0.2 (initializer + paper + `.full`), 0.05 (all MCMC presets) **Range:** 0.0 – 1.0 **Defined in:**

#### ТЕХНИЧЕСКИ

Weight of the D-SSIM term in the combined loss function  $loss = (1 - \lambda) * L1 + \lambda * D-SSIM$ , where  $\lambda = T17$ . The 3DGS paper default 0.2 is optimal for classic densification — V383 tested 0.3 → 28.9 % worse, V373b confirmed 0.2 as the sweet spot. For MCMC it was independently established in V521b/V534: 0.05 is optimal because MCMC, through its stochastic exploration, needs a stronger L1 signal component — higher SSIM weights would dilute the relocation decisions. SSIM is significantly more expensive to compute than L1 (local 11×11 windows over the whole image); RadianceKit uses an MPS-accelerated implementation that stays under 1 ms per 1080p image. Q7 BayesOpt sweeps found scene-specific optima between 0.05 (`.outdoorPreset` : 0.082) and 0.171 (`.indoorPreset`).

#### ПРОСТЫМИ СЛОВАМИ

How important the app considers «structures are similar» alongside «every pixel matches». 0.2 is the default and gives a good image. Lower = more pixel-accurate but can result in softer transitions. Higher = more structurally similar but details get softer. Let the presets decide.


**T18** **ssimWeightRefinement** **ПОДРОБНОСТИ**

**Default:** 0.0 (= «no switch, keep ssimWeight»)

**Range:** 0 or 0 – 1.0 **Defined in:**

 **ТЕХНИЧЕСКИ**

Optional SSIM value for the refinement phase after T2 densifyUntilIteration. V428 tested 0.2 → 0.3 in refinement → 16 % worse loss (both L1 and SSIM degraded); reverted, hence default 0.0. The hypothesis behind the field was that after densification — when no more new Gaussians are being created — a stronger SSIM share would maximize structural sharpness. Empirically false: increasing SSIM weight means indirectly lowering L1 weight, and L1 is the much more meaningful signal in the final refinement phase. The field stays available for future experiments with perceptual loss (T60) or edge loss (T19), where a refinement-specific loss composition might make sense.

 **ПРОСТЫМИ СЛОВАМИ**

Special setting for the second training phase (refinement after splat duplication). At 0.0: same SSIM weighting as before. Tweaking brings nothing empirically, so off.

**T19** **edgeLossWeight** **ПОДРОБНОСТИ**

**Default:** 0.0 (= disabled) **Range:** 0 or 0.001 – 1.0

**Defined in:**

 **ТЕХНИЧЕСКИ**

V437 experimental loss: weight of a Sobel-gradient-domain L1 loss that compares image edges directly (ground-truth Sobel vs render Sobel) on top of L1+SSIM. Hypothesis: edge information is a perceptual cornerstone of image quality and an explicit term should encourage Gaussians to hit edges better. Test results: weight 0.1 → 11 % worse loss, 0.01 → quality-neutral but 10 % slower. The Sobel pass costs an extra MPS forward on ground truth and render. Hence permanently disabled. Future use case: scenes with hard artificial edges (architecture, furniture, renderings) could benefit — Q7 scene class presets didn't pick it though, instead they scaled the SSIM weight.

 **ПРОСТЫМИ СЛОВАМИ**

Experimental extra that treats edges as particularly important. Brings nothing empirically. Stays off.

**T20** skyMaskingEnabled **ПОДРОБНОСТИ**

**Default:** false (initializer and all presets) **Range:** boolean **Defined in:**

 **ТЕХНИЧЕСКИ**

Enables sky masking. In every image the sky region is masked out via Apple Vision Framework (`VNGenerateForegroundInstanceMaskRequest`), and the loss in this region is set to zero. Reason: outdoor scenes often suffer from blue/gray/white sky pixels driving the app to place Gaussians exactly there — which is perceived as «floaters». Without a sky mask the loss in this region would never be zero because the sky in the image varies slightly and the app keeps trying to rebuild it with splats. The Vision mask is computed once per camera before training and held in RAM. Typically activated together with `T45 skyDomeEnabled` (UI logic in the Settings view). Leave disabled for indoor scenes or synthetic renderings — the mask would erroneously detect ceilings or walls as «sky».

 **ПРОСТЫМИ СЛОВАМИ**

Enables a special mode for outdoor captures: the sky is ignored during training, so the app doesn't try to rebuild it with splats. Recommended for every outdoor scene. Leave off for indoor or for 3D renderings from Blender.

## SH Degree Progression (T21)

### T21 shDegreeUpgraderIterations

#### ПОДРОБНОСТИ

**Default:** [1\_000, 2\_000, 3\_000] (initializer), [2\_000, 5\_000, 8\_000] ( .full , MCMC), [1\_000, 2\_000] ( .preview — degree 3 skipped) **Range:** [Int] , each value in [0, maxIterations] , monotonically increasing  
**Defined in:**

#### ТЕХНИЧЕСКИ

Iterations at which the active SH degree is upgraded 0→1, 1→2, 2→3. Before the first mark only the DC components are active (i.e. T5 shDCLearningRate), after the first mark DC + 3 degree-1 coefficients, after the second mark + 5 degree-2 coefficients, after the third mark all 15 coefficients. Memory consumption per Gaussian grows in steps accordingly — 4 floats → 16 floats → 36 floats → 64 floats. The Quality presets delay the upgrades compared to initializer defaults (V228) because the geometry should stabilize first, before the color details with their higher frequency are added on top. V384 tested [1K, 2K, 3K] for .full → 9.3 % worse — confirms the delay. .preview caps at degree 2, because degree 3 doesn't converge in 5 000 iterations and just consumes optimizer capacity. Q6 (T80–T81) offers an alternative curriculum logic that dynamically overrides this list.

#### ПРОСТЫМИ СЛОВАМИ

At which points during training the app learns that colors can look different from different viewing angles (highlights, reflections). Only late — so first the shape is right, then the color. The values in the presets are set so this works well. Don't change anything unless you know exactly why.

## Performance (T22–T25)

### T22 trainingRenderScale

#### ПОДРОБНОСТИ

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

#### ТЕХНИЧЕСКИ

Rendering resolution during training relative to the original resolution of the training images. At 0.5 every image is downscaled to 50 % width × 50 % height (i.e. 25 % of the pixels) and the Gaussian rendering happens at this lower resolution. Reduces both memory and compute quadratically. Important: T11 `densifyGradThreshold` has to match the chosen resolution — the gradient magnitudes scale with  $1/\text{resolution}^2$ , which is why `.quickTest` (0.25×) has a much higher threshold ( $4e-6$ ) than `.full` (1.0×,  $1.1e-6$ ). RadianceKit warns at very large images and adjusts automatically — 3 MP target resolution. With extreme 4K input images, 0.5 or even 0.25 would make sense, otherwise any Mac will run only in CPU compaction.

#### ПРОСТЫМИ СЛОВАМИ

How big the images are during training. 1.0 = original, 0.5 = half size. Half size = four times faster but the finest details are missing. The presets pick the right value; with extremely large input images (over 12 megapixels) the app switches down automatically.

### T23 resolutionWarmupScale

#### ПОДРОБНОСТИ

**Default:** 0.0 (= disabled) **Range:** 0 or 0.1 – **Defined in:**

#### ТЕХНИЧЕСКИ

V133 optimization: train the densification phase (iter 0 to T2) at a lower resolution than the refinement phase. V308 turned it off again for `.full` because with `T22 = 1.0` and cosine annealing, the time win was marginal and quality suffered slightly. Stays in the field catalog because it could become useful again with 4K inputs and long training runs — Q6 curriculum (T80) picked up a similar logic, though there it is tied to the LR schedule. If enabled and T80 `curriculumResolutionRamp` is also true, Q6 wins and overrides this value.

#### ПРОСТЫМИ СЛОВАМИ

Special feature: in the first training half learn from smaller images, in the second from large ones. Saves time. Off because the newer Q6 variant solves it better.

**T24** tileSize **ПОДРОБНОСТИ****Default:** 16 **Range:** 8, 16, 32 **Defined in:** **ТЕХНИЧЕСКИ**


Size of the rasterization tiles in pixels. The Gaussian Splatting rendering is tile-based: the image is divided into 16×16 pixel tiles, each tile collects the Gaussians relevant to it, sorts them by depth and blends them. 16 is the standard used by practically all 3DGS implementations and is hard-coded in RadianceKit's Metal kernels; changing this value would require shader recompilation and is not effective in the current state. Stays as a field in case a future engine version supports dynamic tile size.

 **ПРОСТЫМИ СЛОВАМИ**

Internal render parameter.  
Default 16, do not change.

**T25** throttleDelayMs **ПОДРОБНОСТИ****Default:** 0 (initializer, `.full`, MCMC, Scene-Class), 0 ( `.preview` ) **Range:** 0 – 100 **Defined in:** **ТЕХНИЧЕСКИ**

Artificial delay between training iterations in milliseconds. 0 = full speed (default). Higher values make the Mac more «usable» during training, by giving GPU/CPU regular breathers — the responsiveness of other apps improves, but training time grows linearly with the delay. Typical values: 1–2 ms («light» throttling, +5 % training time, Mac feels more responsive), 5 ms («medium», +15 % training time), 10+ ms («eco», potentially double the training time). Offered in the Inspector under «Performance» but not in the default view — see backlog `dev_ux-backlog.md` which suggests removing it from the Expert View because misunderstood it dramatically extends training time.

 **ПРОСТЫМИ СЛОВАМИ**

How many milliseconds of pause the app makes between training steps. 0 = no pause, as fast as possible. Higher values make the Mac more usable during training — but the training takes longer too. On an M3 Ultra or Mac Studio you can leave this at 0; on a MacBook Air 2 or 5 would be a good value.

## Diagnostics and Point Cloud Preparation (T26–T30)

### T26 depthDistortionWeight

#### ПОДРОБНОСТИ

**Default:** 0.0 (= disabled) **Range:** 0 or 0.0001 – 0.05

**Defined in:**

#### ТЕХНИЧЕСКИ

V366 experimental: weight of a depth distortion regularization loss. Penalizes Gaussians that, along a render ray, are stacked in depth but conceptually belong to the same surface — this encourages concentrated depth distributions and reduces floaters. Tests: 0.01 → 4.5 % worse, 0.001 → 8.1 % worse. The theoretical advantage — improving multi-view consistency — does not show up in the L1 loss, because the hypothesis implicitly assumes that the SfM geometry is correct and the Gaussians just need to be «stacked». In practice the SfM point cloud is usually the weakest component, not the stacking. Stays available for multi-view datasets with particularly clean poses (synthetic, Mip-NeRF 360 with ground truth).

#### ПРОСТЫМИ СЛОВАМИ

Experimental feature to avoid multiple splats lined up at the same spot. Not enabled because the tests showed no benefit.

### T27 singleViewOverfit

#### ПОДРОБНОСТИ

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

#### ТЕХНИЧЕСКИ

Diagnostic flag: when true, every training iteration must use camera index 0 instead of a random one from the camera pool. Reason: if the model can't even overfit a single view (i.e. the loss on view 0 doesn't go to zero even after 10 000 iterations), there's a fundamental bug in the forward/backward pass. This switch was used intensively during the development of the Metal shaders and the differentiable rasterizer kernels — V42–V47 phase. Today only available as a sanity check if someone has modified backend code and wants to do a regression test. Via CLI with `--single-view`.

#### ПРОСТЫМИ СЛОВАМИ


Test mode for developers. They can use it to check whether the app can even learn from a SINGLE image. Irrelevant for normal users, always leave off.

**T28 maxCameras** **ПОДРОБНОСТИ**

**Default:** 0 (= «use all cameras») **Range:** 0 or 1 – N  
**Defined in:**

 **ТЕХНИЧЕСКИ**

Diagnostic limit from V43: train only with the first N cameras, ignore all further ones. Original reason: test the hypothesis that too many cameras create gradient conflicts (too many conflicting loss signals for the same Gaussian). Test result: no systematic advantage from artificial limiting — more frames practically always bring more quality. Stays as a CLI flag ( `--max-cameras N` ) for targeted experiments, e.g. «does training work on the first 100 images of a 1 500-image drone flight?» Not exposed in the UI.

 **ПРОСТЫМИ СЛОВАМИ**


Diagnostic field for developers — use only the first N images, ignore the rest. Normal user doesn't need this, value at 0 = all images. More images = better result (see `feedback_more-frames-better.md`).

**T29 maxInitialPoints** **ПОДРОБНОСТИ**

**Default:** 0 (= «use all SfM points») **Range:** 0 or 1 000 – 200 000+ **Defined in:**

 **ТЕХНИЧЕСКИ**

V54 safety net: limits the number of initial SfM points the training starts with. Dense COLMAP reconstructions can produce > 60 000 points, which with large initial scales leads to 200–300 Gaussians per pixel overlap — this creates a «fog field» in which training does not converge. Subsampling to ~16 000 points (hard-cap logic in the training engine) brings initial density to the level used by reference 3DGS, and dramatically reduces overlap. Set automatically with very dense SfMs; via CLI with `--max-points N`.

 **ПРОСТЫМИ СЛОВАМИ**

How many starting points from the camera reconstruction are used. With very dense reconstructions (more than 60 000) the app automatically limits to 16 000 — otherwise there is too much fog at the beginning. You don't have to set this; the app handles it.

**T30 cameraClusterOutlierMultiplier** **ПОДРОБНОСТИ**

**Default:** 10.0 (all presets — never overridden)

**Range:** 1.0 – 100.0 **Defined in:**

 **ТЕХНИЧЕСКИ**

Multiplier for the camera cluster outlier filter, introduced in Phase 3.10 A.1. Before training the training engine computes the centroid of all camera positions and the maximum distance of any camera from the centroid. SfM points whose distance from the centroid exceeds `multiplier × maxCameraDistance` are discarded as outliers. Default 10× preserves the pre-Phase 3.10 behavior. A subtle bug: tighter SfM (cameras closer together) → smaller → smaller threshold → more points are discarded as outliers. Looser SfM → larger threshold → fewer points discarded. This is one of the causes of the Phase 3.9 funnel-vs-training anti-correlation: better SfM can downstream lead to worse training because too many initial points are killed. The field is available as a CLI override (`--camera-cluster-outlier-multiplier`) for the A.3 sweeps; not exposed in the UI. Values below 5 are usually too restrictive, above 20 ineffective.

 **ПРОСТЫМИ СЛОВАМИ**

Special filter that discards points from the reconstruction lying far away from the camera cloud. 10 = the app is generous, keeps almost everything. Increasing can make sense when far-away points (mountains in the distance) look in the image like floating blobs. Lower setting only in emergencies — you'll lose detail in the distance.

## Regularization (T31–T37)

### T31 coarseToFineBlurRadius

#### ПОДРОБНОСТИ

**Default:** 0 (= disabled) **Range:** 0 or 1 – 10 **Defined in:**

#### ТЕХНИЧЕСКИ

V369 experimental: box blur radius applied at the start of the densification phase to the ground truth image, reduced linearly to 0 by the end of densification ( T2 ). Hypothesis: coarse-to-fine training — first learn coarse structures, then details — should yield more stable geometry. Tests: r=3 → 9.6 % worse, r=1 → 5.1 % worse. Reason for failure: densification decides based on image domain gradients, and blurring reduces exactly the signals important for «must clone here». Stays in the field catalog for future tests with a different density control scheme.

#### ПРОСТЫМИ СЛОВАМИ

Experimental «coarse-first-then-detail» mode. Brought nothing, stays off.

### T32 scaleRegWeight

#### ПОДРОБНОСТИ

**Default:** 0.0 (= disabled) **Range:** 0 or 0.0001 – 0.05 **Defined in:**

#### ТЕХНИЧЕСКИ

V370 experimental: L1 regularization on world-space scale. Penalizes Gaussians that grow too large — prevents «mega splats» covering whole wall surfaces with a single Gaussian. Tests: 0.01 → 200 % worse loss (2 M Gaussians, total explosion), 0.001 → 214 % worse. Reason: scale regularization conflicts with density control — smaller scales mean more Gaussians are needed, so density control splits more often, which in turn means more gradient work. Disabled, but documented for Mip-Splatting experiments (T74): in that context a scale lower bound could make sense.

#### ПРОСТЫМИ СЛОВАМИ


Regularization that forces splats to stay small. Triggered splat explosions in tests (millions of splats). Do not enable.

**T33 anisotropyRegWeight** **ПОДРОБНОСТИ**

**Default:** 0.0 (= disabled) **Range:** 0 or 0.0001 – 0.05  
**Defined in:**

 **ТЕХНИЧЕСКИ**

V445 experimental: penalty on the  $\max(\text{scale})/\min(\text{scale})$  ratio, intended to prevent extremely elongated «needle» Gaussians perceived as floaters. Tests: 0.01 → 69 % worse, 0.001 → 15 % worse. Reason: regularization forces splats toward «round» shape, which on a flat surface (wall, table, floor) is exactly wrong — there a flat, broad Gaussian is more efficient than a spherical one. Disabled. V549f offered with T34 `scaleRatioPruneThreshold` an alternative more targeted approach, which was also reverted.

 **ПРОСТЫМИ СЛОВАМИ**

Regularization that penalizes very long thin splats. Sounds sensible but was worse in tests. Off.

**T34 scaleRatioPruneThreshold** **ПОДРОБНОСТИ**

**Default:** 0.0 (= disabled) **Range:** 0 or 5.0 – 100.0 (typically 10.0 – 30.0) **Defined in:**

 **ТЕХНИЧЕСКИ**

Experimental post-training pruning that deletes each Gaussian whose  $\max(\text{scale})/\min(\text{scale})$  ratio exceeds the linear threshold set here. Targets extremely elongated «needle/disc» floaters that can't be eliminated by regularization alone. In tests the pruning removed floaters as hoped, but also useful flat splats on walls and floors — the image became hole-ier. Hence off by default; the CLI flag (`--scale-ratio-prune N`) remains for targeted experiments. Recommended values if you do want to test: 30 (very conservative, removes only extreme outliers), 10 (aggressive, costs detail).

 **ПРОСТЫМИ СЛОВАМИ**


Attempt to filter out very elongated splats after training. Was net-negative in tests — floaters gone but also detail gone. Off.

**T35 opacityRegWeight** **ПОДРОБНОСТИ**

**Default:** 0.0 (= disabled) **Range:** 0 or 0.0001 – 0.05  
**Defined in:**

 **ТЕХНИЧЕСКИ**

V446 experimental: binary cross-entropy penalty that pulls opacity toward 0 or 1 (i.e. away from «semi-transparent»). Hypothesis: sharper opacity distribution would improve image clarity. Tested combined with T33 → regularization costs quality, both disabled. Disabled. Attention: in 1.4.3 beta a bug appeared that had exactly this field with a changed default value (initializer = 0.01), which led to mass extinction of the Gaussian count (460 K → 5 in one iteration). Since 1.4.4 hard-pinned to 0.0 as default.

 **ПРОСТЫМИ СЛОВАМИ**


Regularization that makes splats either fully transparent or fully solid. Brings nothing, can even become dangerous (1.4.3 bug mass extinction). Leave at 0.

**T36 opacityDecayFactor** **ПОДРОБНОСТИ**

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

 **ТЕХНИЧЕСКИ**

V546 implementation of the HTGS scheme (Hierarchical Time-Gating, Eurographics 2025): every T37 `opacityDecayInterval` iterations the sigmoid opacity of each Gaussian is multiplied by this factor. 0.9995 × 100 applications gives ~95 % residual per densification phase — a slight but steady downward pressure on all opacities, which reliably drops weakly contributing Gaussians below the T14 `pruneOpacityThreshold`. The result: 14 % better L1 loss on Horse Full (3-trial avg V546) vs V438 without decay. Active only during the densification phase (until T2 ), after that training continues without decay so the opacities established during refinement stay stable. Not used under MCMC (MCMC has its own mechanisms via T67 `mcmcRelocationInterval` + T68 `mcmcDeadOpacityThreshold`).

 **ПРОСТЫМИ СЛОВАМИ**

«Gentle fade» of all splats over the training time. Makes inactive splats transparent faster, so they disappear during cleanup. Was the most important quality lever of the V546 update: 14 % better. Built into the Quality preset. Even tweaking not recommended because precisely balanced.

**T37** `opacityDecayInterval` **ПОДРОБНОСТИ****Default:** 50 **Range:** 10 – 500 **Defined in:** **ТЕХНИЧЕСКИ**

Iteration interval at which T36 `opacityDecayFactor` is applied. HTGS paper default 50, left at that in `.full`. Long intervals (>200) partially cancel the effect because between two applications enough gradient updates happen that opacity rises again. Shorter intervals (<20) make decay too aggressive. Active only in the densification phase.


 **ПРОСТЫМИ СЛОВАМИ**

How often the «fade» is applied. 50 = every 50 iterations a small fade step. Fits.

## Refinement (T38–T44)

**T38** `gradientAccumulationSteps` **ПОДРОБНОСТИ****Default:** 1 (= «one view per Adam step») **Range:** 1 – 8 **Defined in:** **ТЕХНИЧЕСКИ**

V424 feature: number of views whose gradients are accumulated before an Adam update runs. With `> 1` the app runs on a separate «unfused» backward project path that sums the gradients into a separate buffer; the final application scales by  $1/N$  to keep magnitude constant. V424 tested 2-view → quality-neutral but 10 % slower (because unfused is more expensive than fused). Reverted for `.full` but deliberately used for MCMC — `.fullMCMC` runs with, but V544a tests showed that with the quality gap to Classic shrinks to 5 % (instead of 11 %). In the initializer default 1, in the current preset 1, remains a CLI flag (`--accum-steps N`).

 **ПРОСТЫМИ СЛОВАМИ**


How many images the app considers before adjusting the splats. 1 = each image individually. Higher = look at several images together and then apply an average. Brings nothing in the standard case; under MCMC 2 can help a bit.

**T39** testViewIndices **ПОДРОБНОСТИ**

**Default:** `[]` (= empty, all views are used for training) **Range:** `Set<Int>`, arbitrary subset of camera indices **Defined in:**

 **ТЕХНИЧЕСКИ**

V546 feature: set of camera indices that are NOT used for training but saved as holdout for PSNR/SSIM/LPIPS evaluation. Set automatically when the `--benchmark` CLI flag is active: then every 8th view starting from index 0 (LLFF standard, identical to Mip-NeRF 360 and 3DGS paper conventions). Without benchmark empty — training uses all views. **Caution:** manually setting this field without understanding the indices can make the benchmark unusable (e.g. when all indices are set above N while there are only N-50 views → no holdouts → no evaluation). When you export your own preset, testViewIndices is not persisted because it is scene-dependent and would otherwise leave nonsensical values between different datasets.

 **ПРОСТЫМИ СЛОВАМИ**

Which images during training are «left out» to use later for quality measurement. You don't set this yourself; the `--benchmark` flag does it automatically (every eighth image is test). If you set your own indices: dangerous, can corrupt the benchmark.

**T40** refinementPruneInterval **ПОДРОБНОСТИ**

**Default:** 0 (= disabled) **Range:** 0 or 100 – 5 000 **Defined in:**

 **ТЕХНИЧЕСКИ**


V425 feature: every N iterations during the refinement phase (after `T2`) an additional prune pass runs that removes Gaussians with  $\text{sigmoid}(\text{opacity}) < T_{41}$  `refinementPruneOpacityThreshold`. Reason: during densification there are regular density control calls, after that not anymore — but Gaussians whose opacity continues to drop stay in the buffer. V425 tested and reverted: the additional pruning correlated with V426 (Two-Phase Densification, which also ended in 0-Gaussians cascade failure). Disabled. CLI flag available for experiments; if enabled, 1 000 or 2 000 are sensible values.

 **ПРОСТЫМИ СЛОВАМИ**

Additional cleanup during the refinement phase. Brings nothing, stays off.

**T41 refinementPruneOpacityThreshold** **ПОДРОБНОСТИ****Default:** 0.0 (= «use T14 ») **Range:** 0 or 0.001 – 0.1**Defined in:** **ТЕХНИЧЕСКИ**


V425b: separate opacity threshold for refinement pruning. After densification most Gaussians have reached a clearly higher opacity ( $> 0.001$ ), so the default T14 pruneOpacityThreshold would be too lax. If T40 is active, this field determines its own threshold. At 0.0 T14 is used as before. Only relevant if T40  $> 0$ .

 **ПРОСТЫМИ СЛОВАМИ**

Threshold for the additional refinement cleanup (see T40). Both fields inactive, so irrelevant.

**T42 midTrainingCompactificationIterations** **ПОДРОБНОСТИ****Default:** [] (= disabled) **Range:** [Int], values in (densifyUntilIteration, maxIterations) **Defined in:** **ТЕХНИЧЕСКИ**

V549 feature: explicit iteration points during the refinement phase at which a compactification pass runs (removes sigmoid(opacity)  $< 0.01$  + outlier-scale Gaussians, same logic as T56 postTrainingCompactification). Reason: long refinement phases can show confetti/floater accumulation, whose SH then overfits to view-specific artifacts. Typical configuration if enabled: [10000, 20000, 30000] for 40K Classic. **BUT:** V549 A/B tests on the Family dataset showed worse L1 in all configurations: [10K, 20K, 30K]@0.01  $\rightarrow$  -48 % count but +36 % L1; [20K, 30K]@0.005  $\rightarrow$  -44 % count but +45 % L1; [20K, 30K]@0.001  $\rightarrow$  -17 % count but +87 % L1. Hence disabled. CLI flag `--mid-compact "10000, 20000"` available, if you prefer the visual floater tradeoff (less confetti in the viewport) over the loss regression.

 **ПРОСТЫМИ СЛОВАМИ**

Mid-training cleanup actions during training. In tests the cleanup made the end result worse (yes, fewer floaters, but also less detail). Off, can be enabled via CLI, in case floaters bother you more than a somewhat mushier image.

**T43 frustumCullEnabled** **ПОДРОБНОСТИ****Default:** false **Range:** boolean **Defined in:** **ТЕХНИЧЕСКИ**

V549b feature: after training all Gaussians outside the union of all training camera frusta are removed. Such Gaussians were never constrained by the loss signal and are always floaters. Particularly effective for scenes where the novel view sits behind or beside the camera path (e.g. behind a linear drone flight) — the floaters there are never visible during training but very much so later when moving in the 3D viewer. V549b A/B on drone flights showed positive results, hence available as opt-in. Default false because for object captures with full orbit coverage the frustum union encompasses the whole scene and the feature removes nothing — offered in Settings under «Floater Reduction» and also implicitly tested in Q9 Outdoor preset via T44 `frustumCullExpansion` (Q7 BayesOpt didn't enable it though, because outdoor sky dome solves the same problem better).

 **ПРОСТЫМИ СЛОВАМИ**

Special filter for drone flights or linear captures: after training, splats that were not «seen» in any camera are deleted. Optional, switchable on in Settings. For simple object captures unnecessary.

**T44 frustumCullExpansion** **ПОДРОБНОСТИ****Default:** 1.1 **Range:** 1.0 – 2.0 **Defined in:** **ТЕХНИЧЕСКИ**

NDC margin for T43 `frustumCullEnabled`. 1.0 would cut exactly at the image edge, which would trim wobbly splats near the edge too aggressively. 1.1 = 10 % padding beyond the exact camera framing — gives some tolerance for edge pixels that might still become visible in a slightly shifted novel view. Values > 1.2 make the cull practically ineffective because the expanded frustum encompasses too much space.

 **ПРОСТЫМИ СЛОВАМИ**

How strictly the above-mentioned filter clips. 1.1 = a bit of safety margin to the image edge. Leave the value.

## Sky Dome (T45–T48)

### T45 skyDomeEnabled

#### ПОДРОБНОСТИ

**Default:** false (initializer + all presets except P9 Outdoor) **Range:** boolean **Defined in:**

#### ТЕХНИЧЕСКИ

V549e feature: before training starts a spherical point cloud is generated (Fibonacci sphere with T46 sample points), placed at a radius of T47  $\text{skyDomeRadiusMultiplier} \times \text{scene\_extent}$  around the scene center, and initialized with the colors from sky-masked pixels of all training cameras (see T20 skyMaskingEnabled). These sky dome Gaussians are inserted at the beginning of the Gaussian buffer and during training «frozen» (position/scale/rotation gradients = 0, only SH and opacity remain optimizable). Effect: instead of black «confetti» areas in the distance, the user sees a real sky in novel views. The V549e MVP works very well on drone and landscape scenes; in P9 Outdoor preset default-on. Leave off for indoor scenes — the sphere would dangle uselessly outside the room.

#### ПРОСТЫМИ СЛОВАМИ

Enables an artificial «sky dome» around the scene. Makes outdoor captures much nicer: instead of black blobs at the image edge, the app shows the real sky. Mandatory for drone flights and landscapes, pointless for interiors.

### T46 skyDomeSampleCount

#### ПОДРОБНОСТИ

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

#### ТЕХНИЧЕСКИ

Number of Fibonacci sphere sample points on the sky dome sphere. Higher values → denser sky dome (better at large resolutions and lots of visible sky), but higher memory consumption. 5 000 is the sweet spot for 4K renderings; at lower resolutions 2 000–3 000 suffice. The points are initialized by cosine distance to each training camera view vector with the corresponding sky-masked pixels — sample points whose view cone is seen by no camera keep a low opacity initial value, but stay unchanged during training (frozen).

#### ПРОСТЫМИ СЛОВАМИ


How dense the artificial sky is. 5 000 points usually suffice. More = smoother transition from the distance, but costs some memory.

**T47 skyDomeRadiusMultiplier** **ПОДРОБНОСТИ**

**Default:** 30.0 (initializer + most presets), 59.0 (P9 Outdoor, Q7 BayesOpt optimum) **Range:** 5.0 – 200.0 **Defined in:**

 **ТЕХНИЧЕСКИ**

Radius of the sky dome sphere relative to the scene extent (= mean distance between camera positions). 30 = the sphere has 30× the diameter of the camera cloud. Too small (< 5) → sky dome interferes with the scene itself (e.g. a sky dome splat lands in the foreground); too large (> 100) → float32 precision loss at the sky dome positions, which triggers render glitches in the distance. Q7 BayesOpt on Bicycle (Mip-NeRF 360) found 59.0 as scene-specific optimum for outdoor — this suggests that the default 30.0 is too small for deep landscapes and that sky dome pixels visibly render as a «wall» in image-edge regions.

 **ПРОСТЫМИ СЛОВАМИ**


How far away the artificial sky dome should be. 30 = quite far. With big landscapes 50–60 is better (Outdoor preset handles that automatically). Too small would be like having blobs right in front of the lens.

**T48 frozenGaussianCount** **ПОДРОБНОСТИ**

**Default:** 0 (= no frozen Gaussians) **Range:** 0 or 1 – **T46 Defined in:**

 **ТЕХНИЧЕСКИ**

Number of Gaussians at the start of the buffer whose position/scale/rotation gradients are zeroed in the optimizer — they remain spatially rigid over the entire training. Density control may not clone, split or prune them. Used for sky dome injection (see **T45**): when sky dome is on, this field is automatically set to **T46 skyDomeSampleCount**. Manual setting is possible (e.g. to freeze a pre-placed point cloud from a LiDAR scan), but not directly accessible in the UI. Important: the first N Gaussians in the buffer are always the frozen ones — the order in the buffer decides, not an explicit index.

 **ПРОСТЫМИ СЛОВАМИ**

How many splats at the beginning are fixed and may not move. Set automatically to the sky dome count when sky dome is on. You don't need to tweak this yourself.

## Adam + LR Schedule (T49–T55)

### T49 adamResetIteration

#### ПОДРОБНОСТИ

**Default:** 0 (= disabled) **Range:** 0 or 100 – **Defined in:**

#### ТЕХНИЧЕСКИ

V430 feature: iteration at which the Adam optimizer momentum accumulators (m1, m2) are reset to zero. Bias correction afterwards runs with `(iter - adamResetIteration)` instead of `iter`. V430 tested reset at 5 000 (after densification end) → 12.8 % worse loss. Reason: the Adam momentum that built up during densification carries information about the typical gradient magnitudes and accelerates the refinement phase. Throwing it away costs the first ~500 iterations of refinement in convergence. Disabled. Stays as CLI flag for research experiments.

#### ПРОСТЫМИ СЛОВАМИ

Reset button for the internal Adam optimizer «memory». Hurt in tests, stays off.

### T50 positionLRScheduleEndIteration

#### ПОДРОБНОСТИ

**Default:** 0 (initializer = «use maxIterations»), 20 000 ( `.full` — cosine ends at 20K although `maxIter=35K`), 30 000 ( `.fullClassicPaper` )  
**Range:** 0 or 1 000 – **Defined in:**

#### ТЕХНИЧЕСКИ

V431 feature: iteration at which the cosine annealing curve for position LR reaches its minimum. If 0, this is identical to `T1 maxIterations`. If > 0, the schedule runs up to this value and stays constant at `T4 positionLearningRateFinal` afterwards. This allows an «extended refinement phase» with minimal but constant learning rate — refines positions slowly without a new decay. `.full` does this (schedule ends at 20K, training runs to 35K), V434c/V434d confirmed: 15K and 25K both about the same, 20K marginally optimal. Used together with `T51` to also modify the non-position LRs in the extended phase.

#### ПРОСТЫМИ СЛОВАМИ


When the app stops lowering the position learning rate further. If lower than the maximum iteration, the rest runs with a constant mini rate — this refines very slowly but very stably. Built into the Quality preset, you don't need to tweak.

**T51** extendedPhaseLRDecay **ПОДРОБНОСТИ**

**Default:** 0.0 (= disabled, constant LRs) **Range:** 0 or 0.01 – 1.0 **Defined in:**

 **ТЕХНИЧЕСКИ**

V433 feature: minimal multiplier for the non-position LRs (scale, rotation, opacity, SH) in the «extended phase» — i.e. after T50 is reached and position LR is already at T4. If 0.1, scale/rotation/opacity/SH are themselves cosine-decayed from 1.0 (= their standard LR) to 0.1× of their standard. If 0.0 (default), they stay constant. V457 tested full decay (0.0, i.e. decay to zero) against no decay and found: avg 0.0400 (2 runs), the same loss as V438 without decay. Behavior cleaner with decay but not measurably better. Hence disabled. Stays in the CLI as `--nonpos-lr-scale F`.

 **ПРОСТЫМИ СЛОВАМИ**

In the late refinement phase also make color and shape learning rates smaller. Makes training «more stable» but empirically not better. Off.

**T52** adaptiveDensifyThreshold **ПОДРОБНОСТИ**

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

 **ТЕХНИЧЕСКИ**

V440 experimental: when true, the app computes each densification step the p98 of the current gradient distribution and uses it as dynamic threshold (clamped to at least 0.5× of the configured value from T11 so it doesn't drift too far). Hypothesis: automatic adaptation to the current scene phase would make density control more robust — e.g. stricter pruning at the start, laxer later, or vice versa. V440 tested and reverted: catastrophic drop to 63 K Gaussians (mass pruning, because the p98 in the first iterations is extremely high and afterwards almost nothing exceeds the threshold). The fixed threshold is already well calibrated, dynamic adjustment hurts more than it helps. Q5 (T77) offers an alternative adaptive logic via rolling median that avoids the problem.

 **ПРОСТЫМИ СЛОВАМИ**


Adaptive version of the densify threshold. In tests catastrophic (splat count crashed to 63K). Off. Q5 has a better variant.

**T53 mergeAfterDensification** **ПОДРОБНОСТИ**

**Default:** false (initializer), true ( `.full` , `.classicBalanced` , `.fullClassicPaper` ) **Range:** boolean **Defined in:**

 **ТЕХНИЧЕСКИ**

V438 feature: at the end of the densification phase (iter `T2` ) a one-time merge pass runs that combines nearby Gaussians with similar scale and color. Reduces the Gaussian count by typically 5–15 % without visible quality loss. Reason: after intense cloning, clusters of nearly identical Gaussians arise that contribute nothing new — merging frees optimizer capacity for other areas. Default in Classic Quality presets. Not used under MCMC, because MCMC through its relocation logic doesn't let such clusters form in the first place.

 **ПРОСТЫМИ СЛОВАМИ**

At the end of the splat duplication phase combine clones that are nearly identical. Reduces data size without visible effect. By default on in the Quality preset.

**T54 densifyPhase2FromIteration** **ПОДРОБНОСТИ**

**Default:** 0 (= disabled) **Range:** 0 or `T2` – `T1` **Defined in:**

 **ТЕХНИЧЕСКИ**


V426 experimental: enables a second densification phase that starts after the refinement pause at this iteration and runs until `T55` . Hypothesis: after a refinement phase the gradient accumulators have more stable magnitudes and can more precisely indicate which regions still need additional Gaussians. V426 tested and reverted: two-phase densification fell into 0-Gaussians cascade failure (combined with V425 refinement pruning it destroyed the buffer). Disabled. CLI flag available for experiments.

 **ПРОСТЫМИ СЛОВАМИ**

Second multiplication round after a pause. In tests it annihilated the splat inventory. Off.

**T55** densifyPhase2Untillteration **ПОДРОБНОСТИ****Default:** 0 **Range:** 0 or T54 – T1 **Defined in:** **ТЕХНИЧЕСКИ**

End of V426 two-phase densification. Only relevant when `T54 > 0`. Both fields together disabled.

 **ПРОСТЫМИ СЛОВАМИ**

End of the second multiplication round (see T54). Both off.

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

**T56** postTrainingCompactification **ПОДРОБНОСТИ****Default:** true (in all production presets), false ( `.quickTest`, `.preview` ) **Range:** boolean **Defined in:** **ТЕХНИЧЕСКИ**

V443 feature: after training ends, Gaussians with `sigmoid(opacity) < 0.01` are hard-removed (they practically don't contribute to the image anymore). Reduces Gaussian count by typically 58 % and export file size by 55 % without visible quality loss. On by default in production presets — the end result should ship as compact as possible. Off in `.quickTest`, because a diagnostic run isn't exported anyway. Unlike T42 `midTrainingCompactificationIterations` (V549) the compactification only happens at the end — refinement can use all Gaussians until then.

 **ПРОСТЫМИ СЛОВАМИ**


Cleanup after training: nearly invisible splats are removed. Makes the export file roughly half as large without quality loss. Mandatory feature, leave off only in diagnostic runs.

**T57 metalFXUpscaling** **ПОДРОБНОСТИ**

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

 **ТЕХНИЧЕСКИ**

V444 feature: enables Apple’s MetalFX Spatial Upscaler instead of bilinear interpolation in the 3D viewer output. When training resolution < viewport size (e.g. training at 0.5x, viewport display at full resolution), MetalFX can deliver a clearly sharper image. Changes live in the viewport, no retraining required. Mutually exclusive with T58 `mpsLanczosScaling` — MetalFX takes precedence. Recommendation: enable when the image in the viewer looks «washed out» compared to expected detail.

 **ПРОСТЫМИ СЛОВАМИ**

Apple ML-based image sharpening in the 3D viewer. Helps when you trained at a lower resolution and want to show the result in full screen. Live toggle, give it a try.

**T58 mpsLanczosScaling** **ПОДРОБНОСТИ**

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

 **ТЕХНИЧЕСКИ**

V444 feature: `MPSImageLanczosScale` for viewport scaling instead of bilinear interpolation. Lanczos is a classic Sinc-based resampling method that delivers significantly sharper results than bilinear with minimal overhead. Live toggle. Overridden by T57 when both are on.

 **ПРОСТЫМИ СЛОВАМИ**


Classic sharpening method for the 3D viewer (Lanczos). MetalFX (T57) is ML-based and usually better; Lanczos is a less aggressive alternative.

**T59** `livePreviewInterval` **ПОДРОБНОСТИ**

**Default:** 50 (initializer and most presets) **Range:** 0 (off) or 10 – 5 000 **Defined in:**

 **ТЕХНИЧЕСКИ**

How often during training the 3D viewer is refreshed with the current Gaussians. 50 = every 50 iterations a new render in the viewer — enough to observe progress without slowing training. 0 = viewer is not updated at all (background training, max speed). Typical adjustment: under `.quickTest` drop to 10 (you want to see every step), on long MCMC runs raise to 500–2000 (update overhead in sum noticeable).

 **ПРОСТЫМИ СЛОВАМИ**


How often the 3D preview during training is refreshed. 50 = every 50 iterations. Higher = less often = a bit faster, but you see progress less often. 0 = no preview (for maximum speed).

**T60** `perceptualLossWeight` **ПОДРОБНОСТИ**

**Default:** 0.0 (= disabled) **Range:** 0 or 0.001 – 0.5 **Defined in:**

 **ТЕХНИЧЕСКИ**

V444 future feature: weight of a perceptual loss term via MPSGraph (VGG-like small network). Would capture structural and textural similarity at a higher semantic level than L1+SSIM — typical in research pipelines where «pixel-perfect» matters less than «looks realistic». Implementation pending (code stub present, but forward pass not implemented). Default 0.0. Stays in the field catalog for future activation; CLI flag `--percep-weight F` reserved.

 **ПРОСТЫМИ СЛОВАМИ**

Planned feature that with AI assistance aims for «natural looking» instead of «pixel-accurate». Not yet fully implemented.

## MCMC Densification (T61–T73)

### T61 densificationStrategy

#### ПОДРОБНОСТИ

**Default:** `.classic` (initializer + Classic presets),  
`.mcmc` (all MCMC presets + Scene-Class) **Range:**  
`.classic` or `.mcmc` **Defined in:**

#### ТЕХНИЧЕСКИ

Chooses between classic densification (clone/split/prune, Kerbl et al. 2023) and MCMC densification (Stochastic Gradient Langevin Dynamics with relocation, Kheradmand et al. NeurIPS 2024). With `.classic` T11–T16 is evaluated, with `.mcmc` the T62–T73. Beware when switching: Classic defaults and MCMC defaults are completely differently calibrated — whoever flips the picker in the Expert View without loading a matching preset risks 1.4.3-bug-style mass extinction (460 K → 5 in one iteration, because MCMC opacity reg at 0.01 kills the Classic opacities). Hence the MCMC initializer defaults are deliberately «soft-washed» (all reg values 0.0).

#### ПРОСТЫМИ СЛОВАМИ

Which algorithm is used to multiply the splats. Classic = original method (fast, many splats). MCMC = newer method (slower, far fewer splats, but more compact). The presets pick the right one. Only switch yourself if you also load the matching preset (P5–P7 or P8–P10).

## T62 mcmcMaxGaussians

### ПОДРОБНОСТИ

**Default:** 150 000 (initializer + `.fullMCMC` + `.mcmcBalanced`), 100 000 (`.mcmcPreview`), 1 500 000 (`.fullMCMCMip` — Mip-Splatting variant with 10× budget), 1.19 M (`.renderPreset`), 1.25 M (`.outdoorPreset`), 670 K (`.indoorPreset`)  
**Range:** 0 (= «use buffer capacity») or 10 000 – 5 000 000 **Defined in:**

### ТЕХНИЧЕСКИ

Hard upper bound for the number of Gaussians under MCMC strategy. The count grows gradually by `T70 mcmcGrowthRate` (typically 5 %) per relocation step up to this cap. V473/V531 found 150 K as sweet spot — above 200 K dilutes splat quality (too many small redundant Gaussians), below 100 K leaves the scene under-densified. With very large scenes (e.g. 1 545-photo drone flight with 158 K SfM init), 150 K is too low — hence the 1.4.5 extension `T72 mcmcCapMultiplier` + `T73 mcmcAutoScaleByScene`. Q7 BayesOpt found scene-specific optima between 670 K (Indoor) and 1.25 M (Outdoor). With value 0 the engine uses the full buffer capacity as cap.

### ПРОСТЫМИ СЛОВАМИ

Maximum number of splats under MCMC. 150 000 is the default and suffices for most scenes. Outdoor and render presets (P8, P9) go to 1+ million for more detail-rich scenes. Raising can bring detail, costs memory; lower is more an emergency brake.

## T63 mcmcNoiseScale

### ПОДРОБНОСТИ

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

### ТЕХНИЧЕСКИ

Multiplier for the Gaussian noise that in each MCMC iteration is added to the position of each Gaussian (SGLD logic). Higher = more exploration (Gaussians wander more, finding potentially better spots), lower = more exploitation (Gaussians stay where they're already good). V467 and V536 confirmed 5e-5 as optimal — 1e-5/2e-5 too little exploration, 1e-4 too much (splats diffuse). Cosine-decayed over training time until `T69 mcmcNoiseDecayEnd` — at the end of the decay range noise is effectively 0 and the Gaussians converge.

### ПРОСТЫМИ СЛОВАМИ


How much random «wobble» the app allows the splats so they can find the best spot themselves. The default value is optimal in tests. If you crank this up, the splats get restless.

**T64** **mcmcOpacityRegWeight** **ПОДРОБНОСТИ**

**Default:** 0.0 (= disabled in RadianceKit defaults, paper: 0.01) **Range:** 0 or 0.001 – 0.05 **Defined in:**

 **ТЕХНИЧЕСКИ**

MCMC-specific L1 penalty on opacity. Paper default 0.01 (pushes unused Gaussians toward zero, makes them available for relocation). V464b showed however: without reg it's measurably better in RadianceKit (Session 28 confirmed). Reason: the pruning criterion defined with T68 `mcmcDeadOpacityThreshold` is enough on its own — an additional L1 penalty also forces valuable low-opacity Gaussians to die. Hence default 0. **Caution:** in the 1.4.3 beta build the initializer default was mistakenly 0.01, which resulted in the mass extinction bug (see T61 explanation); since 1.4.4 fixed at 0.0.

 **ПРОСТЫМИ СЛОВАМИ**


MCMC special regularization. Off because the other MCMC mechanism (threshold in T68) already covers this. Leave at 0.

**T65** **mcmcScaleRegWeight** **ПОДРОБНОСТИ**

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

 **ТЕХНИЧЕСКИ**

MCMC-specific L1 penalty on the scale eigenvalues. Paper default 0.01. V464b: without reg better, same reasoning as T64. Disabled in all RadianceKit MCMC presets. Caution as with T64: 1.4.3 bug.

 **ПРОСТЫМИ СЛОВАМИ**


Like T64 but for splat size. Off.

**T66** **mcmcRelocationInterval** **ПОДРОБНОСТИ**

**Default:** 100 (initializer + all MCMC presets, paper standard), 155 (P9 Outdoor — Q7 BayesOpt optimum) **Range:** 50 – 500 **Defined in:**

 **ТЕХНИЧЕСКИ**

Iteration interval at which MCMC relocates dead Gaussians ( $\text{sigmoid}(\text{opacity}) < T68 \text{ mcmcDeadOpacityThreshold}$ ) to new positions. V537 tested 50 (too disruptive, loss fluctuates) and 200 (marginally worse, MCMC loses responsiveness). 100 is optimal. Q7 BayesOpt on Bicycle found 155 as scene-specific optimum for outdoor — the slightly longer intervals give Adam more time to integrate newly placed Gaussians before the next reloc event puts them under pressure.

 **ПРОСТЫМИ СЛОВАМИ**


How many iterations between MCMC moving dead splats elsewhere. 100 is standard. You don't need to tweak this yourself — Outdoor preset already has the optimal value.

**T67** **mcmcWarmupIterations** **ПОДРОБНОСТИ**

**Default:** 500 **Range:** 100 – 5 000 **Defined in:**

 **ТЕХНИЧЕСКИ**

Number of initial iterations during which no MCMC relocation happens. Only after this warmup does the reloc logic kick in. Reason: in the first iterations the opacity values have not yet stabilized — if reloc started directly, Gaussians would be placed at the wrong spots and have to be moved again right away, which would destroy Adam momentum. Paper default 500. RadianceKit adopts this value because V464b showed it is robust.

 **ПРОСТЫМИ СЛОВАМИ**

How many iterations MCMC is allowed to first «arrive» before starting to move splats. 500 is standard and fits.

**T68** `mcmcDeadOpacityThreshold` **ПОДРОБНОСТИ**

**Default:** 0.005 (initializer, paper standard), 0.01 (`.fullMCMC` and all MCMC presets — V535 optimum) **Range:** 0.001 – 0.05 **Defined in:**

 **ТЕХНИЧЕСКИ**

sigmoid(opacity) threshold below which a Gaussian counts as «dead» and is eligible for relocation. V535 found 0.01 as optimal (0.005 marginal, 0.02 worse). Higher = more aggressive reloc (more Gaussians moved), lower = more cautious. 0.01 corresponds to roughly «0.5 % visual visibility». P10 Indoor uses 0.0142 as optimum via Q7 BayesOpt.

 **ПРОСТЫМИ СЛОВАМИ**

Below which transparency a splat counts as «dead» so MCMC pushes it elsewhere. 0.01 is optimal in our tests. You don't need to tweak this yourself.

**T69** `mcmcNoiseDecayEnd` **ПОДРОБНОСТИ**

**Default:** 0 (initializer = «no decay»), 160 000 (`.fullMCMC` = 80 % of 200K), 96 000 (`.mcmcBalanced` = 80 % of 120K), 40 000 (`.mcmcPreview`) **Range:** 0 or 1 000 – **Defined in:**

 **ТЕХНИЧЕСКИ**

Iteration at which the `T63 mcmcNoiseScale` noise is fully damped to zero (cosine decay from iter 0 to here). V497c/V502 found 80 % of maxIterations optimal — gives MCMC enough exploration time but leaves the last 20 % to convergence without noise. 0 = constant noise over all iterations (rarely sensible, MCMC can't converge then).

 **ПРОСТЫМИ СЛОВАМИ**


When the random «wobbling» of splats stops. In the MCMC presets at 80 % of total iterations — first exploration, then convergence. Leave the value.

**T70** **mcmcGrowthRate** **ПОДРОБНОСТИ**

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

 **ТЕХНИЧЕСКИ**

Growth rate of the MCMC population target per relocation step. The logic: at each reloc event the target population size is multiplied by  $(1 + \text{growthRate})$ , until T62 `mcmcMaxGaussians` (or the variant scaled via T72/T73) is reached. V512/V522 found 0.05 as optimal — higher values lead to too-fast growth (Gaussians are inserted before Adam momentum can integrate them), lower values to under-densified scenes at the end.

 **ПРОСТЫМИ СЛОВАМИ**


How fast the splat count grows under MCMC. 5 % per step is optimal. Leave the value.

**T71** **mcmcSigmoidK** **ПОДРОБНОСТИ**

**Default:** 100.0 **Range:** 10.0 – 500.0 **Defined in:**

 **ТЕХНИЧЕСКИ**

Sigmoid sharpness parameter for the MCMC noise attenuation. In the SGLD step the per-Gaussian noise is damped by — high-opacity Gaussians (whose logit is positive) get exponentially less noise than low-opacity ones.  $K = 100$  is sharp, meaning the transition from «full noise» to «no noise» happens very quickly around opacity 0.5. V484–V487 found  $K = 100$  optimal — smaller values (10–50) also let high-opacity Gaussians wobble (destroys converged Gaussians), larger ones ( $> 500$ ) make the transition artificially hard and dead Gaussians don't get moved at all anymore.

 **ПРОСТЫМИ СЛОВАМИ**

Special parameter that determines how sharply MCMC distinguishes between «transparent enough to move» and «solid, don't touch». The default is optimal. Don't tweak.

**T72** **mcmcCapMultiplier** **ПОДРОБНОСТИ**

**Default:** 3.0 (initializer + `.fullMCMC`), 2.0 (`.mcmcPreview`), 2.5 (`.mcmcBalanced`), 2.98 (P8 Render), 5.32 (P9 Outdoor), 1.76 (P10 Indoor)

**Range:** 0 (= disabled) or 1.0 – 10.0 **Defined in:**

 **ТЕХНИЧЕСКИ**

1.4.5 feature: scene-adaptive cap scaling. When T73 `mcmcAutoScaleByScene` is true, the effective cap is computed as (clamped to buffer capacity). Background: with large scenes (e.g. 1 545-photo drone flight → 158 K SfM init) T62 = 150 000 is too low — density control couldn't grow at all. With multiplier 3.0 the cap is scaled to 474 K in this example (158 K × 3.0). Q7 BayesOpt found scene-specific optima: outdoor benefits from a high multiplier (5.32 → ~830 K cap with 156 K bicycle init), indoor is happy with 1.76 (walls saturate faster). For the complete resolution of the cap see the method.

 **ПРОСТЫМИ СЛОВАМИ**

Multiplier that automatically adapts the splat cap to scene size. Large scene = more starting points = higher cap. The default 3× fits for most scenes; Outdoor preset goes to 5× (large depth ranges), Indoor to 1.76× (walls bound things anyway).


**T73** **mcmcAutoScaleByScene** **ПОДРОБНОСТИ**

**Default:** true (initializer + all MCMC presets)

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

 **ТЕХНИЧЕСКИ**

1.4.5 feature: master switch for the scene-aware cap logic (see T72 +). When false, only T62 `mcmcMaxGaussians` is used as cap (back to 1.4.4 behavior). On by default because the mass extinction problems with large scenes from 1.4.3 would otherwise return. Manually disable only when you explicitly want to set a hard cap — e.g. to train a 150 K variant whose final size is planable.

 **ПРОСТЫМИ СЛОВАМИ**

Enables automatic adjustment of the splat cap to scene size. On by default. Leave off only when you want exactly a specific splat count yourself.

**Mip-Splatting (Q1.5) (T74–T76)**

**Status:** Q1.5 was rejected on 2026-05-25 after 14 autonomous iterations + overnight 1.5M confidence check as «closed no-win» (max  $\Delta@2\times = +0.27$  dB,

original gate required  $\geq +1.5$  dB mean over 0.5x/2x, FAILS on 0/11 pair scenes). The fields remain **opt-in** for research experiments; all production presets have them off. See verdict: docs/plans/2026-05-25-phase-q1.5-final-verdict.md.

### T74 useMipSplatting

#### ПОДРОБНОСТИ

**Default:** false (all production presets), true (`.fullMCMCmip` — research sibling) **Range:** boolean  
**Defined in:**

#### ТЕХНИЧЕСКИ

Enables Mip-Splatting (Yu et al. CVPR 2024): 3D smoothing filter + 2D filter +  $\alpha$  compensation that limits per-Gaussian frequency to the Nyquist bound of the densest training camera sampling rate. Theoretical goal: eliminate aliasing when rendering at off-training scales (0.5x or 2x of training resolution). Enabled in the preprocess and backward projection shaders, functional correctness verified in Q1.5-D test. But: the original acceptance gate ( $\Delta@1x \geq +0.3$  dB AND  $\text{avg}(\Delta@0.5x, \Delta@2x) \geq +1.5$  dB) was reached on none of 11 pair scenes. Maximum observed: family 750K classic  $\Delta@2x = +0.270$  dB. Outdoor scenes (Truck, Flowers) even showed worsening 1x and 0.5x. Hypothesis: 3D smoothing competes with MCMC relocation at high Gs. Field remains for future multi-scale re-eval with correct Mip-NeRF-360 methodology (see O3 backlog in the benchmark path).

#### ПРОСТЫМИ СЛОВАМИ

Aliasing filter from a 2024 paper. Theoretically great, practically it brought nothing in our tests and sometimes even hurt. Remains available for tinkerers, but we don't recommend it. Leave off.

**T75 mipSmoothing3DScale** **ПОДРОБНОСТИ****Default:** 0.2 (paper default) **Range:** 0.05 – 1.0**Defined in:** **ТЕХНИЧЕСКИ**


3D smoothing scale parameter (Yu et al. §3.3, paper default 0.2). Larger = more world-space smoothing per Gaussian (= more anti-aliasing but also more blur at the default scale), smaller = sharper but more aliasing-prone. Only consulted when `T74 useMipSplatting = true`. Not further optimized in Q1.5 tests — the A/B gate already lost with the paper default 0.2, further sweeps would be pointless.

 **ПРОСТЫМИ СЛОВАМИ**

. If you haven't enabled Mip, irrelevant.

**T76 mipFilter2DVariance** **ПОДРОБНОСТИ****Default:** 0.3 (= exactly the V242 legacy behavior)**Range:** 0.1 – 1.0 **Defined in:** **ТЕХНИЧЕСКИ**

2D Mip filter variance added to the  $\Sigma_{2D}$  diagonal (variance directly, not squared). 0.3 is exactly the V242 legacy value that was hardcoded in the kernel before Mip-Splatting. When `T74 useMipSplatting = false`, the kernel ignores this value completely and writes the hardcoded 0.3 — so that the baseline cannot regress (Codex round 1 S3-1 guarantee). When, the value set here is used. Stays in the field catalog for Mip sweeps.

 **ПРОСТЫМИ СЛОВАМИ**

Further Mip-Splatting parameter. With Mip off: irrelevant.

## Adaptive Densification (Q5) (T77–T79)

### T77 adaptiveDensification

#### ПОДРОБНОСТИ

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

#### ТЕХНИЧЕСКИ

Q5 feature: rolling-median tracker as alternative to the fixed T11 `densifyGradThreshold`. When true, in every densify step the current threshold is overwritten with `median(last N avgGrad samples) × T79 adaptiveDensifyMultiplier`. `N = T78 adaptiveWindow`. Stricter than V440 p98 (the catastrophic 63 K pruning trap), median + 2× sits around the p70–p80 of the gradient distribution in steady state. Q5 tests: alone FAIL 0/3 scenes, but together with Q6 (see T80/T81) PASS 1/3 scenes — the Q5+Q6 bundle passed on 2026-05-25 as opt-in and is enabled via CLI `--adaptive-densify`. Q6 is the «carrier» of the quality gain, Q5 contributes more to stability.

#### ПРОСТЫМИ СЛОВАМИ

Self-learning densify threshold. Instead of a fixed sensitivity setting, the app adapts to the scene. On its own not better in tests, but together with the curriculum from Q6 yes. Either both on or both off.

### T78 adaptiveWindow

#### ПОДРОБНОСТИ

**Default:** 1 000 **Range:** 100 – 10 000 **Defined in:**

#### ТЕХНИЧЕСКИ

Rolling median window in densification events (NOT iterations — each T13 `densifyInterval` step yields one sample). Default 1 000 — with that means the last 100 000 training iterations contribute to the median, so typically the entire training history up to here. Early phase (before T78 samples): tracker returns nil → fallback to fixed threshold T11. Only relevant when.

#### ПРОСТЫМИ СЛОВАМИ

How many old densify steps feed into the median for T77. Default 1000 is good. Only relevant when Q5 adaptive is on.

**T79** adaptiveDensifyMultiplier **ПОДРОБНОСТИ****Default:** 2.0 **Range:** 1.0 – 4.0 **Defined in:** **ТЕХНИЧЕСКИ**

Multiplier on the rolling median for the adaptive threshold. Default 2.0 corresponds to roughly p70–p80 of the typical gradient distribution. Lower = more aggressive growth (more clones), higher = stricter (fewer clones). Q5 tests in range 1.5–3.0 — 2.0 best default. Only relevant when.

 **ПРОСТЫМИ СЛОВАМИ**

Factor for T77/T78. Default 2.0 = stricter than the typical median. Don't tweak.

**Curriculum (Q6) (T80–T81)****T80** curriculumResolutionRamp **ПОДРОБНОСТИ****Default:** false **Range:** boolean **Defined in:** **ТЕХНИЧЕСКИ**

Q6 feature: training resolution starts at 0.5x and switches at  $T50 \text{ positionLRScheduleEndIteration} / 2$  (or  $T1 \text{ maxIterations} / 2$ , if T50 is not set) to  $T22 \text{ trainingRenderScale}$ . Uses the `resize/restoreImageBuffers` infrastructure developed in Q1.5.1. Overrides `T23 resolutionWarmupScale` when enabled. Q6 passed as «carrier of the quality gain» in the Q5+Q6 bundle (see T77) — the gradual resolution increase gives the app time to find coarse geometry at lower resolution before moving on to fine detail work. Via CLI: `--curriculum-resolution`.

 **ПРОСТЫМИ СЛОВАМИ**

«First coarse, then fine» for the training resolution. Half resolution in the first half, then full resolution. Helps in certain scenes, not in others — best to enable together with T81.

**T81 curriculumSHProgression****ПОДРОБНОСТИ****Default:** false **Range:** boolean **Defined in:****ТЕХНИЧЕСКИ**

Q6 feature: overrides T21

shDegreeUpgradeIterations with [maxIter/4, maxIter/2, maxIter\*3/4], distributing SH upgrades evenly across training time instead of front-loading them. Hypothesis: stable geometry is established before color detail explosion, which places the view-direction-dependent gloss effects more precisely. Q5+Q6 together PASS 1/3 scenes, Q6 as carrier of the gain (Q5 alone FAIL). Via CLI:

`--curriculum-sh .`**ПРОСТЫМИ СЛОВАМИ**

«First shape, then color» — the gloss effects are unlocked only late in training, so the splats first find their position and size. Can be enabled together with T80; alone it doesn't bring quite as much.

**Static Presets (TP1–TP9)**

Only the structural differences from the initializer default here. The full marketing description of the eleven UI presets P1–P11 is in Chapter 7.

**TP1 .preview****ПОДРОБНОСТИ**

Diagnostic/preview preset for systems  $\geq 10$  GB RAM. Overrides vs. initializer: - 30 000  $\rightarrow$  5 000 - 15 000  $\rightarrow$  3 500 (70 % of maxIter) -  $1.6e-6 \rightarrow 1.6e-5$  (10 $\times$  higher, less aggressive decay) -,,,,, each 2 $\times$  (V176) - 3 000  $\rightarrow$  100 000 (effectively off, V172: reset destroys short trainings) - [1K, 2K, 3K]  $\rightarrow$  [1K, 2K] (V182: degree 3 doesn't converge in 2K iter) - 1.0  $\rightarrow$  0.5

**ПРОСТЫМИ СЛОВАМИ**

any initial assessment of a newly imported image series — 2–3 min wait, after which the result is enough for a binary «is a Quality run worth it?».

**TP2** `.full`

### ПОДРОБНОСТИ

Production Quality Classic. Overrides: - 30 000 → 35 000 (V550: 40K tests Truck overtraining +10.7 % Gs at -1.3 % L1) - 15 000 → 5 000 (V310 sweet spot, V338 7K worse) - All LR<sub>s</sub> 2× (V188) - 1.6e-6 → 1.6e-5 (V45 10×) - 2e-6 → 1.1e-6 (V335) - 100 → 200 (V112) - 0.005 → 0.001 (V393) - 3 000 → 100 000 (V194 disabled, V421 confirmed) - [1K, 2K, 3K] → [2K, 5K, 8K] (V228 delayed) - 0.0 → 0.9995 (V546 HTGS, 14 % improvement) - 50 (unchanged, V546) - false → true (V438) - 0 → 20 000 (V431) - true (V443, already initializer default for `.full`)

### ПРОСТЫМИ СЛОВАМИ

any standard photo capture (object, small room, sculpture) with < 500 images. The 14 % loss improvement announced in V546 vs V438 was confirmed as 3-trial average on Horse Full.

**TP3** `.fullClassicPaper`

### ПОДРОБНОСТИ


Q1.5-A test sibling of TP2, paper-faithful Classic. Overrides vs. TP2: - 35 000 → 30 000 (paper standard) - 5 000 → 15 000 (paper: 50 % of maxIter) - 1.6e-5 → 1.6e-6 (paper default) - ,, back to paper defaults (0.05, 0.005, 0.001) - 1.1e-6 → 2e-7 (calibrated for ~1-2M Gs on Bicycle) - 200 → 100 (paper) - 0.001 → 0.005 (paper default) - 100 000 → 3 000 (paper §5.2, risky — can trigger V194 regression) - 0.9995 → 0.0 (paper has no decay) - 20 000 → 30 000 (cosine runs to 100 % of maxIter)

### ПРОСТЫМИ СЛОВАМИ

Q1.5 research experiments that need paper-magnitude Gaussian budgets (1–2 M) for Mip-Splatting tests. After the Q1.5 «closed no-win» verdict the preset remains accessible for advanced users but is not production-recommended.

**TP4** `.fullMCMC` **ПОДРОБНОСТИ**

Production Quality MCMC. Overrides vs. initializer: - 30 000 → 200 000 (V534, MCMC needs 5× more iter than Classic) - 15 000 → 160 000 (V504b 80 % of maxIter) - 1.6e-6 → 1.6e-5 - LR schedule like TP2 (all 2×) - 0.2 → 0.05 (V521b/V534: MCMC needs stronger L1 signal) - [1K, 2K, 3K] → [2K, 5K, 8K] - `.classic` → `.mcmc` - 150 000 (already in initializer, confirmed in preset) - 5e-5 (V467/V536 optimal) - 0.005 → 0.01 (V535 optimal) - 0 → 160 000 (80 % of maxIter, V497c/V502) - 3.0 (already in initializer) - true (already in initializer) - 3 000 → 200 000 (effectively off, MCMC uses reloc instead of reset)


 **ПРОСТЫМИ СЛОВАМИ**

Web delivery, object captures with detail aspirations, drone flights (although then P9 Outdoor is even better). 71 % fewer Gaussians than Classic at comparable L1.

**TP5** `.fullMCMCMip` **ПОДРОБНОСТИ**

Q1.5-D test sibling of TP4, with Mip-Splatting + paper-magnitude MCMC budget. Overrides vs. TP4:

- `mcmcMaxGaussians` 150 000 → 1 500 000 (10×, paper magnitude)
- `useMipSplatting` false → true (Mip on)

 **ПРОСТЫМИ СЛОВАМИ**

All other fields identical to TP4. Q1.5 D-PASS on Bicycle 2026-05-24 (breaks 12-iter multi-scale FAIL streak). Q1.5 final verdict 2026-05-25 still closed-no-win — Mip-Splatting gain not reproducible across 11 pair scenes. Preset remains opt-in.

**TP6** `.classicBalanced` **ПОДРОБНОСТИ**

Mid-tier Classic. Overrides vs. TP2: - 35 000 → 20 000 (V149: 20K = 30K at 33 % less time) - 20 000 → 0 (cosine runs to maxIter = 20K, no extended phase)

 **ПРОСТЫМИ СЛОВАМИ**

Standard cases with shorter wait time. V149 identified as sweet spot.

**TP7** `.mcmcPreview`

### ПОДРОБНОСТИ

MCMC diagnostic. Overrides vs. TP4: - 200 000 → 60 000 (V494b) - 160 000 → 48 000 (80 %) - 150 000 → 100 000 (V473b) - 160 000 → 40 000 (V494b) - 3.0 → 2.0 (1.4.5: Preview = lighter scaling)

### ПРОСТЫМИ СЛОВАМИ

Quickly see an MCMC result to judge whether TP4 or a Scene-Class preset is worth it.

**TP8** `.mcmcBalanced`

### ПОДРОБНОСТИ

Mid-tier MCMC. Overrides vs. TP4: - 200 000 → 120 000 (V518) - 160 000 → 96 000 (80 %) - 160 000 → 96 000 (80 %) - 3.0 → 2.5 (between Preview 2.0 and Full 3.0)

### ПРОСТЫМИ СЛОВАМИ

MCMC without the full 200K run. ~120 K iterations are a good compromise between quality and wait time.

**TP9** `.quickTest`

### ПОДРОБНОСТИ

Pure functional test. Overrides vs. initializer: - 30 000 → 1 000 - 15 000 → 500 -  $2e-6$  →  $4e-6$  (calibrated for 0.25x resolution) - 100 → 50 - 3 000 → 100 000 (off, since way too short) - 1.0 → 0.25

### ПРОСТЫМИ СЛОВАМИ

Sanity check «does training even start sensibly?». Duration < 30 s on M3 Ultra. Guaranteed to look mushy.


## Method:

**Signature:** `public func resolveMcmcMaxGaussians(initialPointCount: Int, bufferCapacity: Int) -> Int` **Defined in:**

**TECHNICAL** Single source of truth for the question «how many Gaussians can MCMC be allowed to grow to?» Computed from three inputs: the configured `T62 mcmcMaxGaussians` (with mass extinction floor 150 000 if 0), the (number of SfM init points) and the (preallocated Gaussian buffer size). Logic:

1. `base = T62 > 0 ? T62: 150_000` (the mass extinction floor protects against initializer default bugs like the 1.4.3 mass extinction incident)
2. If `T73 mcmcAutoScaleByScene && initialPointCount > 0 && T72 mcmcCapMultiplier > 0`:
  - `scaled = max(base, ceil(initialPointCount × T72))` else
3. If `bufferCapacity > 0`: `return min(scaled, bufferCapacity)`
4. Else `return scaled`

Example: Bicycle (Mip-NeRF 360, 194 photo frames) → SfM init ~156 K points, `T62 = 150 000`, `T72 = 5.32`, buffer capacity 8 M. Resolved cap =  $\min(8M, \max(150K, \text{ceil}(156K \times 5.32))) = \min(8M, 830K) = 830 K$ . That is the effective growth cap the MCMC relocation logic adheres to.

 Computes the actual maximum splat count under MCMC. Takes your setting, looks at how many points your scene starts with, and scales by the `Multiplier`, if automatic adaptation is on. So the cap adapts to the scene instead of forcing the same value for a tiny and a huge scene. You don't have to call the method yourself — the training uses it internally.

## Which field for what? (Cheat Sheet)

Goal	Fields to tweak
More detail in the distance	<code>T62 mcmcMaxGaussians</code> up, <code>T72 mcmcCapMultiplier</code> 5+
More detail overall (Classic)	<code>T1 maxIterations</code> up ( $\leq 40K$ ), <code>T2 densifyUntilIteration</code> $\leq 14$ % of T1
Reduce floaters in drone flights	<code>T43 frustumCullEnabled</code> on, <code>T20 skyMaskingEnabled</code> on, <code>T45 skyDomeEnabled</code> on
Nice sky in outdoor scenes	<code>T45 skyDomeEnabled</code> on, <code>T47 skyDomeRadiusMultiplier</code> 30–60
Smaller export file	<code>Strategy .mcmc</code> (T61), <code>T56 postTrainingCompactification</code> on, <code>T62 mcmcMaxGaussians</code> $\leq 200K$
Faster training	<code>T22 trainingRenderScale</code> 0.5, <code>T1 maxIterations</code> halved — but not both!
Better highlights	<code>T21 shDegreeUpgradeIterations</code> with [2K, 5K, 8K] (no early front-load), MCMC + 200K iter
Keep Mac responsive	<code>T25 throttleDelayMs</code> 5–10 (costs ~15 % training time)
Live preview more often	<code>T59 livePreviewInterval</code> down to 10–20
Smoother transitions in shadows	<code>T17 ssimWeight</code> slightly up (0.15–0.25), but not above 0.3
Keep interiors compact	<code>P10 Indoor preset</code> (, <code>T72 = 1.76</code> )

## Dangerous Fields

These fields can, with misconfiguration, lead to OOM, app crash, mass extinction of Gaussians, or unusable benchmark data. Handle with care:

- T11 `densifyGradThreshold` — halving can create 2–4× as many Gaussians, quickly blowing up GPU memory. Also note: must match the T22 `trainingRenderScale` (1.0× → 1e-6, 0.5× → 2e-6, 0.25× → 4e-6).
- T72 `mcmcCapMultiplier` — with large scenes with > 200 K SfM init points and a multiplier > 5 a resolved cap of millions of Gaussians arises. On 36 GB RAM Macs OOM is possible. Outdoor preset 5.32 works only because Mip-NeRF 360 Bicycle has 156 K init points → 830 K cap.
- T39 `testViewIndices` — manually setting can make the benchmark unusable (all indices > N → no holdouts). Let the `--benchmark` flag set it.
- T64 `mcmcOpacityRegWeight` **and** T65 `mcmcScaleRegWeight` — In 1.4.3 beta set to 0.01, which led to mass extinction (460 K → 5 Gaussians in one iteration). Since 1.4.4 pinned at 0.0, but manually increasing can reproduce the issue.
- T15 `opacityResetInterval` — if not 100 000+ (effectively off) and the training is shorter than 10 000 iterations, the reset destroys convergence. `.preview` therefore has it at 100 000 despite `maxIterations = 5 000`.
- T54/T55 `densifyPhase2*` — two-phase densification ended in tests in a 0-Gaussians cascade. Leave both at 0.
- T74 `useMipSplatting` — Q1.5 closed-no-win 2026-05-25, can even worsen PSNR on some outdoor scenes. Default off, opt-in only for research.

If a field is on this list and you want to change it, first back up your current preset (export as JSON) and consider whether you can reproducibly measure the result — otherwise you won't know afterwards whether you brought about an improvement or a worsening.

## ГЛАВА

## Глава 7 — Встроенные предустановки качества

---

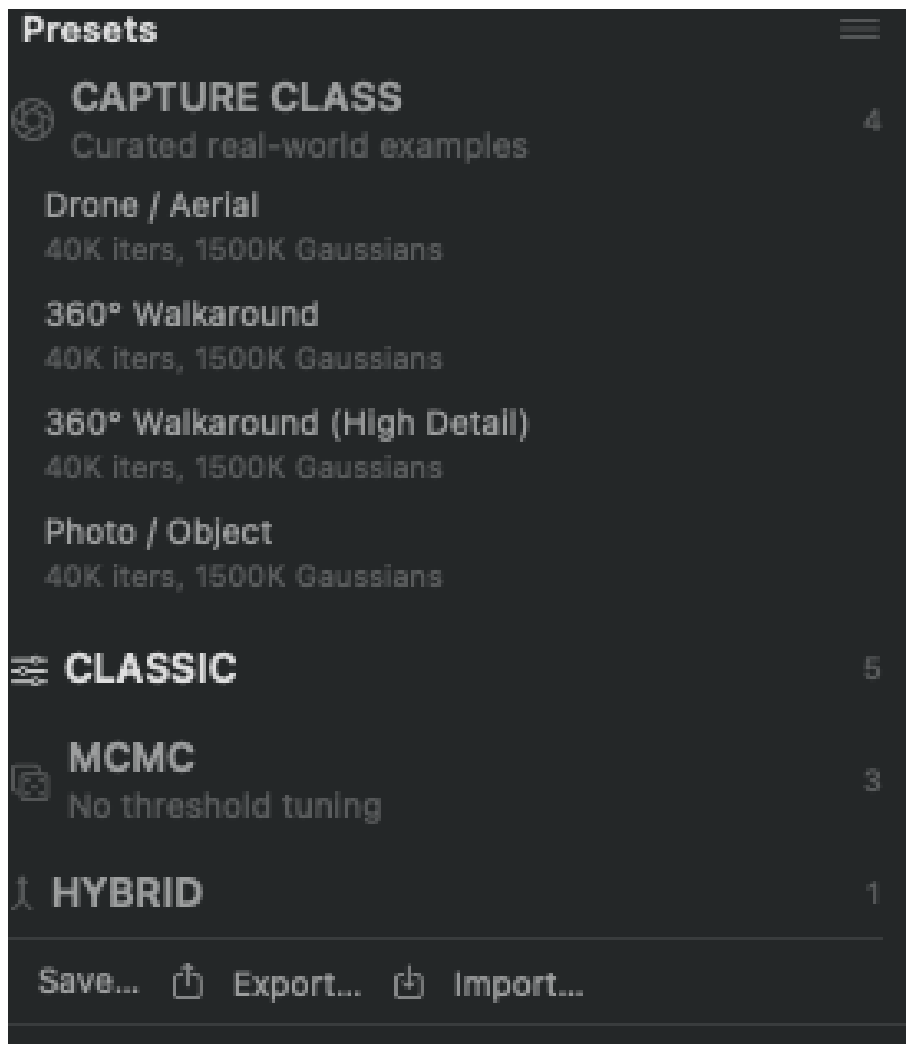


Рисунок 27. Секция Presets со всеми четырьмя развёрнутыми группами — CAPTURE CLASS (4 предустановки: Drone/Aerial, 360° Walkaround, 360° Walkaround (High Detail), Photo/Object), CLASSIC (5 предустановок: Quick/Preview/Balanced/Quality/Ultra Detail), MCMC (3 предустановки, пометка «No threshold tuning»), HYBRID (1 предустановка: Balanced (Hybrid))

**ЧТО ВИДНО НА ИЗОБРАЖЕНИИ** Секция Presets в Инспекторе, все четыре группы развёрнуты. CAPTURE CLASS с четырьмя кураторскими предустановками для реального материала (Drone / Aerial, 360° Walkaround, 360° Walkaround (High Detail),

Photo / Object) — это основная группа и единственная видимая в режиме новичка. CLASSIC с Quick (1K итераций), Preview (5K итераций, активный выбор с голубой галочкой), Balanced (20K итераций), Quality (35K итераций) и Ultra Detail (35K итераций). MCMC с подзаголовком «No threshold tuning» — MCMC не нуждается в пороге Densify-Until: Preview (60K итераций, 100K Gaussians), Balanced (120K, 150K), Quality (200K, 150K). HYBRID с предустановкой Balanced (Hybrid) (20K итераций, 150K Gaussians). Нижняя строка действий: Save..., Export..., Import...

Предустановка — это подготовленная конфигурация для обучения. RadianceKit поставляется с тринадцатью встроенными предустановками в четырёх группах: четыре **Capture-Class**-предустановки (P9–P12) — кураторские, выверенные глазом на реальном материале сообщества рецепты для реальных видов съёмки (дрон, 360°-обход, фото-объект), и основная ось с версии v1.6 —, пять классических предустановок (P1–P5: Quick/Preview/Balanced/Quality/Ultra Detail), три MCMC-предустановки (P6–P8) и одна гибридная предустановка (P13), которая объединяет стратегии Classic и MCMC. Прежние «Scene-Class»-предустановки (Render/3D, Outdoor, Indoor, настроенные в фазе Q7 байесовской оптимизацией на сценах MiP-NeRF-360 и NeRF-Blender) были убраны в v1.6 как видимая группа — выверенный глазом на реальном материале Capture-Class теперь основная ось; настроенные в Q7 конфигурации сохраняются только внутренне. Вы выбираете предустановки в боковой панели в разделе **Presets** или в режиме новичка при импорте. Кнопки **+** открывают диалоги для создания собственных предустановок рядом — тринадцать встроенных удалить нельзя, но их можно дублировать.

В режиме эксперта предустановки сгруппированы по виду съёмки и стратегии (Capture Class / Classic / MCMC / Hybrid). Щелчок по записи записывает сохранённую конфигурацию обучения в текущее состояние. Это не снимок — если затем вы покрутите ползунки, состояние изменится, а сама предустановка останется без изменений; цветная подсказка покажет «modified».

Какая предустановка подходит в каком случае, зависит в основном от типа сцены и оборудования. Три таблицы в конце главы это суммируют.

## I P1 — Quick



ГДЕ

Инспектор → секция Presets → группа «Classic» → запись «Quick». UUID-суффикс ...001 .



ТЕХНИЧЕСКИ

Диагностическая предустановка с 1 000 итераций, классическая (адаптивная) стратегия Densification и масштаб разрешения тренировки 0.25x (входное изображение уменьшается до 25 % перед тренировкой). Не предназначена для выдачи сцены, а лишь для быстрого определения того, проявляет ли установка (позы камер, облако точек, серия изображений) вообще какое-либо осмысленное движение в значениях loss. На M3 Ultra обычно менее 30 секунд на 50–200 изображениях. Малое разрешение скрывает реальное качество изображения, но удерживает расход памяти и затраты на рендеринг очень низкими. Также автоматически выбирается как значение по умолчанию при первом запуске, если система имеет менее 10 ГБ ОЗУ.

### ПРОСТЫМИ СЛОВАМИ

Быстрый тест работоспособности. Загружаете изображения, ждёте около полминуты, смотрите, появляется ли грубый контур сцены. Если изображение в просмотре выглядит как размытое пятно — это нормально, так и должно быть. Если же видны только тёмные точки или совершенно искажённая форма, скорее всего, неверны позы камер (см. главу 9). Для результата, которым можно делиться, вам потребуется минимум P2 или P3.

## I P2 — Preview (Classic)



ГДЕ

Инспектор → секция Presets → группа «Classic» → запись «Preview». UUID-суффикс ...002 .



ТЕХНИЧЕСКИ

5 000 итераций Classic Densification, масштаб разрешения 0.5x, удвоенные коэффициенты обучения по сравнению со стандартом. Densification (clone + split) активна первые 2 500 итераций, затем только pruning. Предустановка по умолчанию для систем с  $\geq 10$  ГБ ОЗУ. На M3 Ultra обычно от 90 секунд до 3 минут для сцены из 200 изображений. Даёт пригодное представление о геометрии и позах камер, но текстуры заметно мягкие — обойти масштаб 0.5x при дальнейшей тренировке с P3 или P4 напрямую не получится, поскольку коэффициенты обучения откалиброваны под половинное разрешение.

### ПРОСТЫМИ СЛОВАМИ

Стандарт для «быстро взглянуть». Если вы только что импортировали новые изображения и хотите увидеть, восстанавливается ли сцена в принципе, это правильный уровень. Подождать 2–3 минуты, потом можно покрутить в 3D-просмотре и оценить, стоит ли вкладываться в дальнейшие прогоны тренировки. Только когда результат превью уже выглядит хорошо, имеет смысл переходить к Balanced или Quality.

## I P3 — Balanced (Classic)



ГДЕ

Инспектор → секция Presets → группа «Classic»  
→ запись «Balanced». UUID-суффикс `...005`.



ТЕХНИЧЕСКИ

20 000 итераций Classic Densification на полном разрешении. Densification идёт первые 15 000 итераций, начиная с итерации 3 000 с интервалом densify 100. Эмпирически «sweet spot» из задокументированных тренировочных сессий: при классическом Densification на Horse Full и Truck L1 loss стабилизируется между итерациями 18 000 и 22 000, более долгая тренировка не приносит существенного улучшения относительно Quality (P4). На M3 Ultra обычно 30–60 секунд на 200 изображениях, 5–8 минут на 1 000+ изображениях.

### ПРОСТЫМИ СЛОВАМИ

«Хороший компромисс». Большинство сцен с этим уже выглядят хорошо, без необходимости ждать час. Если вы хотите где-нибудь показать конечный результат (соцсети, веб-сайт, демо для клиента), часто этого достаточно. Только когда нужно приближаться к splat-модели или нужны детали текстуры поверхности, имеет смысл переходить к P4 Quality или P7 MCMC.

## I P4 — Quality (Classic)



ГДЕ

Инспектор → секция Presets → группа «Classic»  
→ запись «Quality». UUID-суффикс `...003`.



ТЕХНИЧЕСКИ

35 000 итераций Classic Densification с V546 «Opacity Decay» (HTGS, Eurographics 2025): после каждого цикла densify прозрачность всех существующих Gaussians умножается на коэффициент  $< 1.0$ , что надёжно удаляет ставшие неактивными при pruning Gaussians и тем самым даёт на 14 % лучший L1 loss при том же количестве итераций, чем классический прогон на 35 000. SSIM-loss включён (`ssimWeight=0.05`). На M3 Ultra обычно 2–4 минуты на 200 изображениях. Достигает финального  $L1 \approx 0.023$  на NeRF-Blender (Lego, Chair, Drums) — лучший Classic-вариант из 560+ задокументированных экспериментов. Примечание: требуется ~3–5 ГБ памяти GPU; на системах с 8 ГБ безопасный выбор — P3.

### ПРОСТЫМИ СЛОВАМИ

Лучший классический вариант. Даёт чёткую текстуру и тонкую геометрию, особенно при сканах объектов (скульптура, стул, ваза). Однако на больших уличных сценах или комнатах разница с Balanced едва заметна — там переход к MCMC-предустановке (P6–P8) или Capture-Class-предустановке (P9–P12) оправдан больше, чем переход с P3 на P4. Кто хочет абсолютного максимума семейства Classic, берёт P5 Ultra Detail.

## I P5 — Ultra Detail (Classic)



ГДЕ

Инспектор → секция Presets → группа «Classic» → запись «Ultra Detail». UUID-суффикс `...008`.



ТЕХНИЧЕСКИ

Около 35 000 итераций Classic Densification — победитель held-out-прогона матрицы качества (2026-06-10). На всех трёх протестированных сценах Mip-NeRF-360 Ultra Detail обходит встроенную предустановку MCMC «Quality» (P8) при сопоставимом времени wall-clock примерно на +0.94 dB PSNR. Тем самым это сильнейшая Quality-предустановка группы Classic и самый резкий Classic-вариант, который поставляет RadianceKit. На M3 Ultra обычно в тех же временных рамках, что и P4 Quality (2–5 минут на 200 изображениях), но требует немного больше памяти GPU; на системах с 8 ГБ безопасный выбор — P3.

### ПРОСТЫМИ СЛОВАМИ

Самый резкий уровень Classic и held-out-победитель наших тестов качества: на реальных сценах примерно на децибел лучше MCMC-варианта «Quality» — при сопоставимом времени ожидания. Если вам нужна максимальная детализация при проверенном классическом Densification и достаточно памяти GPU, это первый выбор. Если памяти не хватает или нужен как можно меньший файл экспорта, оставайтесь на P4 Quality или MCMC-предустановке.

## I P6 — Preview (MCMC)



ГДЕ

Инспектор → секция Presets → группа «MCMC» → запись «Preview». UUID-суффикс `...006`.



ТЕХНИЧЕСКИ

60 000 итераций MCMC Densification (3DGS-MCMC, NeurIPS 2024) с ограничением 100 000 Gaussians. MCMC заменяет эвристическую логику clone/split марковской цепью Монте-Карло для перерасположения: мёртвые Gaussians перемещаются через выборку глубин со сглаживанием сигмоидой, что даёт контролируемое и воспроизводимое количество Gaussians. Ограничение жёстко ограничивает максимум на 100K — это экономит память и время рендеринга, но стоит детализации. На M3 Ultra обычно 5–8 минут на 200 изображениях. Подходит как «тест работоспособности MCMC» — помогает оценить, имеет ли смысл переход с Classic на MCMC, прежде чем вкладывать больше времени в P7 или P8.

### ПРОСТЫМИ СЛОВАМИ

Как P2 Preview, но с более новым методом MCMC. Часто даёт более компактные, более равномерно распределённые splat-облака, чем Classic-вариант. Для первой оценки сцены 5–8 минут обычно достаточно. Если результат превью нравится, следующим шагом будет P7 (Balanced) или сразу P8 (Quality MCMC).

## I P7 — Balanced (MCMC)



ГДЕ

Инспектор → секция Presets → группа «MCMC» → запись «Balanced». UUID-суффикс ...007 .



ТЕХНИЧЕСКИ

120 000 итераций MCMC с ограничением 150 000 Gaussians. Средний уровень MCMC — почти финальное число Gaussians как у P8 Quality, но только 60 % итераций. Эмпирически L1 loss в задокументированных тренировочных сессиях составляет 0.026–0.028 на Horse Full, по сравнению с P8 с 0.0246 — то есть примерно на 7 % выше, но половина времени ожидания. На M3 Ultra обычно 8–15 минут на 200 изображениях. Использует процедуру, масштабирующую эффективное ограничение Gaussians по плотности точек входного облака SfM (см. T75 в главе 6).

### ПРОСТЫМИ СЛОВАМИ

MCMC с приличной глубиной детализации, но без длительного полного прогона P8. Для большинства сцен этого достаточно, особенно когда хочется уложить MCMC-прогон в обеденный перерыв. Если памяти впритык (напр. на процессорах M только с 16 ГБ), оставайтесь здесь — P8 требует больше памяти GPU.

## I P8 — Quality (MCMC)



ГДЕ

Инспектор → секция Presets → группа «MCMC» → запись «Quality». UUID-суффикс ...004 .



ТЕХНИЧЕСКИ

200 000 итераций MCMC с ограничением 150 000 Gaussians, SSIM loss 0.05, спад шума MCMC по 80 % итераций. Лучший single-run L1 в 560+ экспериментах: 0.0238 на Horse Full, среднее по 3 прогонам 0.0246 (по сравнению с P4 0.0230 на той же сцене). MCMC даёт на 71 % меньше Gaussians (150K против ~524K) — решающий фактор, если результат планируется доставлять в веб, поскольку меньшее облако даёт заметно меньшие файлы экспорта. Время тренировки на M3 Ultra обычно 20–35 минут на 200 изображениях; на наборах из 1 000+ изображений скорее 1–2 часа. Лучший выбор, когда требуется максимальное качество изображения при минимальном итоговом размере.

### ПРОСТЫМИ СЛОВАМИ

Лучший MCMC-вариант. Даёт очень чистые, компактные splat-облака — идеален, когда результат позднее планируется встраивать как 3D-просмотрщик в веб или отправлять файлом (файл меньше, чем у P4 Quality, при сопоставимом оптическом качестве). Но нужно терпение — на больших захватах ждать больше часа. Планируйте это скорее как прогон «на ночь».

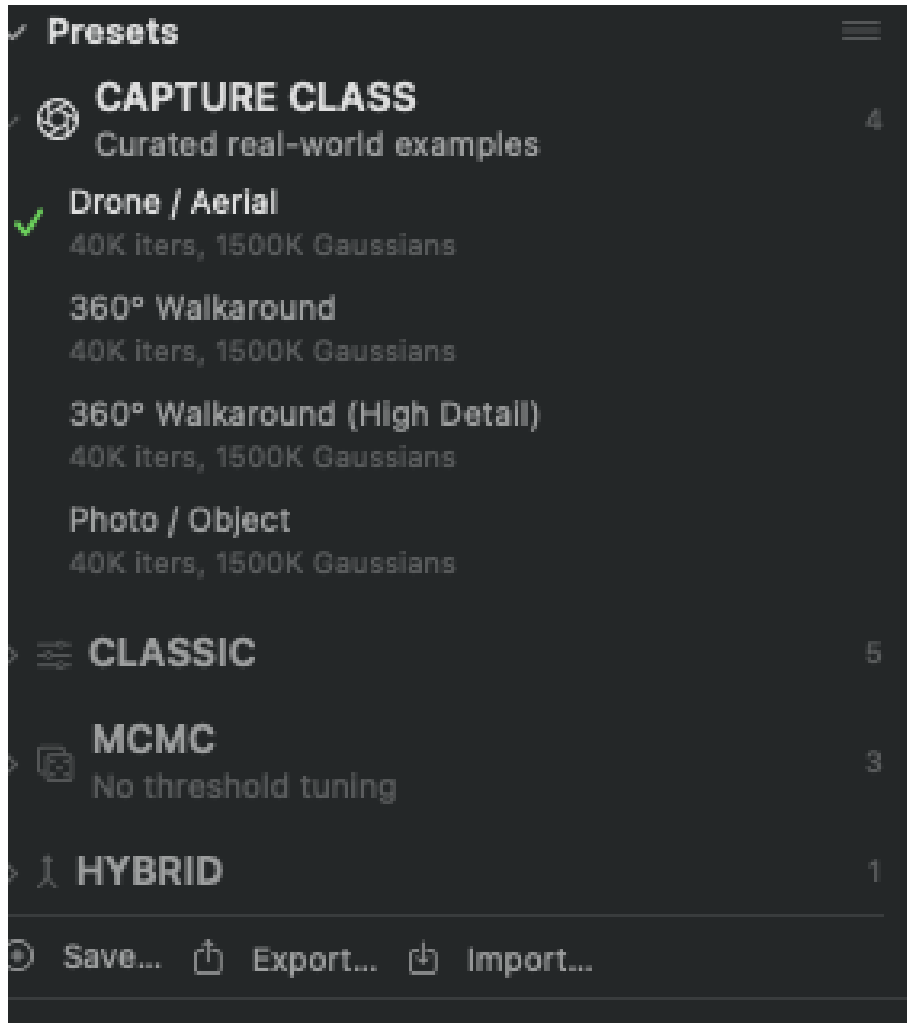


Рисунок 28. Группа CAPTURE CLASS, развёрнутая со всеми четырьмя кураторскими предустановками для реального материала — Drone / Aerial (MCMC, 40K итераций, Cap 1,5 M), 360° Walkaround (MCMC, 40K, Cap 1,5 M), 360° Walkaround (High Detail) (Hybrid, 40K, Cap 1,5 M, opt-in) и Photo / Object (Hybrid, 40K, Cap 1,5 M). Эта группа стоит в самом верху и единственная видимая в режиме новичка.

**ЧТО ВИДНО НА ИЗОБРАЖЕНИИ** Инспектор с развёрнутой группой CAPTURE CLASS — основной группой предустановок с версии v1.6, единственной отображаемой в режиме новичка. Каждая запись — это выверенный глазом на реальном материале сообщества рецепт для конкретного вида съёмки (дрон, 360°-обход, фото-объект), а не значение, оптимизированное под академический тестовый набор. Выбор щелчком записывает сохранённую конфигурацию обучения в текущее состояние.

## I P9 — Drone / Aerial



ГДЕ

Инспектор → секция Presets → группа «Capture Class» → запись «Drone / Aerial». UUID-суффикс ...010 .



ТЕХНИЧЕСКИ

Capture-Class-предустановка для воздушных и дроновых орбит вокруг зданий и ландшафтов. MCMC-Densifier, 40 000 итераций, Cap 1,5 млн Gaussians, SSIM loss 0.5 плюс edge-aware-терм 0.1. Решающим является штраф за анизотропию с весом 0.003 при пороге соотношения 6 — «убийца спагетти» против типичных игловидных артефактов, которые порождает дроновый материал. Проверено на реальном DJI-4K-полёте дрона над виадуком Pensford (проверено глазом, не только метрически).

### ПРОСТЫМИ СЛОВАМИ

Для съёмок с воздуха — полёты дрона вокруг здания, над ландшафтом, вдоль фасада. Мощный штраф за анизотропию убирает игловидные и спагетти-образные артефакты, которые охотно порождает дроновый материал. Если ваш материал снят с земли, лучше подойдёт Photo / Object или Classic-предустановка.

## I P10 — 360° Walkaround



ГДЕ

Инспектор → секция Presets → группа «Capture Class» → запись «360° Walkaround». UUID-суффикс ...011 .



ТЕХНИЧЕСКИ

Capture-Class-предустановка для видео 360°-обхода. MCMC-Densifier, 40 000 итераций, Cap 1,5 млн Gaussians, SSIM loss 0.5 плюс edge-aware-терм 0.1, мягкий штраф за анизотропию (вес 0.001 при пороге соотношения 15). Маска людей и маска неба активны. Эта предустановка ожидает 360°-equirect-видео, которое внутренне перепроецируется в перспективные кропы шириной около 90° перед началом тренировки. Проверено на 8K-360°-обходах с selfie-stick (сцена Monument, проверено глазом).

### ПРОСТЫМИ СЛОВАМИ

Для видео 360°-обхода — вы проходите с 360°-камерой или selfie-stick по комнате или вокруг объекта. RadianceKit сам разбивает сферическую панораму на обычные ракурсы и маскирует прохожих и небо. Для максимальной резкости на том же материале попробуйте дополнительно вариант High Detail (P11).

## I P11 — 360° Walkaround (High Detail)



ГДЕ

Инспектор → секция Presets → группа «Capture Class» → запись «360° Walkaround (High Detail)». UUID-суффикс ...013 (opt-in).



ТЕХНИЧЕСКИ

Opt-in-Capture-Class-предустановка для видео 360°-обхода с максимальной детализацией. Hybrid-Densifier (классические clone/split по abs-градиенту + MCMC-шум + relocation), 40 000 итераций, Cap 1,5 млн Gaussians, штраф за анизотропию 0.0015 при пороге соотношения 15, SSIM loss 0.2 и edge-aware-терм 0 — зафиксированный «r50»-screen-split-рецепт. На 360°-материале он обходит стандартную MCMC-предустановку «360° Walkaround» (P10) по PSNR, LPIPS и видимому конфетти, и это примерно при трети числа сплатов. Сознательно стоит opt-in *рядом* со стандартной 360-предустановкой, пока не проверено на большем числе сцен.

### ПРОСТЫМИ СЛОВАМИ

Более резкая альтернатива стандартной 360-предустановке (P10): больше деталей, меньше конфетти, заметно меньший файл. Сознательно стоит рядом, а не заменяет её — пока подтверждено на горстке сцен. Если ваш 360°-обход снят чисто, попробуйте сначала эту предустановку и сравните результат с P10.

## I P12 — Photo / Object



ГДЕ

Инспектор → секция Presets → группа «Capture Class» → запись «Photo / Object». UUID-суффикс ...012 .



ТЕХНИЧЕСКИ

Capture-Class-предустановка для орбит вокруг объекта из резких отдельных фотографий (не видео). Hybrid-t1-Densifier (с relocation), 40 000 итераций, Cap 1,5 млн Gaussians, SSIM loss 0.5 плюс edge-aware-терм 0.1, мягкий штраф за анизотропию (вес 0.001 при пороге соотношения 15), Opacity Decay 0.9995 каждые 50 итераций, **без** маскирования. Проверено на 163 высоко-разрешённых 41-MP-фотографиях скелета (проверено глазом). Небольшое число видов (примерно до 600) остаётся при этом ниже порога коллапса Hybrid.

### ПРОСТЫМИ СЛОВАМИ

Для съёмки объекта из резких отдельных фотографий — вы обходите скульптуру, модель, продукт с камерой и делаете фотографии, а не видео. Без маскирования, потому что у резких фотографий обычно чистый фон. Для видео-источников вместо этого возьмите 360°- или Drone-предустановку.

## I P13 — Сбалансированный (Гибридный)

### ГДЕ

Инспектор → секция Presets → группа «Гибридный» → запись «Сбалансированный (Гибридный)». UUID-суффикс `...009`.

### ТЕХНИЧЕСКИ

20 000 итераций с гибридной стратегией уплотнения: классический clone/split по градиенту размещает ёмкость там, где она нужна функции потерь, шум MCMC SGLD продолжает исследование, а мёртвые Gaussians перераспределяются вместо того, чтобы теряться при отсечении. Opacity Decay (V546) заменяет сбросы непрозрачности; штраф за анизотропию (вес 0.001, порог соотношения 15) держит игловидные сплаты под контролем. Ограничение Gaussians масштабируется со сценой (база 150K, scene-aware ×3.0). Проверено на пяти сценах против чистого MCMC при том же бюджете: в среднем +0.45 dB PSNR при на 20–30 % меньшем числе Gaussians (stonehenge +1.23, family +0.82, garden +0.47 dB). На M3 Ultra обычно 5–10 минут на 200 изображениях.

### ПРОСТЫМИ СЛОВАМИ

Сильный первый выбор для финального результата: более резкая детализация, чем у MCMC-предустановок, при столь же компактном файле и за долю времени обучения P8. Если у вас есть время только на один качественный прогон и ни один из Capture-классов явно не подходит, начните отсюда. Классические предустановки остаются лучше для быстрых тестов, а Capture-Class-предустановки (P9–P12) — первый выбор, когда ваша сцена явно соответствует одному из этих видов съёмки.

## Какая предустановка когда?

Сценарий	Первый тест	Основной прогон
Тест работоспособности новых изображений, < 30 с	<b>P1 Quick</b>	—
Орбита вокруг объекта из резких отдельных фотографий	P2 Preview	<b>P12 Photo / Object</b>
Скан одного объекта (видео), < 500 фотографий	P2 Preview	<b>P4 Quality</b> или <b>P8 Quality MCMC</b>
Видео 360°-обхода	P6 Preview MCMC	<b>P10 360° Walkaround</b> (резко: <b>P11 High Detail</b> )
Воздушная / дроновая орбита, ландшафт	P6 Preview MCMC	<b>P9 Drone / Aerial</b>
Веб-доставка (малая, компактная)	P2	<b>P8 Quality MCMC</b> (наименьший файл при полном качестве)
Резкая детализация за короткое время, компактный экспорт	P2 или P6	<b>P13 Сбалансированный (Гибридный)</b>
Максимальная детализация, стратегия Classic	P3 или P6	<b>P5 Ultra Detail</b>
Печать, маркетинг, полная детализация	P3 или P6	<b>P4 Quality (Classic)</b> или <b>P5 Ultra Detail</b>

## Быстрое сравнение

Пред-установка	Стратегия	Итер.	Макс. Gs	Масштаб рендера	Типичное время (200 изобр., M3 Ultra)	Q-Sweep
P1 Quick	Classic	1 000	∞	0.25x	~30 с	—
P2 Preview	Classic	5 000	∞	0.5x	2–3 мин	—
P3 Balanced	Classic	20 000	∞	1.0x	30–60 с	—
P4 Quality	Classic	35 000	∞	1.0x	2–4 мин	V546 HTGS
P5 Ultra Detail	Classic	~35 000	∞	1.0x	2–5 мин	Matrix Δ+0.94 dB
P6 Preview MCMC	MCMC	60 000	100 К	1.0x	5–8 мин	—
P7 Balanced MCMC	MCMC	120 000	150 К	1.0x	8–15 мин	—
P8 Quality MCMC	MCMC	200 000	150 К	1.0x	20–35 мин	V544a
P9 Drone / Aerial	MCMC	40 000	1.5 М	1.0x	10–25 мин	Глаз / Viadukt
P10 360° Walkaround	MCMC	40 000	1.5 М	1.0x	10–25 мин	Глаз / Monument
P11 360° Walkaround (High Detail)	Гибридный	40 000	1.5 М	1.0x	10–25 мин	Глаз (opt-in)
P12 Photo / Object	Гибридный	40 000	1.5 М	1.0x	10–25 мин	Глаз / Скелет
P13 Сбалансированный (Гибридный)	Гибридный	20 000	150 К	1.0x	5–10 мин	Matrix Δ+0.45 dB

## Пользовательские предустановки

Через кнопку **Save...** в секции Presets (I1 в главе 2) вы сохраняете текущую конфигурацию обучения как собственную предустановку. Пользовательские предустановки не являются «Built-in» и могут быть переименованы, экспортированы (в JSON), переданы перетаскиванием, дублированы и удалены. Тринадцать встроенных предустановок P1–P13 остаются нетронутыми кнопкой удаления.

**Правило большого пальца:** Если вы меняете что-то в предустановке, что захочется чаще — Sky-Dome on, более высокий вес SSIM для конкретного класса сцены, иные количества итераций — то сохраните этот вариант как собственную предустановку. Так в следующий прогон вы сразу будете знать, что это конфигурация, отклоняющаяся от стандартной.

## ГЛАВА

## Глава 8 — Форматы экспорта

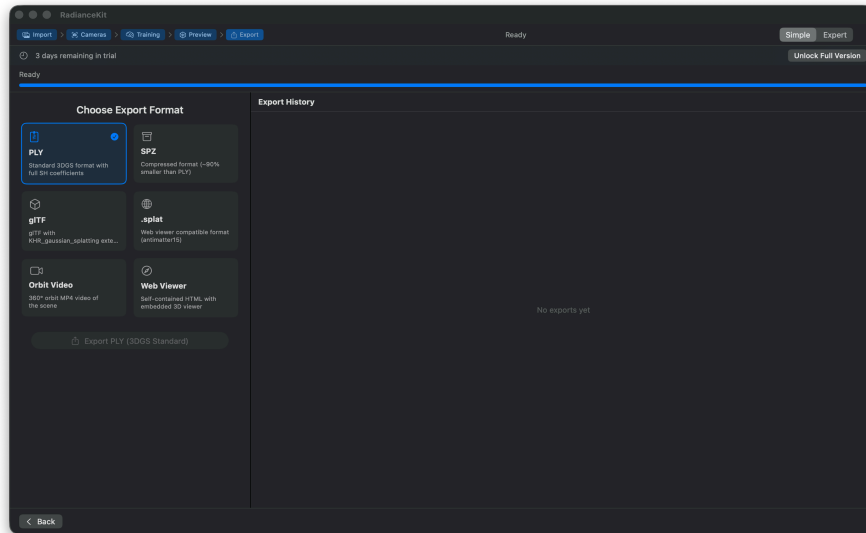


Рисунок 29. Выбор формата экспорта в режиме новичка — шесть карточек формата

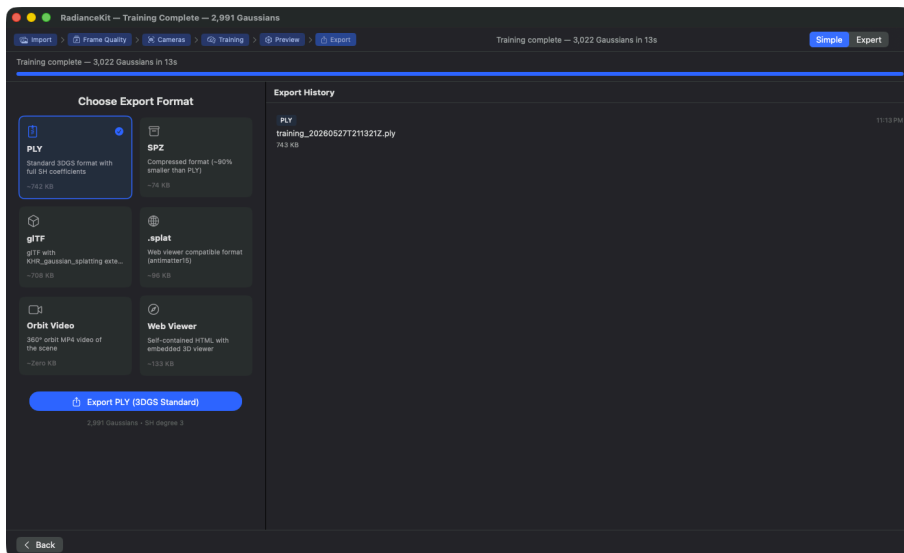


Рисунок 30. Сетка форматов экспорта в реальном времени после 5K-итерационного обучения на букете цветов — все шесть карточек с динамическим расчётом размера (PLY 742 KB выбрано, SPZ 74 KB, glTF 708 KB, .splat 96 KB, Orbit Video ~Zero KB, Web Viewer 133 KB), история экспортов справа с уже сохранённым PLY

**Что видно на изображении (2 991 Gaussians, SH-степень 3, синтетический букет из Blender как чистый по IP тестовый набор):** Размеры под каждой

карточкой формата рассчитываются в реальном времени из текущего количества Gaussians и накладных расходов формата — не зашиты жёстко. Из 2 991 Gaussians (SH-степень 3) получается 742 KB PLY, 74 KB SPZ (примерно в ~10x меньше за счёт квантования), 708 KB glTF (с расширением `KHR_gaussian_splatting`, поэтому практически эквивалентно PLY), 96 KB `.splat` (сжатый формат 24 байта на Gaussian). Orbit Video показывает «~Zero KB», потому что размер известен только после MP4-кодирования. Web Viewer (133 KB) упаковывает автономный HTML-файл со встроенным WebGL-просмотрщиком и сжатыми `splat`-данными — больше, чем чистый `.splat`, из-за накладных расходов на просмотрщик. История экспортов справа перечисляет уже выполненный PLY-экспорт («`training_20260527T211321Z.ply`, 743 KB, 23:13») со значком формата и действием `Reveal-in-Finder`.

Законченное обучение даёт облако Gaussians — набор из нескольких сотен тысяч или миллионов 3D-распределений Гаусса, которые вместе восстанавливают сцену. RadianceKit умеет десятью способами записать это облако на диск. Шесть из них — это чистые 3D-форматы данных (PLY, Compressed PLY, SPZ, SOG, glTF, `.splat`), один упаковывает облако вместе с готовым HTML-просмотрщиком (Web Viewer), один отрисовывает MP4-файл по орбитальной траектории камеры (Orbit Video), а два не экспортируют содержимое Gaussian, а только результат SfM (позы камер и грубое облако точек) для повторного использования в других конвейерах обучения (`transforms.json` + COLMAP Workspace).

Какой формат подходит в каком случае, зависит от цели. Для архивирования полных данных без потери качества берётся PLY. Для веб-просмотрщиков на собственной странице обычно достаточно `.splat` или встроенного Web Viewer. Если файл должен быть минимальным, стоит брать SPZ или SOG. Для повторного использования результата SfM в Nerfstudio, Postshot или Brush правильные пути — `transforms.json` и COLMAP Workspace.

Все функции экспорта находятся в меню «Export», а также в режиме новичка на последнем шаге мастера. Большинство форматов полностью совместимы с песочницей и работают в версии для App Store. Только SOG требует внешнего бинарного файла (`cwebp`), который не обязательно присутствует в сборке для App Store — подробности см. в E4.

## I E1 — PLY (.ply)



ГДЕ

Строка меню → Export → 3D Formats → Export PLY... (⌘E). Режим новичка: шаг мастера Export → карточка формата «PLY». **Размер:** обычно 100 % (опорное значение). **Совместимо с:** SuperSplat, PolyCam, все 3DGS-просмотрщики.



ТЕХНИЧЕСКИ

PLY — канонический формат хранения для 3D Gaussian Splatting. RadianceKit записывает двоичный little-endian файл со стандартизированной раскладкой свойств 3DGS: на один Gaussian — трёхкомпонентная позиция, три нормали всегда равны нулю, три DC-SH-коэффициента (  $f_{dc_0..2}$  ) для базового RGB-цвета, далее до 45 дополнительных SH-коэффициентов (  $f_{rest_0..44}$  ) в транспонированном channel-major-расположении, определённом статьёй Kerbl 2023 (сначала все коэффициенты канала R, затем все G, затем все B), за которыми следуют logit-непрозрачность (сырые значения до сигмоиды), три log-space-масштаба и кватернион поворота wxyz. Максимальная экспортируемая SH-степень ограничивается минимумом из пожелания пользователя и реально обученной степени; по умолчанию — 3 (45 rest-коэффициентов). Перед записью размер полезной нагрузки рассчитывается в 64-битном целочисленном виде, чтобы поймать переполнение на экстремально больших облаках. Файл пишется атомарно, что на больших облаках кратковременно занимает удвоенный объём диска.

### ПРОСТЫМИ СЛОВАМИ

Это «оригинальный файл». Самый большой файл, самая высокая совместимость, без потерь. Если не знаете, какой формат брать, берите PLY — он открывается почти в любом 3DGS-инструменте. Для 1 миллиона Gaussians это от 200 до 800 МБ в зависимости от SH-степени. Если файл становится слишком большим, посмотрите на E2 (compressed PLY) или E3 (SPZ).

## I E2 — Compressed PLY (.ply)

### ГДЕ

Строка меню → Export → 3D Formats → Export Compressed PLY.... Режим новичка: карточка формата «Compressed PLY». **Размер:** прим. 10–20 % от PLY (5–10-кратное сжатие). **Совместимо с:** SuperSplat, движок PlayCanvas, веб-просмотрщики.

### ТЕХНИЧЕСКИ

Вариант PlayCanvas формата PLY с chunked-квантованием. Gaussians группируются в чанки по 256. На чанк — min/max-границы для позиции, масштаба и цвета хранятся отдельно в заголовке; отдельные Gaussians ссылаются на свои значения относительно этих границ и сжимаются в 32 бита: позиция и масштаб с упаковкой 11-10-11 бит, поворот — как «Smallest-Three»-кватернион 2-10-10-10 бит, цвет — 8-8-8-8 RGBA. Высшие SH-коэффициенты квантуются с всего 8 битами на компоненту ( `shCoeffCount * 3 uchar` на Gaussian). Сам формат всё ещё — ASCII-заголовок PLY и поэтому в принципе валидируется PLY-инструментами, но vertex-properties объявляются полями `uint`. SH-степень по умолчанию — 0 (нет rest-коэффициентов), чтобы максимизировать сжатие — более высокие SH-степени можно выбрать явно.

### ПРОСТЫМИ СЛОВАМИ

Экономящий место вариант PLY. Совместимость с движками такая же, как у обычного PLY, но в 5–10 раз меньше. SuperSplat и PlayCanvas читают его нативно. Для веб-доставки почти всегда лучше, чем обычный PLY. Потеря качества из-за квантования обычно визуально незаметна, если сцена не содержит экстремально высокочастотных деталей.

## I E3 — SPZ (.spz)



ГДЕ

Строка меню → Export → 3D Formats → Export SPZ... Режим новичка: карточка формата «SPZ».

**Размер:** прим. 10 % от PLY (на 90 % меньше).

**Совместимо с:** Niantic Scaniverse, Niantic Spatial Fields, MetalSplatter.



ТЕХНИЧЕСКИ

Формат SPZ v2 от Niantic. Положения упакованы как 24-битный fixed-point (что даёт разрешение около 0.25 мм), масштабы — как 8-битное квантование в логарифмическом пространстве, повороты — как 8-битный Smallest-Three (в v2 хранится только  $xyz$ ,  $w$  выводится в декодере из нормы кватерниона), непрозрачности — как сигмоидированные 8-битные значения. DC SH хранится по специальной SPZ-формуле упаковки ( $dc_{raw} * 0.15 * 255 + 0.5 * 255$ ), высшие SH-полосы — 5 битами (полоса 1) или 4 битами (полосы 2–3) на коэффициент. Весь упакованный двоичный блок затем сжимается стандартным `gzip` (RFC 1952), что даёт `gzip`-контейнер с magic-байтами `1f 8b`. RadianceKit вызывает для этого системный `gzip`, поскольку встроенный `zlib-API` Apple генерирует фирменное обрамление Apple, несовместимое с SPZ-читалками в Spatial Fields или MetalSplatter. Системный `gzip` к тому же остаётся запускаемым внутри песочницы macOS.



ПРОСТЫМИ СЛОВАМИ

Наименьший стандартный файл. Если вы знаете Scaniverse от Niantic — это и есть формат, который использует то приложение. Очень маленький, очень удобный для мобильных приложений. Напрямую пригоден для облачного просмотра Niantic (Spatial Fields). Примерно на 90 % меньше PLY с теми же данными, при этом для большинства сцен визуально едва отличим.

## I E4 — SOG (.sog)



ГДЕ

Строка меню → Export → 3D Formats → Export SOG.... Режим новичка: карточка формата «SOG». **Размер:** прим. 5–6 % от PLY (15–20-кратное сжатие — самый маленький вариант). **Совместимо с:** движок PlayCanvas, редактор SuperSplat.



ТЕХНИЧЕСКИ

«Spatially Ordered Gaussians» — формат PlayCanvas, который хранит облако в формате, готовом для GPU, в нескольких WebP-изображениях без потерь. Сначала все Gaussians сортируются пространственно через 3D-код Мортонa (30-битный Z-порядок, по 10 бит на ось), что даёт изображениям кэш-локальность в рендере. Затем позиции квантуются в 16-битные значения с симметричным логарифмическим преобразованием (для лучшего динамического диапазона) и разделяются на два RGBA-изображения ( `means_l.webp` для младших 8 бит, `means_u.webp` для старших). Повороты кодируются как Smallest-Three с 3×8-битами плюс 2-битный режим в RGBA-изображении (режим попадает в альфа как `252 + largest`). Масштабы и DC SH каждый квантуются с кодовой книгой на 256 записей (распределённой по перцентильям всех значений), индексы попадают в `scales.webp` и `sh0.webp`. Пять изображений плюс `meta.json` с кодовыми книгами и границами упаковываются в ZIP-файл (собственный энкодер, поскольку песочница блокирует системный `zip`) и сохраняются с расширением `.sog`.

**Предупреждение о песочнице:** SOG — единственный вариант формата, требующий внешний бинарный файл. Этап WebP-энкодера вызывает `cwebp` из `/usr/local/bin/cwebp` или `/opt/homebrew/bin/cwebp`. Если бинарный файл `cwebp` не найден, код откатывается к необработанному PNG-кодированию — но: **PNG-fallback не работает в SuperSplat**. В версии для App Store доступность оценивается по варианту сборки; в варианте для разработчиков `cwebp` должен быть установлен через Homebrew (`brew install webp`).

### ПРОСТЫМИ СЛОВАМИ

Самый маленький 3DGS-формат в целом, заметно меньше SPZ. Но: нужен инструмент `cwebp` на вашем Mac, потому что RadianceKit сам не умеет генерировать все форматы изображений. Установите его один раз через Homebrew, потом всё работает. В версии для App Store, возможно, не полностью функционален — если из экспорта получается PNG вместо WebP, файл нельзя открыть напрямую в SuperSplat. Тем, кто работает без Homebrew, стоит брать вместо этого SPZ (E3).

## I E5 — glTF (.glb)



ГДЕ

Строка меню → Export → 3D Formats → Export glTF... Режим новичка: карточка формата «glTF».

**Размер:** сопоставим с PLY. **Совместимо с:** glTF-просмотрщиками с расширением KHR\_gaussian\_splatting (черновой стандарт Khronos).



ТЕХНИЧЕСКИ

Записывает самодостаточный двоичный файл `.glb` (без отдельного приложения bin-файла) по спецификации расширения KHR\_gaussian\_splatting. Позиции хранятся как обычные glTF POSITION vertex-data (float3), все остальные атрибуты (поворот как float4, масштаб как float3, непрозрачность как float, SH-коэффициенты как float3 × shCoeffCount) находятся в дополнительных vertex-атрибутах и ссылаются через расширение. Важно: glTF использует правую систему координат с Y-up, тогда как COLMAP/3DGS работает в Y-down/Z-forward. Поэтому экспортёр применяет поворот на 180 градусов вокруг оси X — позиции переписываются как  $(x, -y, -z)$ , кватернионы корректируются как  $(w, x, -y, -z)$ . Это даёт геометрически корректное, правильно хиральное (не зеркальное) отображение в glTF-просмотрщиках. JSON- и двоичные чанки выровнены по 4-байтовой границе, как требует стандарт GLB.



ПРОСТЫМИ СЛОВАМИ

Официальный формат стандарта Khronos для 3D-данных, в свежем расширении для Gaussian Splats. Преимущество: glTF распространён во всех крупных 3D-движках (Babylon.js, Three.js, Unity, Unreal). Недостаток: расширение в 2026 году ещё находится на стадии черновика, многие просмотрщики его пока не читают. Имеет смысл в основном, если вы интегрируете splat-данные в существующий glTF-конвейер или пишете просмотрщик, уже умеющий работать с glTF.

## I E6 — Splat (.splat)

### ГДЕ

Строка меню → Export → 3D Formats → Export .splat... Режим новичка: карточка формата «.splat». **Размер:** ровно 32 байта на Gaussian. **Совместимо с:** gsplat.js, веб-просмотрщики (эталонная реализация antimatter15), большинство браузерных 3DGS-демо.

### ТЕХНИЧЕСКИ

Формат `.splat` от antimatter15 — 32 байта на Gaussian, без заголовка, без косвенности. Раскладка на запись: 3 × float32 позиция (мировые координаты), 3 × float32 масштаб (экспреобразование из логарифмического пространства внутреннего буфера), 4 × uint8 RGBA-цвет (DC-SH-коэффициент, масштабированный с `SH_C0 = 0.282...` и обрезанный в `[0,255]`), 4 × uint8 кватернион (w,x,y,z, нормализованный и закодированный в байтовом диапазоне как `128 + 128*q`). Хранится только DC SH — высшие SH-полосы отбрасываются. Это делает формат чрезвычайно компактным, но стоит зависящих от направления взгляда изменений цвета, возникающих при отражениях или бликах. Порядок записи соответствует индексному порядку облака (без пространственной сортировки), веб-просмотрщики вроде `gsplat.js` рендерят, исходя из этого предположения.

### ПРОСТЫМИ СЛОВАМИ

Формат выбора, если хотите отображать splat в собственном веб-просмотрщике с `gsplat.js`. Очень компактный (32 байта/Gaussian), но без высших SH-степеней — значит, без блестящих отражений или едва уловимых изменений цвета в зависимости от угла обзора. Для большинства веб-применений не проблема, поскольку DC-цвета вполне достаточно, а отсутствие зависимости от угла обзора едва заметно.

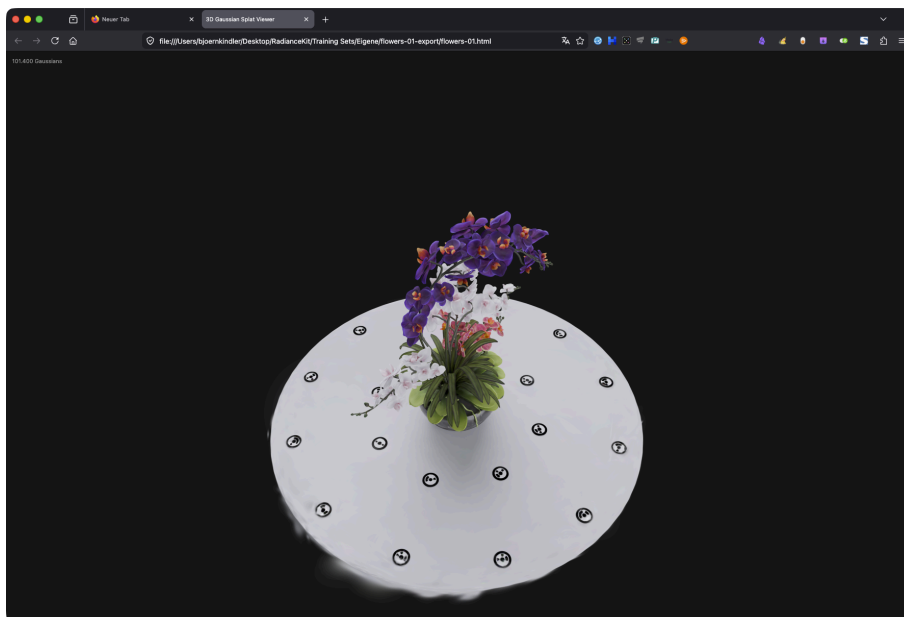


Рисунок 31. Web Viewer, открытый в Firefox — splat букета отрендерен с окружающими сферами-маркерами камер, видна верхняя панель браузера, не нужна настройка CDN/сервера. Автономный `flowers-01.html` открыт напрямую из Finder двойным щелчком в браузере по умолчанию — встроенная программа WebGL2 немедленно рендерит облако *Gaussian*, без сети или сервера. Чёрные маркеры вокруг букета — тренировочные камеры, опционально переключаемые. Drag мышью вращает, scroll приближает.

## | E7 — Web Viewer (.html)



ГДЕ

Строка меню → Export → Media → Export Web Viewer... Режим новичка: карточка формата «Web Viewer». **Размер:** splat-данные в base64-кодировке ( $\approx 4/3$  накладных) + прим. 5 KB HTML/JS-оболочка. **Совместимо с:** любым современным браузером с WebGL2 (все десктопы, iOS 15+, Android 5+).



ТЕХНИЧЕСКИ

Упаковывает облако Gaussian вместе с полностью встроенным WebGL2-рендерером в один файл `.html`. Нет зависимостей от CDN, нет WASM, нет второго файла. Облако внутри сначала кодируется как двоичный `.splat` (та же 32-байтовая логика, что у E6), потом встраивается в base64, потом декодируется `atob` в браузере. Встроенный рендерер делает собственную WebGL2-сортировку, `mouse-orbit-controls` и CPU-сортировку каждый кадр; весь JS-код (шейдер, математика, цикл) виден в выходном HTML. Соглашение об осях на границе хранения и рендерера — ровно такое же, как у E5: позиция `(x, -y, -z)`, кватернион `(w, x, -y, -z)`. Опционально может отображаться брендинг-оверлей (переключатель `free-tier`). Поскольку всё inline, файл работает и напрямую с протокола `file://` — никакого локального веб-сервера для теста не нужно.

### ПРОСТЫМИ СЛОВАМИ

Один HTML-файл, который можно отправить кому-нибудь по почте или встроить на сайт. Двойной щелчок в Finder — и браузер показывает вашу сцену с поворотом мышью. Не нужно загружать в облако, нет второго файла, нет сервера. Идеально для презентаций клиентам, портфолио, вложений в почту. Недостаток: файл становится примерно на треть больше, чем чистый `.splat`, из-за base64-кодирования — для очень больших сцен поэтому имеет смысл размещать `.splat`-файл отдельно вместе со стандартным просмотрщиком.

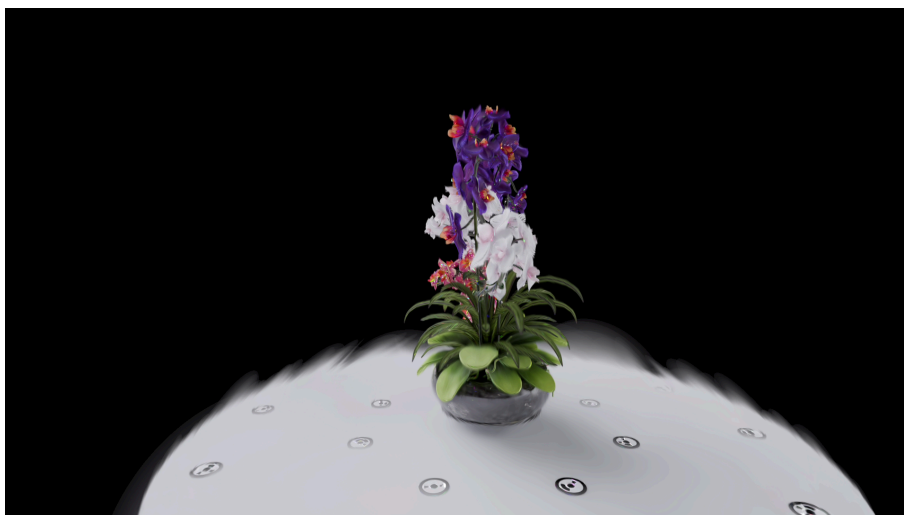


Рисунок 32. Один кадр, извлечённый из *flowers-01.mp4* — букет в профильном рендере, белая площадка с маркерами камер видна, чёрный фон (стандартный фон viewport, изменяемый в настройках). Камера движется вокруг сцены по параметрической траектории (наклон + расстояние фиксированы, yaw вращается), длительность обычно 6–10 секунд при 30 или 60 fps. Разрешение кадра масштабируется от 480p до 8K через VideoPreset.

## | E8 — Orbit Video (.mp4/.mov)



ГДЕ

Строка меню → Viewport → Record Turntable Video ИЛИ строка меню → Export → Media → Export Orbit Video.... Режим новичка: карточка формата «Orbit Video» с ползунком длительности 3–30 с. **Размер:** зависит от длительности, разрешения, битрейта. **Совместимо с:** всеми платформами (H.264 и HEVC — стандарт Apple).



ТЕХНИЧЕСКИ

Рендерит облако Gaussian вдоль параметрической орбитальной траектории камеры и кодирует каждый кадр через AVAssetWriter в файл MP4 или MOV. Орбитальная конфигурация управляет скоростью вращения (revolutions), расстоянием, наклоном, FOV, длительностью и фактором ease-in/out. Экспорт orbit-видео проходит через СОБСТВЕННЫЙ ForwardPass RadianceKit с полным вычислением SH — пиксель-в-пиксель идентично viewport в приложении (WYSIWYG). На кадр матрица выравнивания мира (рассчитанная рендерером для поворота внутренних координат в мир орбиты с Y-up) умножается на камеру, после чего применяется переворот соглашения камеры (camFlip: orbit Y-up → COLMAP Y-down). Цель оффскрин-рендеринга вытаскивается через IOSurface в CVPixelBuffer для энкодера. Энкодер поддерживает H.264 и HEVC, настраиваемый битрейт и разрешение от 480p до 8K. Перед первым кадром рендерер ждёт 200 мс, чтобы завершилась начальная сортировка splat. Этот экспорт ограничен GPU — при 8K и миллионах Gaussians время рендера на кадр составляет несколько секунд, так что суммарное время рендера 10–30 минут для 6 с видео вполне возможно.

### ПРОСТЫМИ СЛОВАМИ

Готовый MP4-файл с поворотом вокруг вашей сцены. Идеален для соцсетей, маркетинга, презентаций. Можно задать длительность (3–30 секунд), направление и скорость вращения. Файл встраивается напрямую на YouTube, в Instagram, в PowerPoint и куда угодно ещё. Иногда медленно, потому что приложение должно полностью отрисовать каждый кадр — на 8K видео можно планировать от пяти до тридцати минут в зависимости от числа Gaussians.

## I E9 — SfM Transforms (transforms.json)

### ГДЕ

Строка меню → Export → Photogrammetry → Export SfM (transforms.json).... **Размер:** обычно 1–10 KB (только позы + intrinsics, без изображений, без Gaussians). **Совместимо с:** nerfstudio, Brush, gsplat, OpenSplat, Meshroom, со всеми современными feed-forward 3DGS-тренировщиками.

### ТЕХНИЧЕСКИ

Записывает формат `transforms.json` от nerfstudio со списком поз камер плюс общие intrinsics. На камеру view-матрица (внутреннее представление RadianceKit: World-to-Camera в соглашении COLMAP) инвертируется, после чего камеро-локальные векторы базиса Y и Z зеркалируются для конвертации в соглашение nerfstudio (стиль OpenGL, камера смотрит вдоль `-Z`, `+Y` направлено вверх). Финальная матрица 4×4 попадает как row-major вложенный массив double в поле `transform_matrix` каждого кадра. Intrinsics хранятся на верхнем уровне (focal length x/y, principal point x/y, ширина/высота изображения, camera\_model = "OPENCV", плюс коэффициенты искажения `k1`, `k2`, `p1`, `p2`) — кроме случая, когда экспортёр обнаруживает несколько разных наборов intrinsics, тогда они пишутся на кадр. Пути изображений пишутся как `images/<filename>` относительно JSON-файла; пользователь должен создать соседнюю папку `images/` с тренировочными фотографиями.

### ПРОСТЫМИ СЛОВАМИ

Этот JSON-файл описывает для каждого фото, где стояла камера и куда смотрела. Сам по себе файл маленький и бесполезен — он используется вместе с исходными изображениями в одной папке. Nerfstudio, Brush и пара других тренировщиков читают ровно этот формат, и с ним можно передать ваши SfM-результаты RadianceKit в другой инструмент, без того чтобы там пришлось заново рассчитывать камеры. Экономит часы на больших сценах.

## I E10 — COLMAP Workspace (sparse/0/)

### ГДЕ

Строка меню → Export → Photogrammetry → Export SfM (COLMAP Workspace).... **Размер:** три двоичных файла вместе обычно 4–8 МВ — `points3D.bin` доминирует (одна строка на каждую 3D-точку sparse-облака), `images.bin` и `cameras.bin` — каждый заметно меньше 100 КВ. **Совместимо с:** самим COLMAP, Nerfstudio, Postshot, Meshroom, всеми инструментами, ожидающими директорию COLMAP `sparse/`.

### ТЕХНИЧЕСКИ

Записывает стандартную раскладку COLMAP `sparse/0/` с тремя двоичными файлами: `cameras.bin`, `images.bin`, `points3D.bin`. Эталон формата — официальная документация COLMAP. `cameras.bin` содержит дедуплицированный список `intrinsics` (камеры с идентичными `intrinsics` + размером изображения объединяются в одну запись); используемая модель камеры — `OPENCV` (model 4), с `fx/fy/cx/cy` плюс четыре коэффициента искажения `k1/k2/p1/p2`. `images.bin` перечисляет на каждое изображение позу как `wxyz`-кватернион плюс перемещение, за которым следует ID камеры и имя файла; 2D-3D-соответствия не сохраняются. `points3D.bin` содержит SfM-облако точек с позицией, цветом (0–255 RGB) и значениями по умолчанию для репроекции и длины треков. Всё пишется в `little-endian`. Повторный импорт в RadianceKit работает через меню `File` → «Import COLMAP/Metashape Workspace...» (см. Q3 в главе про SfM-бэкенды).

### ПРОСТЫМИ СЛОВАМИ

Официальный формат COLMAP. Если хотите продолжить обучение в Postshot, Nerfstudio или другом ПО, умеющем работать с COLMAP, это путь. Три маленьких файла плюс ваши исходные изображения, и целевая программа принимает это так, будто исходной программой был сам COLMAP. Этот формат понимает больше программ, чем формат `transforms.json` (E9), но он чуть менее удобен, потому что двоичный, а не текстовый.

## Какой формат когда?

Цель	Формат
Веб-просмотрщик на своей странице	E7 Web Viewer (.html)
Веб-просмотрщик с <code>gsplat.js</code>	E6 Splat (.splat)
Повторное использование в конвейере Postshot / Nerfstudio	E9 transforms.json + E10 COLMAP Workspace
Редактирование в SuperSplat	E1 PLY или E2 Compressed PLY
Niantic Scaniverse / Spatial Fields	E3 SPZ
Максимальное сжатие	E4 SOG (требуется swebp)
Маркетинговое/соцсетевое видео	E8 Orbit Video

## Быстрое сравнение

Формат	Расширение	Песочница	Размер (1M Gauss)	Лучшее применение
E1 PLY	<code>.ply</code>	да	~250 MB	Архив, наивысшая совместимость
E2 Compressed PLY	<code>.ply</code>	да	~40 MB	Веб + SuperSplat
E3 SPZ	<code>.spz</code>	да (gzip spawn)	~40 MB	Niantic + мобильные
E4 SOG	<code>.sog</code>	условно (сwebp)	~20 MB	Максимальное сжатие
E5 glTF	<code>.glb</code>	да	~250 MB	Конвейер Khronos
E6 Splat	<code>.splat</code>	да	~32 MB	gsplat.js веб-просмотрщик
E7 Web Viewer	<code>.html</code>	да	~45 MB	Автономный браузерный файл
E8 Orbit Video	<code>.mp4 / .mov</code>	да	переменно	Соцсети/маркетинг
E9 SfM Transforms	<code>.json</code>	да	~5 KB	Передача поз
E10 COLMAP Workspace	Директория	да	~4–8 MB	Передача поз двоично

Колонка размера — грубые опорные значения для 1 миллиона Gaussians с SH-степенью 3. Реальные значения варьируются в зависимости от сжимаемости сцены; SH-степень 0 уменьшает PLY/glTF в 4 раза.

## ГЛАВА

## Глава 9 — SfM-бэкенды

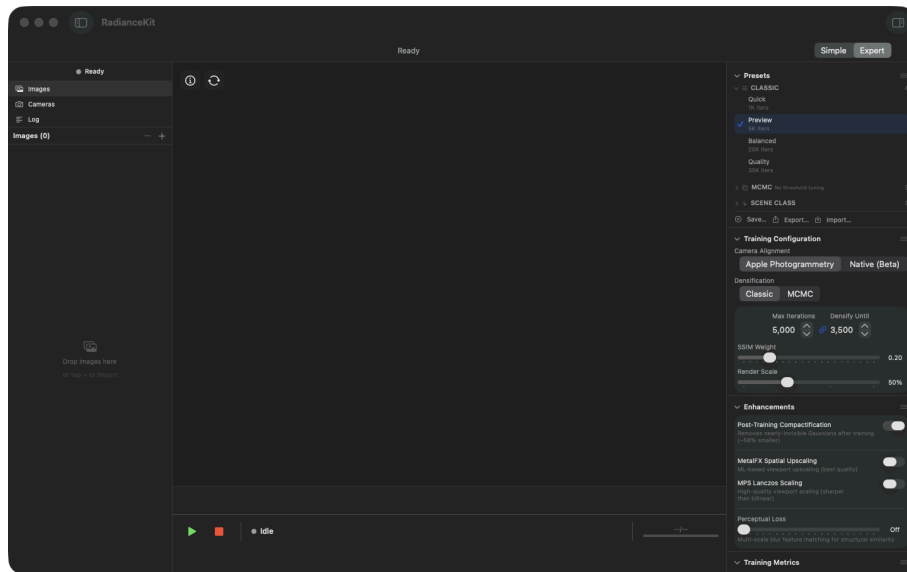


Рисунок 33. Режим эксперта с переключателем Camera Alignment в Инспекторе (Apple Photogrammetry / Native (Beta))

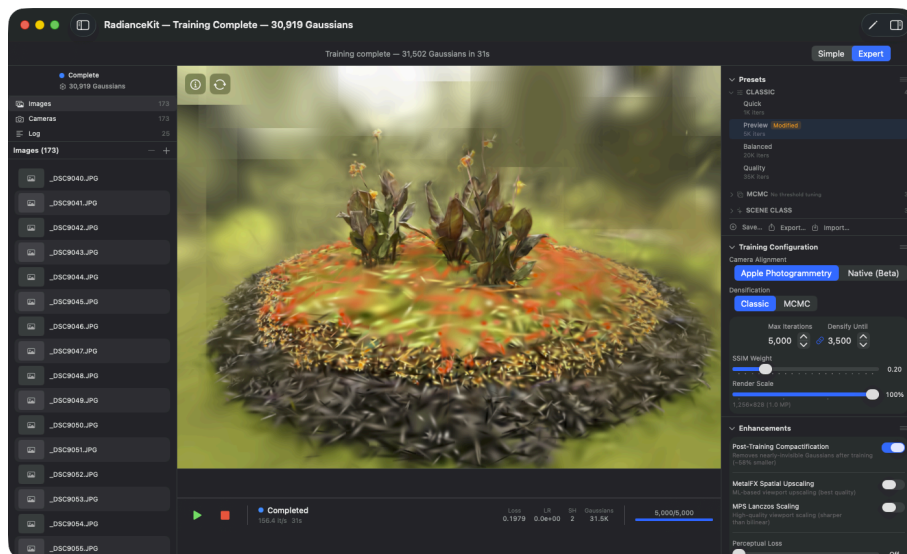


Рисунок 34. Инспектор с активной опцией Native (Beta) — выбран второй вариант переключателя Camera Alignment, все остальные параметры конфигурации обучения без изменений

**ЧТО ВИДНО НА ИЗОБРАЖЕНИИ** Переключатель Camera Alignment в Инспекторе — это сегментированный элемент управления с двумя вариантами — Apple

Photogrammetry (значение по умолчанию для сборок App Store, полностью совместимо с песочницей) и Native (Beta) (собственный бэкенд RadianceKit на базе FAST+BRIEF+GLOMAP, разработанный в фазах 3.8/3.9, по состоянию на 2026-05). Native (Beta) проверен только на орбитальных съёмках и быстрее Apple Photogrammetry при  $\geq 1\,000$  кадров, но пока не проходит порог качества фазы 3 §5 ( $\text{finalLoss} \leq 0.0115$ ) — отсюда и тег Beta. Внешние результаты SfM из Metashape, COLMAP или любого другого фотограмметрического ПО можно дополнительно импортировать через меню File (текстовый формат Q3 COLMAP, импорт рабочей области Q6) — сам переключатель при этом не меняется, но импортированные позы заменяют результат SfM.

SfM расширяется как **Structure from Motion**. Из набора перекрывающихся фотографий программа восстанавливает для каждого изображения положение и направление взгляда камеры в общей 3D-системе координат. Попутно создаётся грубое 3D-облако точек, которое инициализирует обучение Gaussian Splatting. Результат SfM является входом для собственно обучения и решающим образом определяет конечное качество изображения.

RadianceKit предлагает пять путей SfM: два бэкенда, встроенных в само приложение (Q1 Apple Photogrammetry, Q4/Q5 Native), два пути импорта из внешних инструментов (Q3 текстовый формат COLMAP, Q6 двоичный импорт Workspace), а также Q2 двоичный COLMAP, доступный только в сборках для разработчиков вне App Store. Какой из них подходит, зависит от типа сцены (орбита вокруг объекта, помещение, полёт дрона) и от того, предоставляет ли уже внешнее ПО готовую реконструкцию.

## I Q1 — Apple Photogrammetry



ГДЕ

Режим эксперта → Инспектор → Training Configuration → переключатель Camera Alignment, запись «Apple Photogrammetry».



ТЕХНИЧЕСКИ

Обёртка над встроенным фреймворком фотограмметрии Apple, изначально разработанным для Object Capture. Внутри Apple извлекает признаки своим проприетарным конвейером (этапы публично не документируются), проверяет их многоакурсным сопоставлением и решает bundle adjustment на Apple Silicon Neural Engine + GPU. Бэкенд полностью совместим с App Store (никакого внешнего бинарного файла, Sandbox=true, on-device), но выдаёт только позы камер плюс грубое облако точек — никаких диагностических метрик вроде длины треков или ошибки репроекции. Масштабируется согласно рекомендации Apple до нескольких сотен изображений. При количестве кадров более ~500 в линейных пролётах дронов или больших уличных сценах наблюдались воспроизводимые сбои или тихий пропуск отдельных камер.

### ПРОСТЫМИ СЛОВАМИ

Это самый простой путь. Загружаете снимки, приложение считает. Очень хорошо подходит для классических сканов объектов — когда вы обходите предмет мебели или скульптуру и делаете 50–200 фотографий. Для полётов дронов над пейзажами или при очень большом числе изображений (более 500) метод Apple, однако, склонен к нестабильности. Для таких сцен попробуйте бэкенд Native (Q4/Q5) или просчитайте камеры в Metashape и загрузите их через Workspace Import (Q6).

### ОПЫТНОМУ ПОЛЬЗОВАТЕЛЮ

Q2 двоичный COLMAP — запускает внешнюю программу COLMAP как подпроцесс и поэтому **недоступен** в версии для App Store (песочница). Работает только в сборках для разработчиков вне App Store. Для качества, которое даёт COLMAP, в версии для App Store есть Workspace Import (Q3 или Q6): просчитайте SfM внешне в COLMAP или Metashape и загрузите результат.

## I Q3 — текстовый формат COLMAP (Metashape / ETH3D)



ГДЕ

Меню «File → Import COLMAP / Metashape Workspace...» (Cmd+⇧+I) ИЛИ перетаскивание папки с `sparse/0/cameras.txt`.



ТЕХНИЧЕСКИ

Читает стандартизированный текстовый экспорт COLMAP — три текстовых файла `cameras.txt`, `images.txt`, `points3D.txt` в подпапке `sparse/0/` — и преобразует во внутреннюю модель результата SfM. То же определение формата, что и у двоичного экспорта COLMAP, только в ASCII вместо двоичного представления. Экспортируется Agisoft Metashape, RealityCapture, PolyCam и бенчмарком ETH3D ровно в этом виде. Парсер разделяет с двоичным парсером определение модели камеры (все распространённые модели: SIMPLE\_PINHOLE, PINHOLE, OPENCV, OPENCV\_FISHEYE, FULL\_OPENCV). Устойчив к строкам-комментариям и пустым строкам. В тестах масштабируется до ~1 400 камер (ETH3D Tunnel) без проблем.

### ПРОСТЫМИ СЛОВАМИ

Если вы уже работали с Metashape, RealityCapture или другой коммерческой программой для фото-3D и экспортировали результат — этот экспорт можно загрузить прямо в RadianceKit, не пересчитывая в приложении. Это экономит часы ожидания. Просто загрузите всю папку через меню File или перетащите её в окно.

## I Q4 — Native SfM (инкрементальный)



ГДЕ

Режим эксперта → Инспектор → Training Configuration → переключатель Camera Alignment, запись «Native (Beta)». Инкрементальный — режим по умолчанию для этого бэкенда — отдельного переключателя маппера в Инспекторе нет. Через CLI режим можно задать явно ключом `--native-sfm` или `--sfm-mapper incremental`.



ТЕХНИЧЕСКИ

Собственная реализация всего конвейера SfM с ускорением на GPU: признаки FAST+BRIEF ИЛИ SuperPoint+LightGlue через CoreML (с `--coreml-features`), затем сопоставление Hamming-KNN, RANSAC фундаментальной матрицы, построение треков, выбор начальной пары, двухвидовый bootstrap (F→E плюс DLT), жадный инкрементальный маппер с регистрацией PnP и многокурсной триангуляцией, и наконец bundle adjustment через Schur-редуцированный Levenberg-Marquardt с функцией потерь Хьюбера и аналитическими якобианами через Cholesky-solve. Полностью совместим с App Store: никакого внешнего бинарного файла, Sandbox=true. С детектором коллапса R2, поставляемым в фазе 3.10: если приложение регистрирует менее 60 % входных кадров или количество точек на камеру падает ниже 13, оно автоматически переключается на глобальный маппер (Q5). Эмпирически чист на сценах орбита/поворотный стол; при более общих движениях (полёт дрона, помещения со сложной геометрией) показатель успешности ниже — однако детектор такие случаи перехватывает. Надёжно масштабируется до ~200 камер, выше — со значительно бóльшим временем выполнения.

### ПРОСТЫМИ СЛОВАМИ

Сильные стороны Apple (совместимость с App Store, быстрая работа на орбитах) плюс дополнительные диагностические значения. Особенно хорошо работает, когда вы обходите объект, как для Object Capture. При более сложных съёмках (полёт дрона или жилая комната) RadianceKit автоматически распознаёт, что так не получится, и переключается на глобальный метод. Помечен как «Beta», потому что ещё проверяется — стандартной рекомендацией остаётся Apple Photogrammetry для простых сканов объектов и Workspace Import (Q3 или Q6) для требовательных уличных наборов.

## I Q5 — Native SfM (глобальный)



Вызывается автоматически, когда инкрементальный маппер (Q4) срабатывает детектором коллапса (зарегистрировано менее 60 % входных кадров или количество точек на камеру ниже 13). Принудительно вручную — только через CLI `--sfm-mapper global`. В Инспекторе глобальный метод недоступен через отдельный переключатель — приложение само решает, когда переключаться.



Глобальный вариант собственного конвейера. Сначала извлечение признаков + сопоставление как в Q4, затем оценка относительных поз для всех проверенных пар, после — усреднение поворотов (синхронизирует все повороты камер в мировой системе координат) и усреднение перемещений (на основе LSQR в *matrix-free sparse-формулировке*, чтобы избежать переполнения `integer` при большом числе камер). В принципе масштабируется до ~5 000 камер, на практике с потерей качества при числе свыше нескольких сотен камер — замер порога приёма фазы 3.8 §5 на K-1351 дал `finalLoss 0.07` вместо целевых 0.0115. Трактуются как «fallback-уровень»: вступает в действие, когда инкрементальный маппер вырождается, но сам по себе на качество повторно не проверяется.

### ПРОСТЫМИ СЛОВАМИ

Запасной путь нативного движка. Вызывается автоматически, когда более быстрый инкрементальный путь не справляется. Выдаёт рабочий результат, но на очень больших или сложных сценах обычно не так точен, как то, что получается из Metashape или внешней установки COLMAP. Если Native станет вашим основным рабочим процессом, в таких случаях имеет смысл сделать обход через Workspace Import (Q3 или Q6).

## Q6 — импорт рабочей области Metashape / текстового COLMAP (фаза Q7)

### ГДЕ

Меню File → «Import COLMAP / Metashape Workspace...» (Cmd+⇧+I). Перетаскивание папки с `sparse/0/cameras.{bin,txt}` и `images/`.

### ТЕХНИЧЕСКИ

Автоматически распознаёт, соответствует ли папка, выбранная перетаскиванием или через `open-panel`, одной из трёх схем размещения рабочей области COLMAP (`sparse/0/`, `sparse/` или корень), и является ли реконструкция двоичной (`cameras.bin`) или текстовой (`cameras.txt`). Двоичный путь использует двоичный парсер COLMAP, текстовый — загрузчик ETH3D — оба порождают одну и ту же модель результата SfM, и остальной конвейер (импорт изображений, запуск тренировки MCMC) безразличен к источнику. Изображения открываются через систему `bookmark` песочницы приложения с `security-scored`-доступом, так что импорт работает и в версии для App Store. Специально рассчитан на случай «экспорт из Metashape без перерасчёта реконструкции». Упомянутый в пункте меню File детектор предупреждает в журнале приложения, если выбранная папка не является распознаваемой рабочей областью.

### ПРОСТЫМИ СЛОВАМИ

Это специальная функция для пользователей Metashape. Если у вас есть лицензия на Metashape или RealityCapture и реконструкция камер уже сделана там, можно просто перетащить папку экспорта сюда и сразу запустить обучение. Экономит несколько часов вычислений на больших сценах, потому что RadianceKit сам SfM тогда не делает.

## Какой бэкенд когда?

Сценарий	Рекомендуемый бэкенд
Скан объекта, 50–200 фотографий	Q1 Apple Photogrammetry
Крупная уличная съёмка / дрон / >500 изображений	Q6 Workspace Import (рассчитайте в Metashape или COLMAP, затем загрузите)
Имеется экспорт Metashape/RealityCapture	Q6 Import (SfM не нужен)
Набор ETH3D / академический текстовый COLMAP	Q3 текстовый импорт COLMAP
Строгая совместимость с App Store + орбитальная сцена	Q4 Native инкрементальный
Q4 не справляется	Q5 Native глобальный (автоматически)
Данные бенчмарка ETH3D	Q3 (autotest precomputed)

## Быстрое сравнение

Бэкенд	App Store	Песочница	Внешний бинарный файл	Лучшее применение	Макс. ~камер
Q1 Apple PG	✓	✓	—	Объект-орбита	~300
Q2 COLMAP двоичный	✗ (только сборка для разработчиков)	—	colmap/glomap	Большие уличные	~5 000
Q3 текстовый импорт COLMAP	✓	✓	—	Бенч-наборы	~1 500
Q4 Native инкрементальный	✓	✓	—	Объект-орбита	~200
Q5 Native глобальный	✓	✓	—	Откат с Q4	~1 351
Q6 Workspace Import	✓	✓	—	Повторное использование Metashape	по источнику

## ГЛАВА

## Глава 10 — Режим новичка

---

Режим новичка (нем. *Einsteiger-Modus*, `Cmd+1`) — это направляемый рабочий процесс для всех, кто впервые восстанавливает 3D-сцену *Gaussian Splatting*. Вместо боковой панели, полной полем Инспектора, приложение проводит вас через четыре шага: сначала импорт изображений или видео и выбор *Quality Preset*, затем выполняется обработка (*SfM + Training*), после чего готовую сцену можно осмотреть в 3D-предпросмотре, и наконец — экспорт в нужный формат. Тонкая полоса прогресса в верхней части окна всегда показывает, на каком шаге вы сейчас находитесь.

По сравнению с *Expert Mode* (`Cmd+2`), который показывает все панели управления одновременно, Режим новичка скрывает неиспользуемые опции, выдаёт валидационные предупреждения при слишком малом числе или плохом качестве изображений и на каждом шаге предлагает только те кнопки, которые имеют смысл в текущем состоянии. Переключаться между *Simple* и *Expert Mode* можно в любой момент (`Cmd+1 / Cmd+2`), всё состояние — импортированные изображения, выбранная предустановка, текущая тренировка, готовое облако точек — сохраняется и сразу доступно в другом режиме.

## Z1 — Импорт (выбор изображений и предустановки)

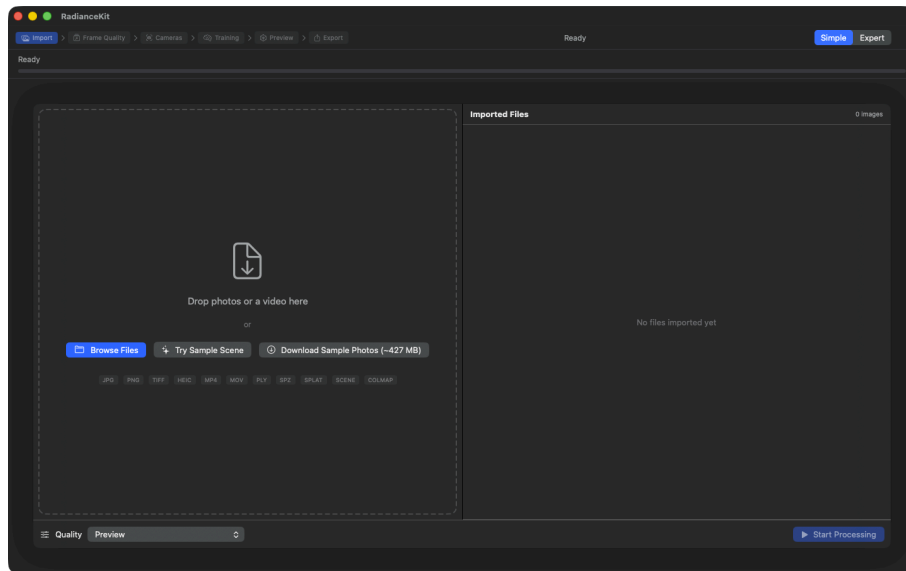


Рисунок 35. Simple Mode шаг 1 — пустая зона дропа до импорта изображений, хлебные крошки сверху (Import → Frame Quality → Cameras → Training → Preview → Export), пилюли форматов JPG/PNG/TIFF/HEIC/MP4/MOV/PLY/SPZ/SPLAT/SCENE/COLMAP

**ЧТО ВИДНО НА ИЗОБРАЖЕНИИ** Хлебные крошки (Import активен) показывают четырёхшаговый рабочий процесс. Слева — зона дропа с тремя СТА: «Browse Files» (NSOpenPanel), «Try Sample Scene» (встроенное демо), «Download Sample Photos (~427 MB)» (подмножество flowers из Mip-NeRF360). Пилюли форматов под ними перечисляют все принимаемые типы файлов. Справа — «Imported Files» со счётчиком «0 images» и пустым состоянием «No files imported yet». Ниже — Quality picker (по умолчанию: Preview) и «Start Processing» (деактивирован, пока нет изображений).

Первый шаг состоит в том, чтобы передать приложению изображения. Через drag-and-drop в большое пунктирное поле посередине, через кнопку «Browse Files» или щелчком по встроенной демо-сцене. Справа появляется список всех импортированных изображений с разрешением и размером файла; ниже во всплывающей панели инструментов вы выбираете Quality Preset и запускаете конвейер кнопкой «Start Processing». Валидационные предупреждения (красное при < 3 или < 10 изображениях, оранжевое при 10–19) показывают, ожидает ли приложение разумную реконструкцию или нет.

**C-01 ProgressIndicator (индикатор шага)****ГДЕ**

Сверху над рабочим процессом, всегда виден.

**ТЕХНИЧЕСКИ**

Показывает горизонтальную полосу прогресса по всему конвейеру (Frame Quality → SfM → Training) с распределением по этапам: Frame Quality занимает 0–5 % (фаза 3.11, очень короткая), SfM — 0–30 % полосы, Training — 30–100 %. Рядом текст статуса и проценты с именем фазы («SfM 41 %», «Training 12 500/20 000»), чтобы пользователи не читали кажущуюся регрессию «41 % SfM → 25 % Training» как ошибку — полоса показывает общий прогресс конвейера, а не подшаг. Расчёт ETA стартует, как только измерена достаточная скорость обучения (обычно после первых 100 итераций). То же отображение используется и в Expert Mode над Инспектором.

**ПРОСТЫМИ СЛОВАМИ**

Тонкая полоса в самом вер-ху — ваша карта по рабоче-му процессу. Она говорит вам не только что приложение сей-час делает (выравнивает каме-ры, идёт обучение, ...), но и насколько далеко в целом про-двинулось. Деление сделано намеренно так, что вычисление камер занимает первую треть полосы, а собственно обучение — задние две трети, иначе бы казалось, что прогресс внезап-но скакнул обратно к нулю по-сле SfM. Так вы можете рас-слабиться, взгляда на полосу достаточно, чтобы увидеть гру-бый этап. Текст рядом говорит, в каком вы сейчас этапе SfM (напр. «SfM 41 %») или в обуче-нии (напр. «Training 12 500/20 000»), чтобы цифры не каза-лись запутанными. Если ETA не показан, обучение просто ещё слишком молодое — приложе-ние оценивает только когда из-мерило достаточную скорость.

**C-03 DropZoneView (зона drag-and-drop)**

ГДЕ

Левая сторона шага импорта, большой пунктирный прямоугольник с символом. В Simple Mode отображается с надписью «Drop photos or a video here».

**ТЕХНИЧЕСКИ**

Зона дропа, которая кратко подпрыгивает символом и тонирует фон, как только над полем появляются drag-объекты. Принимает JPG, PNG, TIFF, HEIC, MP4, MOV, PLY, SPZ, .splat, бандлы .radiancecene и директории. Маршрутизация дропа по типу: изображения собираются и передаются в отсортированном порядке, видео триггерят путь frame-sampling, splat-файлы открывают предпросмотр напрямую, scene-бандлы прочитываются. Директории перечисляются, и все содержащиеся изображения импортируются. Security-scoped bookmarks для sandbox-совместимого доступа корректно приобретаются и отпускаются. Неподдерживаемые расширения отображаются как warning-баннер на 5 секунд.

**ПРОСТЫМИ СЛОВАМИ**

Большое пунктирное поле — главный элемент управления первого шага. Просто перетащите в него фотографии или видео, или целую папку — приложение возьмёт всё, что знает, и проигнорирует остальное. Когда поле становится синим и символ кратко подпрыгивает, приложение распознало drag. Отпустите, и импорт сразу начнётся: изображения переходят в список справа, видео автоматически триггерят шаг frame-sampling, а уже обученные файлы .ply / .spz / .splat открывают предпросмотр напрямую. Если формат вообще не подходит (напр., PDF или BMP), сверху на короткое время появляется уведомление — приложение не глотает молча неизвестный материал.

**C-05 Кнопка Browse Files**

ГДЕ

Внутри зоны дропа, видная кнопка.

**ТЕХНИЧЕСКИ**

Кнопка, открывающая файловый диалог macOS с множественным выбором и типами файлов JPG, PNG, TIFF, MP4, MOV, папки, а также собственным форматом сцены приложения. Результирующие URL — security-scoped и проходят через те же пути импорта, что и drag-and-drop. Когда пользователь выбирает папку, она рекурсивно перечисляется на предмет изображений.

**ПРОСТЫМИ СЛОВАМИ**

Если drag-and-drop вам неудобен, просто щёлкните по этой кнопке и в файловом диалоге macOS перейдите к вашим фотографиям. Можно выбрать несколько файлов сразу (Cmd-клик по отдельным изображениям) или выбрать целую папку — приложение тогда рекурсивно ищет в папке все поддерживаемые типы изображений. Это особенно удобно, когда ваши съёмки разложены по вложенным подпапкам (напр., «shoot-day1/», «shoot-day2/») — одного щелчка по главной папке достаточно. Функционально кнопка делает ровно то же, что и drag-and-drop; выбирайте путь, который удобнее вам.

### C-06 Кнопка Try Sample Scene



ГДЕ

Внутри зоны дропа, видна только когда бандл приложения содержит демо-сцену и пока не импортированы изображения/splats.



ТЕХНИЧЕСКИ

Появляется только если (а) в бандле приложения есть `sample-scene.splat`, `.spz` или `.ply` И (б) не импортированы изображения/видео и нет облака точек. По щелчку загружается готовое облако точек (предпочитая наименьший формат — `.splat` ~3 MB, `.spz` ~1.4 MB, fallback `.ply`) и через 400 мс устанавливаются жёстко зашитые значения камеры из исходных метаданных flower-сцены для эстетически приятной начальной перспективы.

#### ПРОСТЫМИ СЛОВАМИ

Если вы впервые запускаете приложение и просто хотите увидеть, что выходит в конце — щёлкните здесь. Откроется готовая обученная flower-сцена, которую можно сразу крутить и экспортировать, без необходимости что-то вычислять. Камера предустановлена в эстетически приятную начальную перспективу, так что вы сразу видите красивое. Идеально для безрискового испытания 3D-управления и шага экспорта, прежде чем браться за собственные съёмки. Как только вы импортируете собственные изображения, кнопка автоматически исчезает — она показывается только пока проект полностью пуст.

### C-07 Кнопка Download Sample Photos



ГДЕ

Внутри зоны дропа, рядом с «Try Sample Scene»; те же условия видимости.



ТЕХНИЧЕСКИ

Запускает загрузку (репозиторий [github.com/bkindler/radiancekit-sample-photos](https://github.com/bkindler/radiancekit-sample-photos)), которая загружает прим. 427 MB из 960 кадров полного разрешения и скармливает их приложению. Во время загрузки кнопка деактивирована. Прогресс появляется в верхней полосе прогресса как «Downloading X %» на собственном этапе, потому что этот этап держит собственную шкалу 0–100 % и не пересекается с последующим этапом SfM.

#### ПРОСТЫМИ СЛОВАМИ

Как и демо-сцена, только с исходными фотографиями вместо готового результата. Так можно один раз самому прогнать весь конвейер и увидеть, сколько на самом деле длятся SfM и обучение на вашем Mac. Загрузка большая (около половины DVD = 427 MB), но происходит только один раз — после этого фотографии локальны, и можно запускать конвейер сколько угодно с разными предустановками. Пока идёт загрузка, верхняя полоса прогресса показывает текущий статус загрузки в процентах, чтобы вы могли оценить, когда она запустится. Совет: используйте быстрый Wi-Fi или проводную сеть — 427 MB иначе займут какое-то время.

**C-09** Quality Presets Picker

Всплывающая нижняя панель инструментов import-оверлея, слева от Start-кнопки.



Элемент управления с меткой «Quality» группирует доступные предустановки по категориям (Classic / MCMC / Custom). Встроенные предустановки группируются по категориям; заголовки секций жёстко защищены. Пользовательские предустановки видны только если они есть. Locked-состояние: предустановки не из бесплатного списка (Quick + Preview) получают суффикс «» при имени, если пользователь не купил; при выборе picker отскакивает обратно к Preview и автоматически открывает purchase-sheet. При выборе предустановка применяется, что заменяет всю конфигурацию обучения.

 **ПРОСТЫМИ СЛОВАМИ**

Здесь вы выбираете, насколько точно и долго приложение должно считать. «Quick» и «Preview» можно использовать без покупки и они дают первый результат за несколько минут — идеально для проверки, насколько ваши изображения вообще разумны. «Balanced» и «Quality» требуют полной версии и дают существенно более чистые модели, но длятся часами вместо минут. MCMC — другая стратегия, обходящаяся меньшим числом Gaussian splats — хорошо, если позднее хочется компактно экспортировать модель или положить в сеть. Премиум-предустановки узнаются по маленькому значку замка при имени; если нажать на одну из них без лицензии, picker отскакивает обратно к Preview, и purchase-sheet открывается автоматически. Правило большого пальца: всегда начинайте с Preview, посмотрите результат и только потом решайте, стоит ли более длинный прогон.

**C-10** Кнопка Start Processing

ГДЕ

Всплывающая нижняя панель инструментов import-оверлея, справа от preset-picker.

**ТЕХНИЧЕСКИ**

Кнопка, остающаяся серой, пока не импортированы ни изображения, ни видео. По щелчку запускает конвейер и переключает stage-машину в порядке Frame Quality → SfM → Training. У самой кнопки нет дальнейшего статуса; вместо этого идущая обработка появляется как отдельный processing-экран.

**ПРОСТЫМИ СЛОВАМИ**

Кнопка «Go». Пока она серая, не хватает входных изображений или видео. Как только вы перетащили фотографии, она становится активной, и щелчком запускаются SfM и обучение последовательно. Далее приложение берёт на себя весь рабочий процесс, и вы автоматически попадаете на processing-экран (Z2). Больше ничего не нужно нажимать — только после окончания обучения приложение переключается обратно к предпросмотру (Z3). Если вы передумаете, в любой момент после этого можно также отменить через Cancel.

**C-11** Слайдер Video Sampling

ГДЕ

Правый список изображений, виден только когда импортировано видео (вместо изображений).

**ТЕХНИЧЕСКИ**

Слайдер 0.5 fps – 30 fps с шагом 0.5. При изменении обновляется плотность кадров и дополнительно из плотности и длины видео вычисляется целевое число кадров (минимум 10). Слайдер расположен вне списка изображений, поскольку элементы списка блокировали бы события мыши от слайдеров. Под слайдером показываются вычисленные target-frames («247 frames») и длина видео («1m23s video»). Tooltip предупреждает: «Удвоение плотности удваивает количество кадров и увеличивает время SfM на ~100 %».

**ПРОСТЫМИ СЛОВАМИ**

Если вместо фотографий вы импортировали видео, этот слайдер решает, сколько отдельных изображений приложение должно извлечь из видео. Больше изображений = лучше качество, но линейно больше времени вычислений. Для 30-секундного orbit-видео 5 fps (150 изображений) — хорошее начало; для 1-минутных записей 3 fps часто более чем достаточно. Под слайдером приложение показывает вживую, сколько кадров получается при текущей настройке — так сразу видно, попадаете ли вы в разумный диапазон около 100–300 изображений. Если результат становится плохим, потяните слайдер вправо и попробуйте снова; но удвоение частоты кадров также примерно удваивает длительность SfM.


**C-12** Кнопка Clear All

ГДЕ

Правый список изображений, внизу справа; видна только если импортированы изображения.

 ТЕХНИЧЕСКИ

Красная кнопка. Щелчок открывает диалог подтверждения с заголовком «Clear all imported files?» и сообщением «N images will be removed.». Подтверждение очищает все импортированные изображения/видео, staging- директории, облако точек, статус обучения, результат SfM и все кеши; этап прыгает обратно в Import. По Cancel всё сохраняется. Диалог настроен как недеструктивный default-путь (деструктивная кнопка помечена красным).

 ПРОСТЫМИ СЛОВАМИ

Если хотите начать совсем сначала, щёлкните здесь. Запрос подтверждения появляется, потому что удаление выбрасывает все текущие импорты, включая уже рассчитанные камеры и результаты обучения — отменить нельзя. Полезно, когда хотите полностью заменить выбранный материал изображений или избавиться от старого проекта перед началом нового. Примечание: удаление одного изображения делается через список справа (см. следующий пункт), не через эту кнопку. Ваши файлы на диске при этом не удаляются — приложение только забывает свои ссылки на них.

**C-13** File List ForEach (удаление одного изображения)

ГДЕ

Правый список изображений, каждая запись.

 ТЕХНИЧЕСКИ

Список по импортированным изображениям с swipe-to-delete. На каждое изображение — строка со значком, именем файла, разрешением («1920 x 1080») и размером файла (форматировано KB/MB). Разрешение приходит из metadata-кеша, заполняемого асинхронно из заголовков изображений, чтобы UI не блокировался. Действие удаления предлагает типичный для macOS swipe-delete (свайп влево трекпадом по строке) и delete с клавиатуры для выбранной строки. Примечание: расширенный путь удаления изображений с явной кнопкой минус, backspace и Cmd-Z для отмены был добавлен *только в Expert Mode* в Project Navigator — в Simple Mode остаётся swipe-delete.

 ПРОСТЫМИ СЛОВАМИ

Список справа показывает каждое импортированное изображение с разрешением и размером файла — удобно для быстрой оценки, не смешали ли вы высокого разрешения с низким. Для удаления одного изображения свайпните его двумя пальцами влево по трекпаду — как в iOS Mail — или выберите и нажмите Delete. Сам файл приложение не удаляет; оно только убирает его из текущего проекта. Если нужна полноценная кнопка минус или Cmd-Z для отмены, переключитесь в Expert Mode (Cmd+2), там это есть в Project Navigator. В Simple Mode сознательно остаётся простой паттерн swipe.

**C-15** Валидационные предупреждения (3 уровня)

ГДЕ

Под списком изображений, над кнопкой Clear All.

**ТЕХНИЧЕСКИ**

Три последовательных порога на основе числа импортированных изображений (активно только когда есть изображения и нет видео): - < 3 изображений: красный баннер (красный восьмиугольник), текст «At least 3 images are required. Camera alignment cannot be computed from fewer images.» - 3–9 изображений: красный баннер, текст «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 изображений: оранжевый баннер (warning-треугольник), текст «Workable, but quality usually improves with 20+ images and good coverage around the scene.»

Начиная с 20 изображений баннер исчезает. Пороги жёстко защиты и основаны на эмпирике 560+ экспериментов по обучению.

**ПРОСТЫМИ СЛОВАМИ**


Приложение смотрит, сколько изображений вы импортировали, и даёт цветную оценку. Красный означает: это скорее всего не получится — либо SfM не сможет вычислить камеры, либо обучение переобучится на слишком малом материале. Оранжевый означает: возможно сработает, но не ждите того качества, потому что алгоритм находит мало перекрытий между изображениями. Никакой баннер означает: хорошие условия, у вас достаточно материала. Если действительно хотите чистые модели, цельтесь минимум в 30–50 равномерно распределённых снимков вокруг вашего объекта — и значительно больше для уличных сцен или больших комнат. Можно стартовать и при предупреждении, но не удивляйтесь, если SfM прервётся без комментариев или модель окажется дырявой.

**C-16** Распознавание COLMAP-workspace **ГДЕ**

При дропе папки — не видимая кнопка, а логика распознавания.

 **ТЕХНИЧЕСКИ**

При дропе директории проверяется, содержит ли она одну из трёх канонических раскладок workspace: `sparse/0/cameras.bin`, `sparse/cameras.bin` или напрямую `cameras.bin` в корне. Если да, стандартное перечисление изображений прерывается, и вместо этого открывается модальный alert, спрашивающий пользователя, использовать ли существующую реконструкцию или прогнать изображения заново через Apple Photogrammetry. Тот же путь и для workspace в текстовом формате (`cameras.txt`) и для ETH3D-экспортов. Подробности — глава 9, бэкэнд Q6. Работает в Simple Mode так же, как и в Expert Mode.

 **ПРОСТЫМИ СЛОВАМИ**

Если вы уже работали с Metashape, RealityCapture или COLMAP и сделали там вычисление камер, можно просто перетащить папку экспорта сюда. RadianceKit автоматически распознаёт по содержанию, что это COLMAP-workspace (он проверяет `sparse/0/`, `cameras.bin` и т.п.), и спрашивает, взять ли готовое вычисление или вычислить самому. Принять готовое сэкономит часы ожидания на больших сценах, потому что SfM полностью пропускается — обучение запускается сразу. Также распознаются текстовые workspace (`cameras.txt`) и ETH3D-экспорты. Эта функция в Simple Mode доступна так же, как и в Expert Mode; подробнее — в главе 9 под бэкэндом Q6.

**Когда переходить к следующему этапу?**

Щёлкнуть Start Processing можно, как только (а) импортировано хотя бы одно изображение или видео и (b) баннер валидации оранжевый или его нет. При красном баннере приложение даст стартовать, но с высокой вероятностью обработка тут же отменится сама. Рекомендация: минимум 20 изображений, резких, с чётким перекрытием между последовательными снимками, все примерно с одного расстояния до объекта. Перед стартом выберите предустановку под ваш бюджет времени — с 30 изображениями и предустановкой Quick вы готовы за несколько минут, с Quality скорее уйдут 1–2 часа.

## Z2 — Обработка (SfM + Training)

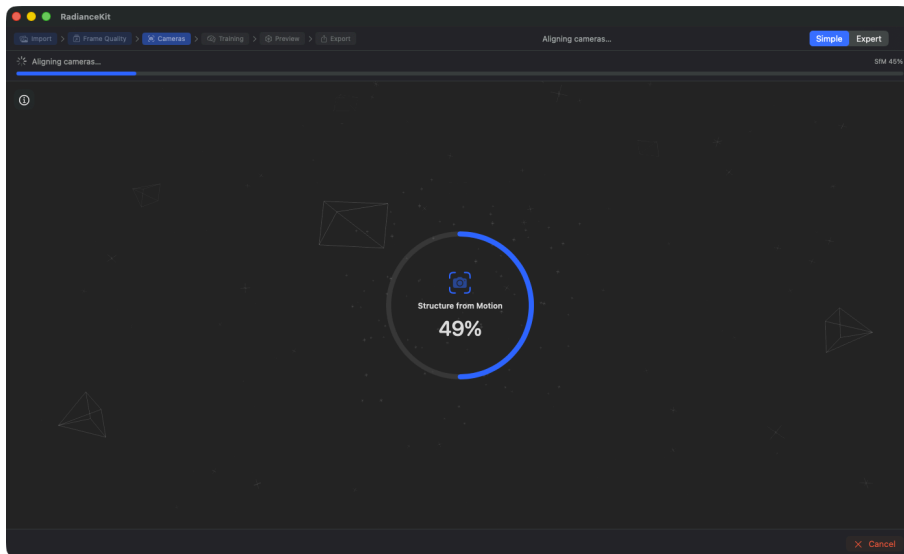


Рисунок 36. Z2 SfM-фаза — значок этапа «Structure from Motion» с 41 % в большом круге, верхняя статус-полоса на «SfM 25 %», кнопка Cancel внизу справа

**SfM-фаза (камеры выравниваются):** Большой круг прогресса показывает прогресс подэтапа (здесь 41 % текущей сессии Apple Photogrammetry). Текст статуса «Aligning cameras...» сверху слева. Хлебные крошки помечают «Cameras» как активный этап. Верхняя статус-полоса показывает общий прогресс конвейера (25 %) — SfM занимает первую половину полосы. Парящие wireframe-камеры на фоне намекают, что идёт оценка поз.

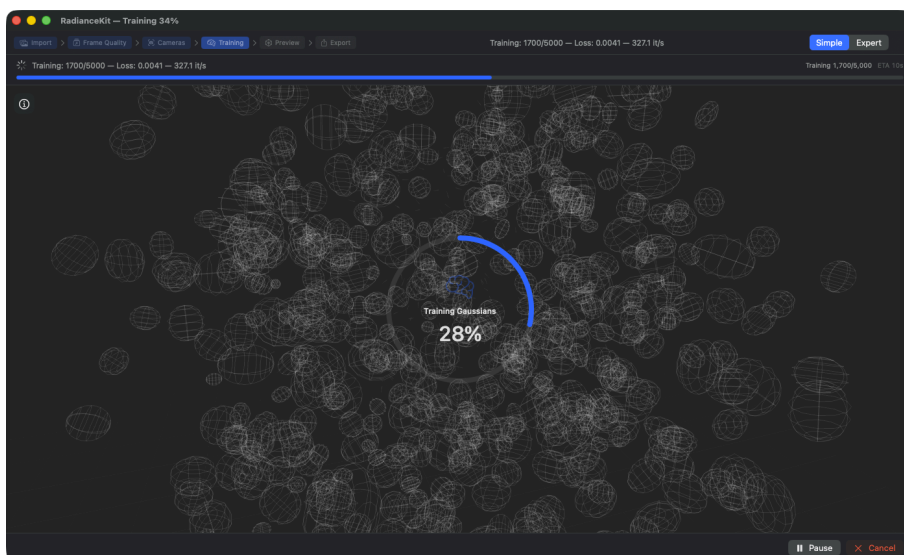


Рисунок 37. Z2 фаза обучения — значок этапа «Training Gaussians» на 6 %, живые метрики сверху (Training: 400/5000 — Loss: 0.1642 — 138.7 it/s), ETA 33 с, Pause/Cancel внизу

**Фаза обучения (оптимизируются Gaussians):** Значок подэтапа переключается на «Training Gaussians», проценты считают итерации выбранной предустановки (здесь 400 / 5 000 для Preview-предустановки = 8 % этапа). Строка живых метрик

показывает значение loss (0.1642), итерации в секунду (138.7 it/s) и ETA (33 с). Общий прогресс конвейера в этой фазе поднимается с 50 % до 100 %. Кнопка Pause (вместо только cancel в SfM-фазе) позволяет позднее возобновить; Cancel отбрасывает результат обучения и возвращает к Z1.

Как только конвейер запущен, приложение скрывает import-overlay и показывает полноэкранный processing-экран. В центре движется большой круг прогресса (220 × 220 пикселей) со значком этапа, текстом статуса и числом процентов; в фоне тонкая splat-анимация символически визуализирует идущие вычисления. Вверху слева можно показать info-панель, отображающую живые метрики из обучения и SfM. Внизу есть Pause/Resume, Cancel, а в случае ошибки — кнопка Retry.

### C-18 SplatTrainingView (фоновая анимация)



ГДЕ

Полноэкранный фон за кругом прогресса, скрывается при отмене или ошибке.



ТЕХНИЧЕСКИ

Декоративная анимация, которая в зависимости от прогресса конвейера (0...1) рендерит растущее число маленьких анимированных splat-частиц. Источник — вычисляемое значение прогресса, которое мапит SfM-фазы в 0–0.2, а обучение — в 0.2–1.0 (Frame Quality в 0–0.05). Так splats заметно «нарастают», пока идёт обучение. Чисто декоративно — отображение не показывает реальные промежуточные результаты текущего обучения (это был бы live preview в Expert Mode). При отмене или сбое скрывается, и виден только круг статуса.

#### ПРОСТЫМИ СЛОВАМИ

На фоне идёт маленькая анимация танцующих точек, чтобы экран не выглядел таким пустым во время вычислений. Это не ваша реальная 3D-модель — её вы увидите только после обучения, на шаге Z3. Анимация, однако, имеет ту же тональность, так что по приблизительной плотности можно прочесть, насколько далеко продвинулось обучение. В начале видны лишь несколько точек, к концу фон заполняется заметно плотнее — симпатичный визуальный индикатор в дополнение к процентному отображению в круге. Если анимация вам мешает (напр., потому что вы хотите параллельно работать в фоне), можно переключиться в Expert Mode, где её нет.

**C-19** Большой круг прогресса

ГДЕ

Центр processing-экрана, 220 × 220 пикселей.

**ТЕХНИЧЕСКИ**

Два кольца, отрисованных поверх друг друга: снаружи — приглушённое track-кольцо, внутри — заполненное кольцо прогресса с акцентной или красной обводкой (красный при ошибке). Внутри круга — значок этапа (мозг для обучения, камера для SfM, плёнка для извлечения кадров видео, искры для Frame Quality), заголовок этапа и живое анимированное процентное число шрифтом 32 пункта rounded. Значок мягко пульсирует, пока обработка активна. Отображение интерполирует на таймере 30 Гц плавно к текущему фактическому прогрессу — с постоянной срееp (0.0003/кадр) плюс пропорциональной долей (4 % от gap) и мягким потолком, устанавливающимся на 80 % следующей ожидаемой вехи (для SfM — из жёстко зашитой таблицы вех). Так прогресс ощущается плавным, даже если фактические обновления SfM приходят раз в несколько секунд.

**ПРОСТЫМИ СЛОВАМИ**

Большой круг посередине — ваш главный экран, пока приложение считает. Он наполняется плавно, даже когда обновления реального вычисления приходят раз в несколько секунд — это даёт ощущение, что что-то происходит, вместо того чтобы минутами смотреть на замёрзший процент. Символ в середине меняется в зависимости от того, идёт ли сейчас извлечение кадров (значок плёнки), выравнивание камер (значок камеры) или обучение Gaussians (значок мозга). Число процентов относится к текущему подшагу — общий конвейер видно в тонкой полосе в самом верху. При ошибке кольцо становится красным вместо синего, а значок перестаёт пульсировать, так что вы сразу замечаете, что что-то пошло не так.

**C-22** Info-кнопка (показать метрики)

ГДЕ

Вверху слева на processing-экране, 32 × 32 пикселя.

**ТЕХНИЧЕСКИ**

Простая кнопка с material-фоном. Переключает info-панель вкл/выкл. Значок переключается между info-circle-outline и info-circle-filled при активации. Плавная fade-in-анимация. Tooltip «Show detailed processing metrics».

**ПРОСТЫМИ СЛОВАМИ**


По умолчанию экран сознательно опрятен — только большой круг прогресса, больше сначала ничего не видно. Если как технически заинтересованному пользователю вам хочется точнее знать, что происходит (какая итерация, какой loss, сколько Gaussians), щёлкните по значку *i* вверху слева. Внизу выезжает маленькая панель и показывает все живые значения. Повторный щелчок снова её скрывает. Настройка не персистентна — при каждом новом прогоне обучения панель сначала снова скрыта, что сознательно выбрано, чтобы не пугать новичков.

**C-23** Info-панель (живые метрики) **ГДЕ**

Внизу слева на processing-экране, видна только когда `showProcessingInfo == true`.

 **ТЕХНИЧЕСКИ**

Двухколоночная панель с ultra-thin material-фоном. Левая колонка: специфичные для этапа info-строки — для SfM текст статуса и проценты; для обучения — итерация, комбинированный loss, L1 loss, D-SSIM loss, число Gaussians (оранжевым), скорость (it/s), прошедшее время, рассчитанный ETA, SH-степень и learning rate. Правая колонка: текст статуса, строка с информацией о времени, инлайновый loss-график (см. C-28) и discoverability nudge (см. C-32). Все значения считываются из training status, который обновляется на каждом тике обучения.

 **ПРОСТЫМИ СЛОВАМИ**

Info-панель показывает все живые значения, которые в Expert Mode постоянно были бы на боковой панели Инспектора: текущая итерация, значение loss (меньше = лучше), число Gaussians, скорость, оценочное оставшееся время, SH-степень и learning rate. На правой стороне дополнительно идёт крошечная кривая loss, которая на одном взгляде говорит, идёт ли обучение в правильном направлении. Если обучение кажется вялым, взгляд сюда помогает — loss, который больше не падает, или ETA, которое больше не уменьшается, указывают на проблемы. Если loss взрывается (внезапно становится огромным) или показывает NaN, обучение стало неустойчивым, и имеет смысл Cancel + Retry или переключиться на другую предустановку.

## C-25 Кнопка Pause/Resume



ГДЕ

Нижняя navigation-панель, видна только во время этапа обучения (НЕ во время SfM) и пока обработка идёт.



ТЕХНИЧЕСКИ

Bordered-кнопка. Вызывает Pause или Resume в зависимости от статуса. Метка переключается между «Pause» (со значком pause) и «Resume» (значок play). На SfM-шаге кнопка не показывается, потому что у Apple Photogrammetry нет семантики pause. Состояние pause полностью сохраняет итерацию, статус Gaussian и optimizer momentum — Resume продолжает с того места, где ранее остановилось.

### ПРОСТЫМИ СЛОВАМИ

Пока идёт обучение, в любой момент можно его остановить и позднее возобновить. Полезно, когда между делом хочется на Mac заняться чем-то другим, требующим много GPU — напр., видеомонтажом, тестом игры или экспортом рендера из другого приложения. Щелчок Pause, делаете своё, щелчок Resume, обучение продолжается ровно там, где остановилось. Счётчик итераций, число Gaussians и optimizer momentum полностью сохраняются, состояние pause не стоит вам качества. На SfM-фазе Pause недоступна — у Apple Photogrammetry нет функции stop, там в крайнем случае приходится работать с Cancel.

## C-26 Кнопка Cancel



ГДЕ

Нижняя navigation-панель, видна пока идёт обработка (SfM или Training).



ТЕХНИЧЕСКИ

Красная bordered-кнопка. Открывает диалог подтверждения с заголовком «Stop and discard progress?», кнопками «Discard Progress» (destructive) и «Keep Running» (cancel). При подтверждении устанавливается cancel-флаг, training-задача завершается, при необходимости завершается SfM-подпроцесс, и в JSONL-лог пишется summary-строка с cancel-статусом. В отличие от Pause, training-буферы и статус отбрасываются.

### ПРОСТЫМИ СЛОВАМИ


Кнопка отмены. В отличие от Pause это окончательно — если потом захотите перезапустить, обработка пойдёт с начала, все уже обученные итерации потеряны. Полезно, когда вы ошиблись с предустановкой, обучение идёт слишком медленно или приложение явно выдаёт мусорные результаты, и не хочется ждать. Перед собственно отменой приложение ещё раз спросит через диалог подтверждения, чтобы вы случайно не потеряли часы вычислений. Если хочется только кратко прервать, лучше возьмите Pause.

**C-27** Кнопка **Retry**

Нижняя navigation-панель, видна когда конвейер сорвался (SfM-статус начинается с «SfM failed» или обучение в error-состоянии).



Ассент-кнопка. Перезапускает весь конвейер. Перед стартом проверяется, есть ли ещё импортированные изображения/видео. Предыдущие error-логи сохраняются в JSONL-директории; новый прогон пишет новый лог-файл с текущим timestamp.

 **ПРОСТЫМИ СЛОВАМИ**

Если SfM или обучение сорвались с сообщением об ошибке, здесь можно попробовать снова. Иногда это помогает, потому что многие шаги (RANSAC, densification) имеют случайные компоненты, и со второй попытки получается там, где первая не сработала. Затем весь конвейер запускается заново с самого начала — SfM и обучение, в свежий JSONL-лог-файл. Если даже вторая попытка не удалась, обычно проблема во входных изображениях (слишком мало, слишком малое перекрытие, motion blur, плохой свет); тогда вернитесь через Back и поменяйте материал. Совет: параллельно загляните в training-логи (Help → Open Training Logs), там подробнее, где именно застряло.

**C-28** Инлайновый loss-график**ГДЕ**

В info-панели, правая колонка, виден только во время обучения с непустой историей.

**ТЕХНИЧЕСКИ**

Компактная область рисования (высотой 40 пикселей), рисует историю loss как линию в 1 пиксель акцентным цветом. Данные фильтруются на конечные значения (NaN-защита для нестабильных обучений). Min/Max вычисляются по всей истории — график автоматически зумит на диапазон значений. Последнее значение loss сидит сверху справа над графиком. Сама история накапливается в app state на каждом training tick (обычно каждые 100 итераций).

**ПРОСТЫМИ СЛОВАМИ**


Крошечная кривая loss, которая на одном взгляде показывает, «сходится» ли обучение (линия падает вправо) или оно застряло или взрывается (линия плоская или растёт). При здоровом обучении линия в начале круто падает, потом вылаживается — это ожидаемый ход, похожий на кривую половинного распада. График автоматически зумит на текущий диапазон значений, так что даже маленькие улучшения в конце обучения остаются видны. Если линия вдруг прыгает вверх или замораживается, это хороший сигнал, что что-то идёт не так — либо материал проблемный, либо лучше бы подошла другая предустановка. График вы найдёте в info-панели, которую показываете сверху слева значком `i`.

**C-32 Discoverability Nudge (подсказка Expert Mode)** **ГДЕ**

В info-панели, правая колонка внизу, видна только во время обучения И в Simple Mode.

 **ТЕХНИЧЕСКИ**

Маленькая строка со значком глаза и текстом подписи «Switch to Expert Mode (⌘2) for live splat preview», в сдержанном тоне и шрифтом 10 пунктов. Никакого интерактивного элемента, просто подсказка. Не реагирует на щелчок — пользователь должен действительно нажать Cmd+2 или щёлкнуть меню Mode → Expert Mode.

 **ПРОСТЫМИ СЛОВАМИ**

Ненавязчивая подсказка, что в Expert Mode во время обучения текущая промежуточная версия вашей 3D-модели видна вживую в viewport. В Simple Mode это сознательно скрыто, чтобы держать UI спокойным — но многие пользователи не знают даже, что такая функция существует, поэтому здесь мы мягко на неё указываем. Нажмите Cmd+2, и обучение продолжит работу в фоне, а вы сможете наблюдать, как ваша модель складывается на глазах. Это также хороший инструмент, чтобы уже после нескольких тысяч итераций оценить, выйдет ли результат хорошим, или лучше отменить и начать сначала. Cmd+1 в любой момент возвращает обратно к Simple-виду.

**Когда переходить к следующему этапу?**

Приложение автоматически переключается на Z3 (Preview), как только обучение успешно завершено — ничего щёлкать не нужно. Нижняя navigation-панель тогда переключается с Pause/Cancel на кнопку Back (назад к Import) и кнопку Export (вперёд к Export). В случае ошибок (красное сообщение об ошибке, значок этапа — X) вместо этого появляется Retry, и надо решить, начать ли снова или вернуться через Back к Import, чтобы сменить материал изображений.

## Z3 — Предпросмотр (повернуть 3D-модель)

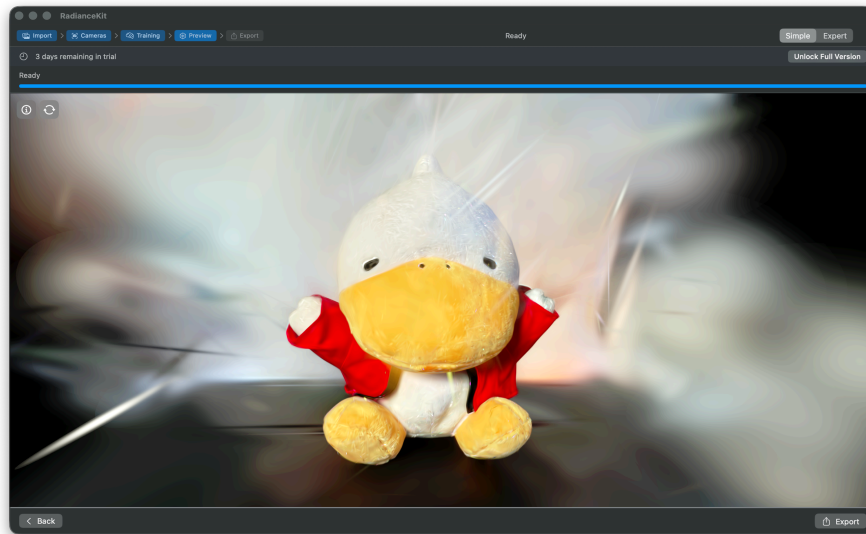


Рисунок 38. Simple Mode шаг предпросмотра с 3D-просмотрщиком

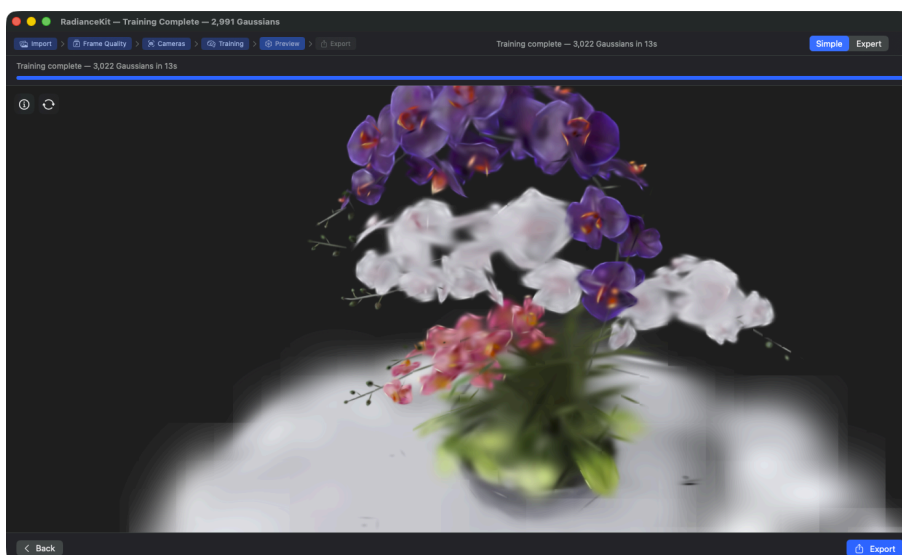


Рисунок 39. Z3 предпросмотр после окончания обучения — реконструированный синтетический букет, заголовок показывает «Training complete — 3,022 Gaussians in 13 s», кнопки Back и Export внизу

**ЧТО ВИДНО НА ИЗОБРАЖЕНИИ** Хлебные крошки помечают «Preview» как активный этап. Полноэкранный 3D viewport рендерит законченно обученную сцену букета (синтетический тестовый набор Blender, подмножество 60 кадров из 960 hemispherical- камер). Заголовочная статус-полоса: «Training complete — 3 022 Gaussians in 13 s» — даёт финальное число Gaussians и время обучения. Drag в viewport вращает камеру (yaw/pitch); колесо скролла зумит вдоль направления взгляда. Кнопка «Back» (внизу слева) возвращает к Z2 для resume или re-run; кнопка «Export» (внизу справа, primary) ведёт дальше к Z4.

После завершения обучения приложение автоматически попадает в предпросмотр. Здесь вы видите вашу законченную модель Gaussian Splatting в полно-

экранном Metal-виде и можете её вращать, зумить и панорнировать мышью и трекпадом. Поверх viewport сидит маленький оверлей с управлением камерой и инфо — авто-вращение, статистика обучения, кнопка сброса. Перед следующим шагом (Export) полезно осмотреть модель с разных ракурсов, чтобы убедиться, что реконструкция чистая.

### **C-36** SplatViewportView (главный 3D-вид)



ГДЕ

Полноэкранный фон шага предпросмотра.

#### ТЕХНИЧЕСКИ

Metal-based 3D viewport, рендерящий законченное облако точек. Рендерер — это СОБСТВЕННЫЙ ForwardPass- растеризатор RadianceKit, тот же самый, который уже рендерит splats во время обучения, поэтому это настоящий WYSIWYG (что обучено, то и показывается, и экспортируется в точности так же). Tile-based рендер-конвейер с order-independent transparency. Если рендерер не может быть инициализирован (напр., потому что Metal недоступен в системе), вместо этого появляется чёрный фон с текстом «Metal not available». Вид игнорирует safe area, так что модель достигает края окна.

#### ПРОСТЫМИ СЛОВАМИ

Главный viewport. Здесь вы видите вашу законченную 3D-модель, реконструированную из ваших фотографий, рендеримую на GPU в реальном времени. Щёлкните и тащите левой кнопкой мыши, чтобы вращать. Колесо скролла или жест трекпада двумя пальцами для зума. Правая кнопка мыши или Cmd+drag для пана. Модель состоит из десятков тысяч полупрозрачных 3D-эллипсоидов («Gaussians»), фотореалистично реконструирующих вашу сцену — у каждого есть позиция, ориентация, форма и цвет, которым научилось обучение. В редком случае, когда ваш Mac не поддерживает Metal, вы вместо этого увидите чёрный фон с сообщением-подсказкой — RadianceKit непременно нуждается в Metal-совместимом GPU.

**C-37** CameraControlsOverlay (оверлей управления)

ГДЕ

Над viewport, плавающий.

**ТЕХНИЧЕСКИ**

Компактный UI-оверлей с кнопками для авто-вращения (turntable), сброса камеры, выбора фона (gray/black/white), сохранения скриншота, переключения info-панели. Привязывается к параметрам камеры (distance, azimuth, elevation, target, FOV) и управляет auto-turntable. Во время обучения (когда пользователь в Expert Mode хочет видеть viewport сопроводительно), оверлей дополнительно показывает компактную строку статуса обучения.

**ПРОСТЫМИ СЛОВАМИ**

Маленькая плавающая панель над моделью. Здесь вы запускаете авто-вращение (модель крутится сама, удобно для скриншотов и коротких демо), сбрасываете камеру через Reset в стартовую позицию (если потерялись), переключаете фон (серый для нейтрального, чёрный для максимального контраста, белый для светлых моделей) и напрямую делаете скриншоты, которые сохраняются под /Pictures. Удобно, когда хочется показать определённую деталь с конкретного ракурса, не экспортируя дополнительно всю модель. Авто-вращение также — хороший тест того, выглядит ли модель одинаково хорошо со всех сторон, или есть «обшарпанная сторона», возникшая из-за недостающих кадров.

**C-38** Кнопка Export (панель навигации)

ГДЕ

Нижняя navigation-панель в Z3.

**ТЕХНИЧЕСКИ**

Accent-кнопка с меткой «Export» и значком share. Щелчок триггерит переключение в Z4. Перед этим parent-вид проверяет, разблокирована ли полная версия — если нет, вместо этапа экспорта показывается lock-вид (см. U-06).

**ПРОСТЫМИ СЛОВАМИ**

Когда вы довольны результатом, щёлкните Export, и попадёте в последний шаг, где выберете формат и сохраните. Без купленной полной версии вы вместо этого попадаете на экранную блокировку с подсказкой разблокировки и кнопкой покупки — приложение не хочет навязывать вам полную версию, но Export — одна из premium-функций. Как только покупка завершена, приложение сразу продолжает работу в разблокированном состоянии, и вы попадаете на привычный этап экспорта. Если передумаете, через кнопку Back вернётесь к предпросмотру и сможете продолжать крутить модель.

## Когда переходить к следующему этапу?

Перед экспортом проверните модель один раз полностью и проверьте: все ли области, которые вы покрыли во входных изображениях, на месте? Есть ли парящие «floaters» (splat-облака Gaussian, свободно висящие в воздухе)? Чисто ли выглядят фон/небо или смазаны? Серьёзные проблемы могут быть исправлены только повторным обучением — либо с большим числом изображений, либо с другой предустановкой, либо в Expert Mode с настройками floater-reduction.

## Z4 — Экспорт (выбор формата и сохранение)

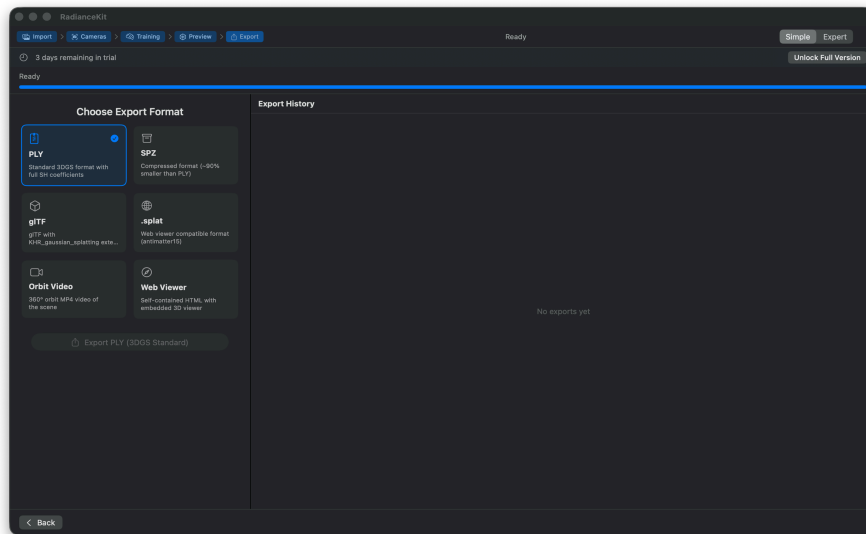


Рисунок 40. Simple Mode шаг экспорта с карточками форматов

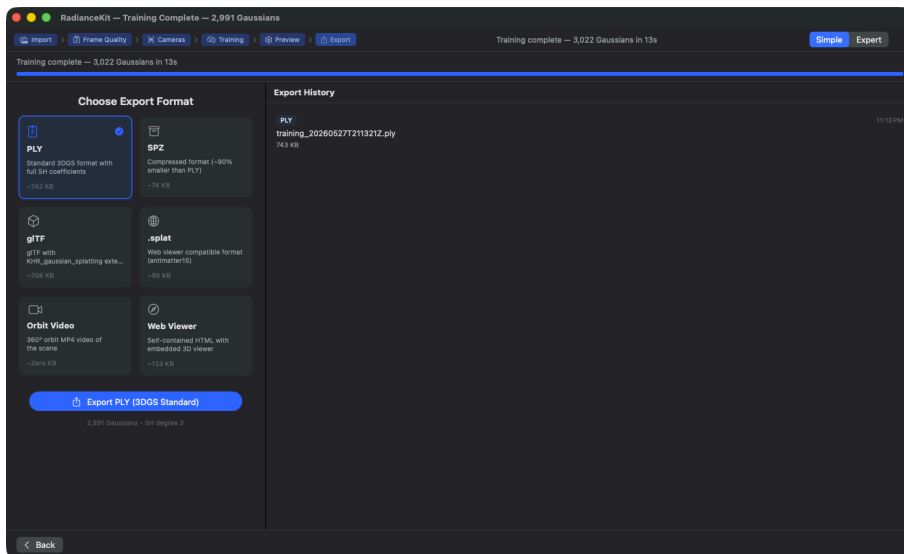


Рисунок 41. Z4 карточки экспорта — 6 форматов (PLY 742 KB выбрано, SPZ 74 KB, glTF 708 KB, .splat 96 KB, Orbit Video, Web Viewer 133 KB), история экспортов в боковой панели справа с уже экспортированным PLY

**ЧТО ВИДНО НА ИЗОБРАЖЕНИИ** Хлебные крошки помечают «Export» как активный этап. Слева — сетка карточек «Choose Export Format» со всеми шестью опциями: PLY (стандартный 3DGS, 742 KB, с полными SH-коэффициентами — здесь предварительно выбран синей галочкой), SPZ (сжатый формат 3DGS, ~90 % меньше PLY, 74 KB), glTF (с расширением KHR\_gaussian\_splatting, 708 KB), .splat (совместим с веб-просмотрщиком через antimatter15, 96 KB), Orbit Video (360° MP4 сцены, live-вычисление размера), Web Viewer (автономный HTML со встроенным 3D-просмотрщиком, 133 KB). Размеры считаются вживую из текущего числа Gaussians и overhead формата. Справа «Export History» перечисляет уже завершённые экспорты со значком формата, именем файла и timestamp — щелчок открывает в Finder. Primary CTA внизу слева: «Export PLY (3DGS Standard)» с подзаголовком про Gaussians «2,991 Gaussians · SH degree 3».

На последнем шаге вы выбираете один из 6 форматов экспорта (PLY, SPZ, glTF, .splat, orbit video, web viewer) через сетку карточек из 2 колонок, щёлкаете Export и выбираете место сохранения в диалоге macOS. Справа идёт история всех предыдущих экспортов — при выборе карточки оценочный размер файла сразу показывается под каждой карточкой, так что вы, напр., предпочтёте SPZ, если хотите идти в веб (маленький), и PLY, если хотите импортировать в другое ПО (SuperSplat, Postshot, Blender через плагин) (большой и полный).

### C-39 Сетка форматов из 2 колонок



ГДЕ

Левая основная сторона шага экспорта.



ТЕХНИЧЕСКИ

Сетка карточек с двумя гибкими колонками и интервалом в 12 пунктов. Итерирует по форматам, предлагаемым в Simple Mode — отфильтрованное подмножество полного списка форматов, содержащее только 6 важнейших форматов: PLY, SPZ, glTF, .splat, orbit video, web viewer. Compressed PLY и SOG предлагаются ТОЛЬКО в Expert Mode.



ПРОСТЫМИ СЛОВАМИ

Сетка карточек с 6 форматами, актуальными в Simple Mode: PLY (стандартный формат для других 3D-инструментов), SPZ (сжатый вариант для веба), glTF (официальный Web3D-стандарт), .splat (для веб-просмотрщика antimatter15), orbit video (готовый MP4 для показа) и Web Viewer (автономный HTML-файл со встроенным 3D-плеером). Так покрываются 90 % вариантов использования. Если нужен один из менее распространённых форматов (compressed PLY или SOG для экстремального сжатия), переключитесь в Expert Mode, там доступны все 8 форматов. Компактный выбор здесь сделан намеренно, чтобы новички не были подавлены разнообразием.

**C-40 Кнопка Format Card**

ГДЕ

Каждая карточка в сетке.

**ТЕХНИЧЕСКИ**

Простая кнопка с card-раскладкой: значок (напр., document zipper для PLY, archive box для SPZ, video icon для orbit video) сверху, имя формата как заголовок, описание caption (обрезаётся до 2 строк), под ним оценочный размер файла (вычисленный вживую из формата, числа Gaussians и SH-степени, отформатирован как KB/MB). По щелчку формат выбирается. Выбранная карточка получает accent-фон, accent-обводку и значок check в правом верхнем углу. Tooltip — описание формата.

**ПРОСТЫМИ СЛОВАМИ**

По карточке на формат. Щёлкнете по одной, она выделяется акцентным цветом и галочкой, а кнопка экспорта ниже подстраивает свой текст («Export PLY», «Export SPZ» и т.д.). На каждой карточке — подходящий символ, имя, двустороннее короткое объяснение и оценочный размер файла для вашего текущего результата обучения. Размер помогает разумно выбрать — если хочется отправить результат по почте, возьмите наименьший вариант (обычно SPZ или .splat); если хочется продолжать работу в другой 3D-программе, возьмите тот, у кого лучшая совместимость (обычно PLY). При наведении на карточку tooltip показывает более подробное описание — если различия между форматами кажутся вам неясными.

**C-41 Слайдер длительности видео**

ГДЕ

Под сеткой форматов, виден только если выбран видеоформат (orbit video или social video).

**ТЕХНИЧЕСКИ**

Слайдер 3–30 секунд с шагом 1 секунда, привязывается к длине видео в app state. Максимальная ширина 300 пикселей. Показывается только если выбран видеоформат. Для не-видео форматов слайдер полностью удаляется из вида — без мёртвого пространства.

**ПРОСТЫМИ СЛОВАМИ**

Если в качестве экспорта вы выбираете orbit-видео, здесь можно задать длину. 3 секунды = очень быстрое вращение, 30 секунд = медленное, спокойное вращение вокруг вашей модели. Для социальных reels (Instagram, TikTok) обычно 6–10 секунд идеальны — достаточно долго, чтобы показать модель, достаточно коротко, чтобы зрители не отскочили. Для презентаций или портфолио-видео можно смело взять 15–20 секунд. Слайдер появляется только если выбран видеоформат; для файловых форматов вроде PLY или SPZ это было бы бессмысленно, и он скрыт.

**C-42 Кнопка Export**

ГДЕ

Под сеткой форматов (и под слайдером длительности, если выбрано видео).

**ТЕХНИЧЕСКИ**

Большая accent-кнопка. Метка: «Export {format-name}», значок share. По щелчку открывается save-диалог macOS с подходящим формату расширением и именем по умолчанию «scene.{ext}»; при подтверждении экспорт пишется в выбранный URL. Деактивирована, если нет результата обучения или уже идёт экспорт.

**ПРОСТЫМИ СЛОВАМИ**

Щелчок, выбор места сохранения в диалоге macOS, готово — приложение записывает файл в выбранном формате в выбранное место. Имя по умолчанию — «scene.{расширение}» (напр., «scene.ply» или «scene.spz»), его можно изменить в диалоге как угодно до сохранения. Кнопка серая, пока ещё нет результата обучения (здесь такого случиться не должно, иначе вы бы не были на шаге экспорта) или уже идёт другой экспорт. Как только экспорт идёт, под ним появляется индикатор прогресса; приложение остаётся работоспособным, так что вы можете уже готовить следующий экспорт.

**C-43 Прогресс-бар экспорта**

ГДЕ

Под кнопкой экспорта, виден только пока идёт экспорт.

**ТЕХНИЧЕСКИ**

Индикатор прогресса с максимальной шириной 300 пикселей, под ним подпись «Exporting... N %». Значение бежит от 0 до 1 и обновляется при записи — у PLY кусками по 10 000 Gaussians, у SPZ один раз после квантования, у orbit video — в frame-интервалах.

**ПРОСТЫМИ СЛОВАМИ**

Пока идёт экспорт, здесь видна тонкая полоса прогресса плюс отображение процентов. PLY обычно готов за секунды, потому что файл просто записывается двоично. SPZ длится чуть дольше, потому что данные при этом квантуются и сжимаются. Orbit video — самый затратный по времени экспорт — здесь каждый отдельный кадр перерисовывается; в зависимости от разрешения и длины может уйти минута или больше. Во время экспорта приложение остаётся работоспособным, так что можно уже готовить следующий формат или продолжать щёлкать в viewport.

## C-44 Отображение ошибки экспорта



ГДЕ

Под прогресс-баром, виден только если при последнем экспорте произошла ошибка.

### ТЕХНИЧЕСКИ

Красная строка со значком warning и текстом ошибки. Красный фон с 8 % opacity, скруглённые углы. Максимальная ширина 400 пикселей. Распространённые причины ошибок: SOG ожидает `cwebp` в системном PATH (не совместимо с App Store); ошибка записи при полном диске; sandbox-ошибка с целями сохранения вне разрешённой области.

### ПРОСТЫМИ СЛОВАМИ

Если экспорт срывается, здесь красным появляется краткое описание проблемы простым языком. Обычно причина очевидна — нет места на диске, нет прав на запись в целевую папку, или цель вне разрешённых sandbox-областей. Конкретно с форматом SOG бывает, что `cwebp` отсутствует в системе; в этом случае SOG непригоден, и придётся откатиться к SPZ. Если сообщение об ошибке неясно, посмотрите в лог-директорию (Help → Open Training Logs), там подробнее описано, что пошло не так. В сомнительных случаях помогает просто выбрать другое место сохранения — напр., рабочий стол.

## C-46 Список истории экспортов



ГДЕ

Правая сторона шага экспорта.

### ТЕХНИЧЕСКИ

Список по истории экспортов (персистентен как JSON в UserDefaults, поддерживается после каждого успешного экспорта). Каждая строка показывает format badge (маленький, accent-цвета), timestamp (HH:mm), имя файла (обрезается до 1 строки) и форматированный размер файла. Щелчок по строке открывает Finder с выбранным файлом. Пустое состояние: «No exports yet».

### ПРОСТЫМИ СЛОВАМИ

Список ваших предыдущих экспортов — формат, время, имя файла, размер, в хронологическом порядке. Щелчок по строке, и файл подсвечивается в Finder — без необходимости самому навигироваться по папкам. Удобно, когда час спустя снова нужен последний экспорт, и вы уже не знаете, куда сохранили — история это помнит. Если вы ещё ничего не экспортировали, здесь стоит дружелюбная подсказка «No exports yet». Список переживает перезапуски приложения, потому что хранится в UserDefaults.

**C-48 История контекстного меню (правый клик)****ГДЕ**

Правый клик по строке истории.

**ТЕХНИЧЕСКИ**

Контекстное меню на каждой записи списка с двумя действиями: «Reveal in Finder» (открывает Finder с выбранным файлом, как одиночный щелчок) и «Copy Path» (помещает полный путь файла как текст в буфер обмена). Последнее полезно для drag-and-drop в другие приложения или для передачи в командную строку.

**ПРОСТЫМИ СЛОВАМИ**

Правый клик по записи истории открывает маленькое меню с двумя действиями. «Reveal in Finder» делает то же самое, что обычный щелчок — открывает Finder с выбранным файлом, чтобы вы сразу его увидели. «Copy Path» помещает полный путь файла в буфер обмена, чтобы можно было его, напр., вставить в команды терминала, в другие приложения или в заметку. Особенно удобно, когда вы хотите передать экспорт кому-то или открыть в другой программе, работающей с вводом пути. Функционально маленькая, но полезная деталь, делающая ставку на типичные для Mac паттерны взаимодействия.

**Когда рабочий процесс завершён?**

После успешного экспорта у вас есть ваша 3D-модель файлом на диске, а в истории появляется новая запись. Нет кнопки «Done» — можно последовательно сделать любое число экспортов в разные форматы без повторного обучения. Если хочется вернуться к предпросмотру (напр., чтобы ещё раз проверить перспективу камеры), используйте кнопку Back в нижней navigation-панели. Если хочется начать совершенно новую сцену, через Back вернитесь к Z1 и используйте там Clear All, либо File → New Project (Cmd+⇧+N).

**Переключение в Expert Mode**

В любой момент нажмите Cmd+2 или выберите Mode → Expert Mode ( M8 ). Сохраняется всё состояние: импортированные изображения, выбранная предустановка, идущее или законченное обучение, готовое облако точек, история экспортов, даже текущий этап. В Expert Mode вместо четырёхшагового этапа показывается полная боковая панель Инспектора со всеми ~150 полями управления. В частности: Project Navigator (см. главу 2) предлагает расширенные операции с изображениями (кнопка минус, удаление backspace, отмена Cmd-Z, предпросмотр Quick Look), живой предпросмотр в viewport во время обучения, а также все параметры loss, MCMC, densification и Mip-Splatting. Cmd+1 переключает обратно в Simple Mode — при этом тоже не теряется никакое состояние.

## Часто задаваемые вопросы

### Почему моя кнопка **Start Processing** остаётся серой?

Вы ещё не импортировали ни изображений, ни видео. Перетащите хотя бы один файл в зону дропа или используйте «Browse Files». Как только список изображений справа содержит хотя бы одну запись, кнопка становится активной. (С всего 1–2 изображениями старт сработает, но SfM сразу прервётся с ошибкой — см. красный валидационный баннер.)

### Почему моя кнопка **Export** заблокирована?

В Simple Mode есть два уровня: (a) Если training-конвейер ещё не завершён и у вас его нет, кнопка деактивирована — сначала надо завершить Z2. (b) Если полная версия ещё не куплена (`PurchaseManager.hasAccess == false`), вы вместо этапа экспорта видите lock-вид со значком замка и кнопкой «Unlock Full Version», открывающей purchase-sheet. Предусстановки Quick и Preview позволяют обучаться бесплатно, но экспорт — премиум.

### Почему я не могу выбрать предустановку?

Можете выбрать — но если нажмёте на premium-предустановку (Balanced, Quality, варианты MCMC) без купленной полной версии, picker автоматически отскочит обратно к Preview, и откроется purchase-sheet. Quick и Preview — единственные свободно используемые предустановки.

### Почему моя зона дропа пустая и пунктирно-серая, хотя я перетаскиваю в неё изображения?

Вероятно, расхождение типа UTI. Приложение принимает JPG, PNG, TIFF, HEIC, MP4, MOV плюс собственные splat-форматы приложения. Другие форматы изображений (BMP, GIF, WebP, RAW-форматы) НЕ распознаются. Если уверены, что ваш тип изображения должен включаться, проверьте расширение имени файла — приложение в первую очередь ориентируется на расширение, а не на содержимое файла.

### Почему SfM длится так долго, хотя у меня всего 30 изображений?

Apple Photogrammetry не масштабируется линейно — при определённых конstellациях изображений (помещения со сложными текстурами, motion blur, плохой свет) она длится значительно дольше, чем подсказывает число изображений. Если SfM с 30 изображениями всё ещё висит через 10+ минут, прервите и попробуйте снова с лучшим материалом, либо переключитесь в Expert Mode и попробуйте COLMAP/Native SfM (Cmd+2 → Inspector → Camera Alignment).

### Где я найду свои training-логи?

Help → Open Training Logs (Cmd+⇧+L). Это откроет ~/Documents/RadianceKit/Logs/. Каждая сессия обучения пишет собственный JSONL-файл с timestamp в имени файла — первая строка — конфигурация, затем каждые 100 итераций идёт progress-строка, последняя строка — summary с финальным loss и success-флагом.



#### КОЛОФОН

*Набрано в SF Pro · Код в SF Mono · Типст 0.14 ·  
22. June 2026*

© 2026 Bjoern Kindler · Bischofshofener Str. 9, 82008 Unterhaching, Германия

Made with ❤️ in Unterhaching