FROM STAINS TO RGB: REVISITING RENDERING FOR CELL PAINTING GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating Cell Painting images requires balancing computational cost with biological utility. To reduce training costs, some existing approaches render multichannel fluorescence images into RGB and compress them using variational autoencoders (VAEs) pretrained on natural images for latent-space generation. Yet the separate effects of rendering and VAE compression on cellular feature recoverability and their implications for generative training remain unclear. Here we introduce CPGENeval, a benchmark that evaluates seven RGB rendering protocols and five VAEs across two datasets and compares pixel- and latent-space generation under a common training setup. Alongside FID, we measure whether cellular features computed from the original fluorescence channels can be recovered from RGB images and their VAE reconstructions, and whether generated images can be matched to real images of the same siRNA perturbation (perturbation retrieval). We find that (i) renderings that rank best by reconstruction FID (rFID) can leave native cellular features poorly recoverable; (ii) changing the rendering substantially changes generation FID (gFID) for the same generated samples, and even within a fixed rendering, gFID and perturbation retrieval can rank generators differently; and (iii) with appropriate rendering, RGB latent-space generation achieves stronger perturbation retrieval than pixel-space generation under most protocols, in substantially less training time. These findings call for reporting the rendering protocol with every FID, holding it fixed across compared models, and validating it with feature recovery and perturbation retrieval.

1 INTRODUCTION

Cell Painting uses multichannel fluorescence microscopy to characterize how cellular morphology changes in response to chemical or genetic perturbations (Seal et al., 2025). Cell Painting screens are costly at scale, and image coverage per experimental condition is often limited. A growing body of work therefore develops generative models to simulate cellular morphological responses to perturbations and complement experimental screens. These methods draw on generative frameworks widely used in computer vision: GAN-based *IMPA* (Palma et al., 2025) and *Phenexplain* (Lamiable et al., 2023); diffusion-based *MorphoDiff* (Navidi et al., 2025), *MorphDiff* (Wang et al., 2025), *MorphGen* (Demirel et al., 2025), and *PhenDiff* (Bourou et al., 2024); bridge-based *FreeBridge* (Wang et al., 2026); and flow-matching *CellFlux* (Zhang et al., 2025) with its extensions *CellFluxV2* (Zhang et al., 2026) and *CellFluxRL* (Wu et al., 2026). Their practical value depends on the biological utility of their outputs and on the cost of training them.

Both depend on how Cell Painting images are represented. Cell Painting records each stain as a separate grayscale channel, giving complementary views of cellular structures, whereas pretrained natural-image variational autoencoders (VAEs) and the Inception network used by Fréchet Inception Distance (FID) typically expect RGB inputs. RGB rendering enables latent-space generation with pretrained VAEs (Navidi et al., 2025) and FID-based evaluation and model selection (Demirel et al., 2025). At least six of the eleven generators we survey compute FID on RGB renderings (Appendix Table 5). Native-channel alternatives either generate directly in pixel space, forgoing VAE compression (Zhang et al., 2025; Jones et al., 2026), or encode each stain separately, so the latent grows with the number of stains (Demirel et al., 2025). **The representation thus shapes both the cellular information available to the generator and the cost of modeling it. In RGB latent-space**

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

pipelines, it also shapes evaluation: a single rendering protocol determines both the images encoded for training and the reference images for FID.

Rendering protocols, however, differ in channel selection, intensity calibration, and mixing, which determine which stains appear in the RGB image, over what intensity range, and in which colors (Fig. 1, bottom row). Applied to IMPA-preprocessed U2OS crops (IMPA96), RECURSION's intensity gains, calibrated for RxRx1, produce large, nearly uniform bright regions that obscure the subcellular structure visible in the native channels (Fig. 1, top row). Yet this protocol yields the lowest reconstruction FID (rFID) among seven protocols for four of five VAEs. Prior work has documented FID's sensitivity to preprocessing and feature representations (Parmar et al., 2022; Kynkäänniemi et al., 2023; Stein et al., 2023); here, rFID compares reconstructions with an already-rendered reference and does not directly measure native cellular feature recoverability. For generation, we complement generation FID (gFID) with retrieval-based evaluation (Kalinin et al., 2025; Wang et al., 2025), which measures how well generated profiles retrieve real images of the same perturbation, reported as mean average precision (mAP). Within a fixed rendering protocol, gFID and retrieval mAP can rank generators differently despite sharing frozen Inception features.

We introduce CPGENeval, a benchmark for choosing representations and generators in Cell Painting generation. It standardizes seven published rendering protocols, scores renderings and their VAE reconstructions against fixed native CellProfiler targets, and pairs matched-rendering gFID with siRNA retrieval, chance levels, and paired intervals. We apply it to 70 reconstruction configurations, 72 trained generators on two dataset settings, and released outputs from prior work. Using this benchmark, we make the following three contributions:

1. **Rendering and reconstruction.** We separate rendering from VAE reconstruction by predicting fixed native radial profiles from RGB inputs and their VAE reconstructions, and identify renderings that rank best by rFID despite weak native cellular readout.
2. **Generation evaluation.** We isolate the effect of rendering on gFID by evaluating the same generated samples under all seven protocols, and show that changing the rendering substantially changes gFID. Pairing gFID with siRNA retrieval, we further show that even within a fixed rendering, the two can rank generators differently. These reversals expose the consequences of fidelity-only model selection.
3. **Training cost and retrieval.** We find that the best RGB latent configurations deliver stronger siRNA retrieval than native pixel-space generation in substantially less training time under most protocols, and that extended pixel-space training narrows this gap substantially but leaves it open under five of seven protocols.

These findings lead to three practical recommendations for Cell Painting generation: validate RGB renderings through native readout and retrieval, report the rendering protocol alongside every FID, and hold it fixed for FID-based model comparisons.

2 PRELIMINARIES

2.1 MULTICHANNEL-TO-RGB RENDERING

A rendering protocol R_ρ maps an 8-bit multichannel image x to an RGB image $y_\rho = R_\rho(x)$ at the same spatial resolution, $R_\rho : \{0, \dots, 255\}^{C\times H\times W} \to \{0, \dots, 255\}^{3\times H\times W}$. Our inputs have $C = 6$ stains in RxRx1 order: DNA, ER, Actin, RNA, Mito, and Golgi. We evaluate seven source-derived protocols, written in small capitals to distinguish them from the italicized source models (e.g., CELLFLUX is the rendering released with *CellFlux*). They differ in three main choices (Appendix Table 2): *selection* (IMPA and FREEBRIDGE keep three stains; MORPHDIFF and CELLCLIP drop Golgi), *mixing* (stacking or weighted combination), and *calibration* (fixed, image-adaptive, or none).

Fixed versus adaptive calibration. RECURSION and CELLCLIP use fixed per-stain upper bounds, $\bar{x}_k = \min(x_k/h_k, 1)$, mapping all intensities above h_k to 1. CELLFLUX applies a fixed affine map. FREEBRIDGE, MORPHGEN, and MORPHDIFF instead rescale each image, before or after mixing; IMPA adds no calibration at the RGB mapping stage. Fixed calibrations are tied to the intensity regime of their original inputs: raw RxRx1 sites for RECURSION and IMPA-preprocessed crops for CELLFLUX. Applying them to differently preprocessed data can therefore saturate or suppress intensities (Fig. 1).

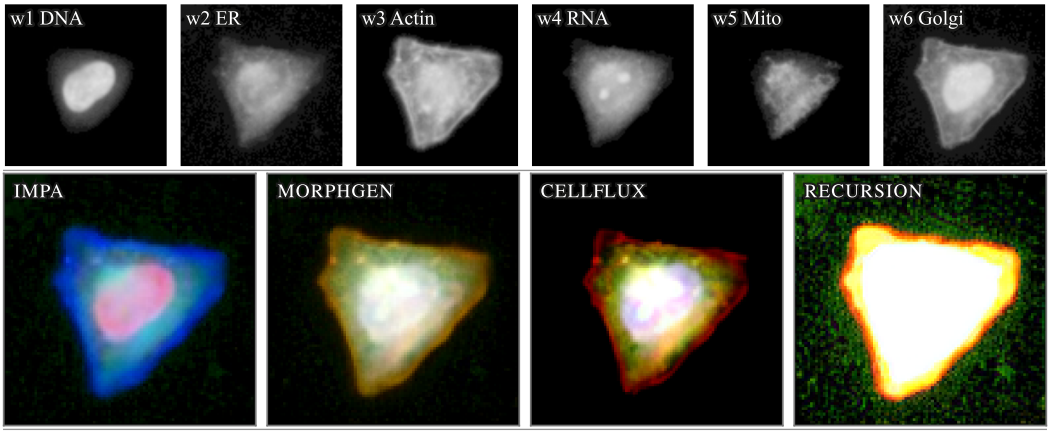


Figure 1: **rFID can favor a rendering that obscures cellular detail.** One IMPA96 cell: six native channels (top) and four of the seven RGB renderings (bottom), with dataset-level rFID and native readout (R^2_{rec}; six-channel reference 0.552) of SD3.5 reconstructions. RECURSION's fixed gains, calibrated for raw RxRx1 sites, saturate these preprocessed crops (63.68% near-white pixels) and leave almost no native readout ($R^2_{\text{rec}} = 0.007$), yet rFID ranks it first among all seven renderings for four of five VAEs. Bold: best; red: lowest R^2_{rec}.

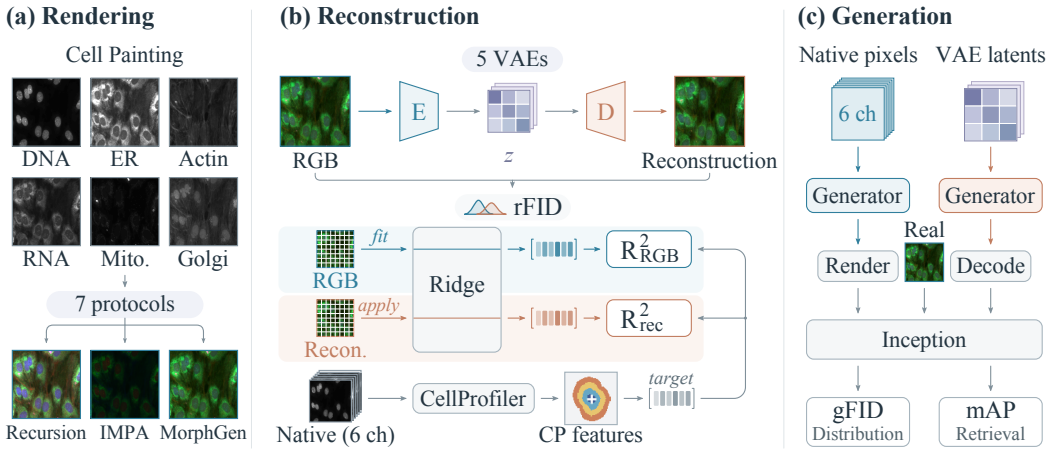


Figure 2: **CPGENeval evaluates each stage against an explicit reference.** (a) Seven published protocols render the six stains to RGB. (b) Five frozen VAEs reconstruct each rendering. rFID compares inputs with reconstructions; a ridge regressor fitted on RGB inputs predicts native CellProfiler profiles and is applied unchanged to reconstructions (R^2_{RGB},R^2_{rec}). (c) Pixel outputs are rendered and latent outputs decoded; gFID and siRNA retrieval compare them with real images under the same protocol and Inception features.

2.2 RECONSTRUCTION AND GENERATION

Conditional generation samples cell images given a condition c, such as perturbation information or a reference image (Demirel et al., 2025; Zhang et al., 2026). In our experiments, c denotes the siRNA identity, which is mapped to a learned embedding.

Existing approaches follow three channel-handling strategies: (i) *direct multichannel pixel-space generation*, as in *CellFlux* (Zhang et al., 2025) and *MiT* (Jones et al., 2026); (ii) *RGB rendering followed by VAE encoding*, as in *MorphoDiff* (Navidi et al., 2025); and (iii) *grouped or per-stain encoding*, used for reconstruction in *Compressing Biology* (Cropsal & Mercado, 2025) and for

generation in *MorphGen* (Demirel et al., 2025). The third strategy supports multichannel outputs without first mixing the stains into RGB. Our main comparison uses a shared backbone family for strategies (i) and (ii).

With $y_\rho = R_\rho(x)$, encoder E, and decoder D, RGB reconstruction, pixel-space generation followed by rendering, and RGB latent generation are expressed as:

$$\hat{y}_\rho = D\big(E(y_\rho)\big), \qquad \text{(reconstruction)} \tag{1a}$$

$$\tilde{y}_\rho = R_\rho(\tilde{x}), \quad \tilde{x} \sim p_\theta(x \mid c), \qquad \text{(pixel-space generation)} \tag{1b}$$

$$\tilde{y}_\rho = D(\tilde{z}), \quad \tilde{z} \sim p_\theta(z \mid c), \qquad \text{(latent-space generation)} \tag{1c}$$

Reconstruction decodes the encoding of an observed RGB image; latent generation decodes a new sample from the learned conditional distribution.

2.3 Evaluation

Let X_{rec} and X_{eval} denote the multichannel images used for reconstruction and generation evaluation, with R_ρ, E, and D applied to each image, and let $\mathrm{FID}(A, B)$ denote the Fréchet Inception Distance between image sets A and B (Heusel et al., 2017; Szegedy et al., 2016). For protocol ρ,

$$\mathrm{rFID}_\rho = \mathrm{FID}\big(R_\rho(X_{\mathrm{rec}}),\ D \circ E \circ R_\rho(X_{\mathrm{rec}})\big), \qquad \mathrm{gFID}_\rho = \mathrm{FID}\big(R_\rho(X_{\mathrm{eval}}),\ \tilde{Y}_\rho\big), \tag{2}$$

where $\tilde{Y}_\rho$ is the set of generated images $\tilde{y}_\rho$ from Eq. 1b or Eq. 1c. Both scores compare against rendered references. Changing the protocol therefore changes both distributions, and neither score can detect information that rendering removes, such as discarded stains or saturated regions.

We pair each score with a measure whose ground truth does not depend on the rendering. For reconstruction, a ridge regressor predicts CellProfiler radial intensity profiles, measured on the six native stains (Carpenter et al., 2006), from grid statistics of the rendered RGB image. Evaluating it on rendered inputs and, unchanged, on their reconstructions gives R^2_{RGB} and R^2_{rec} (Section 3). For generation, the mean Inception profile of each siRNA's generated images queries the real evaluation wells, and mAP (Kalinin et al., 2025) scores how highly wells of the same siRNA rank. Retrieval uses the same features as gFID but tests correspondence to each perturbation, which pooled gFID does not assess (Section 4). A regressor fitted on the native channels and retrieval with real training profiles as queries provide reference values.

3 Reconstruction Fidelity Is Not Feature Recoverability

We examine whether **selection by rFID favors stronger native readout** among RGB renderings and VAEs. rFID evaluates reconstruction against a rendered RGB reference; native readout assesses the predictability of radial intensity profiles of the original stains.

Experimental design. We evaluate five frozen natural-image VAEs: the SD1.4 autoencoder (Rombach et al., 2022), its MSE fine-tune SD ft-MSE (Stability AI, 2022), SDXL (Podell et al., 2024), SD3.5 (Esser et al., 2024; Stability AI, 2024), and FLUX.1 (Black Forest Labs, 2024). Rendered inputs and their reconstructions are assessed against the same native targets: 24 CellProfiler radial intensity fractions, four per stain, describing fluorescence within nuclear and perinuclear regions rather than directly measuring cell shape (Carpenter et al., 2006). A separate ridge reader predicts these targets from RGB patch statistics for each dataset–protocol pairing. Each experimental batch is held out in turn. We fit regressors on RGB inputs from the remaining batches and apply them unchanged to paired RGB inputs and VAE reconstructions in the held-out batch. Prediction R^2, averaged across the 24 targets, is reported as R^2_{RGB} and R^2_{rec}. The difference, $\Delta R^2 = R^2_{\mathrm{rec}} - R^2_{\mathrm{RGB}}$, records the additional readout change after reconstruction. Appendix E details feature extraction, regressor training, and hierarchical bootstrap intervals.

Low rFID can reflect fidelity to an altered reference. Natural-image VAEs can closely match the Inception feature statistics of rendered HUVEC512 images without Cell Painting-specific fine-tuning.

Table 1: Rendering and VAE reconstruction across 70 configurations. Native readout R^2_{RGB} measures radial-profile recoverability from rendered RGB inputs. Each VAE reports rFID and R^2_{rec}; the latter uses the corresponding regressor trained on RGB inputs. Readout confidence intervals are provided in Appendix E.2.

Protocol	RGB	SD1.4		SD ft-MSE		SDXL		SD3.5		FLUX.1	
	$R^2_{\text{RGB}}\uparrow$	rFID ↓	$R^2_{\text{rec}}\uparrow$	rFID ↓	$R^2_{\text{rec}}\uparrow$	rFID ↓	$R^2_{\text{rec}}\uparrow$	rFID ↓	$R^2_{\text{rec}}\uparrow$	rFID ↓	$R^2_{\text{rec}}\uparrow$
HUVEC512											
MorphGen	0.621	**5.036**	0.584	3.314	0.587	4.703	0.588	0.740	0.615	**0.190**	0.619
CellFlux	0.182	5.120	0.159	2.841	0.172	**1.434**	0.177	0.923	0.178	1.309	0.171
Recursion	0.652	5.903	0.622	8.194	0.627	10.532	0.638	**0.531**	0.648	0.239	0.649
IMPA	0.606	5.261	0.487	**2.069**	0.484	1.642	0.557	2.436	0.529	0.904	0.581
FreeBridge	0.549	7.478	0.524	5.126	0.528	7.565	0.532	0.728	0.537	1.405	0.546
MorphDiff	0.584	5.851	0.553	2.954	0.552	5.260	0.563	0.785	0.572	0.271	0.579
CellCLIP	0.646	5.574	0.615	7.121	0.618	9.109	0.631	0.535	0.642	0.237	0.643
IMPA96											
MorphGen	0.496	56.874	0.422	49.723	0.462	37.250	0.474	27.704	0.480	21.130	0.492
CellFlux	0.441	43.353	0.407	35.974	0.417	33.649	0.422	23.827	0.422	22.900	0.432
Recursion	0.041	**20.077**	0.033	**20.760**	0.057	**18.676**	0.048	**15.356**	0.007	25.774	0.054
IMPA	0.558	31.408	0.426	39.003	0.508	27.265	0.543	25.035	0.528	**17.121**	0.553
FreeBridge	0.512	32.450	0.463	42.405	0.483	35.844	0.494	34.680	0.485	25.656	0.509
MorphDiff	0.524	41.562	0.409	42.321	0.477	30.699	0.504	25.107	0.494	23.330	0.519
CellCLIP	0.149	20.635	0.153	23.826	0.165	21.260	0.157	16.402	0.129	25.650	0.160

Native references use a separate ridge regressor with grid statistics from the six original channels: $R^2 = 0.658$ (HUVEC512) and 0.552 (IMPA96); these are not upper bounds. Within each dataset and metric column, bold marks the lowest rFID, underlining the highest R^2, and red the highest rFID or lowest R^2. rFID uses the full training cohorts; R^2 uses valid-object subsets paired across native, RGB, and reconstruction.

FLUX.1 records rFID below 1 under five HUVEC512 protocols. It also has the lowest rFID under five of seven protocols in each dataset, with SD3.5 leading under the other two (Table 1). This agreement can persist even when rendering obscures intracellular detail. Recursion leaves an average of 63.68% of pixels near-white when transferred to IMPA96 (Appendix Table 3). These bright regions persist after SD3.5 reconstruction (Fig. 1), yet rFID ranks this rendering first for four of five VAEs, including SD3.5 (15.36 vs. 25.04 under IMPA).

CellFlux illustrates the opposite intensity extreme. Its HUVEC512 renderings average 99.63% near-black pixels, yet SDXL gives this protocol the lowest rFID among the seven renderings (1.43).

Finding

Looks worst, scores best. Under protocol transfer, renderings that obscure cellular detail can rank first by rFID.

Native readout separates rendering from reconstruction. Recursion ranks last by RGB readout on IMPA96 ($R^2_{\text{RGB}} = 0.041$), while IMPA ranks first (0.558). The same Recursion protocol nearly matches the six-channel reference on HUVEC512 (0.652 vs. 0.658). The highest R^2_{rec} among the five VAEs stays within 0.025 of R^2_{RGB} in all 14 dataset–protocol settings. Even the reconstructions with the highest readout scores remain weak when their RGB inputs have weak readout.

The unchanged regressor also distinguishes VAE effects on the same RGB representation. Reconstructing IMPA inputs on HUVEC512 changes readout by $\Delta R^2 = -0.025$ with FLUX.1 and -0.077 with SD3.5. FLUX.1 also has the lower rFID (0.90 vs. 2.44), so the two criteria agree in this comparison.

rFID-based selection differs within and across renderings. The VAE chosen by rFID has the highest R^2_{rec} in 11 of 14 dataset–protocol settings, with a maximum observed shortfall of 0.050 from the best VAE (Table 1). The agreement falls to 2 of 10 dataset–VAE settings when choosing the rendering for a given VAE. Five settings show complete ranking reversals, in which the lowest-rFID rendering has the lowest R^2_{rec}, falling 0.430–0.521 short of the best rendering. Renderings should therefore be checked against native targets before rFID is used to compare VAE reconstructions.

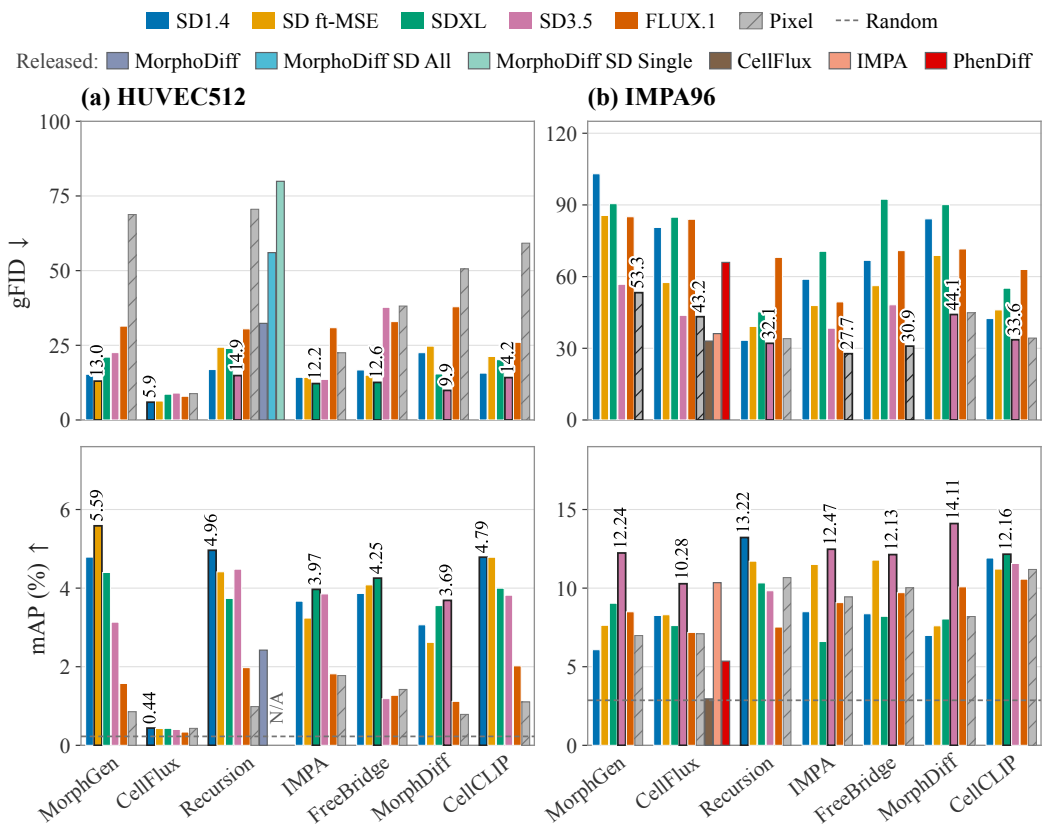


Figure 3: **Generation fidelity and siRNA retrieval.** (a) HUVEC512; (b) IMPA96. Outlined bars mark the best score among five RGB latent generators and one six-channel pixel generator within each protocol at 100k updates. Dashed lines indicate random-label retrieval. Released baselines use their corresponding rendering protocols and retain their original training scopes. MorphoDiff's unconditional SD All and SD Single variants have no siRNA retrieval score (N/A). CellFlux, IMPA, and PhenDiff results use image pools distributed with the CellFlux release.

Finding

The cost of choosing by rFID. The largest R^2_{rec} shortfall from the best candidate was 0.05 when choosing a VAE by rFID and 0.52 when choosing a rendering.

4 Generation Fidelity Can Favor Weaker Retrieval

We next ask whether choosing generators by gFID favors stronger siRNA retrieval. Pooled gFID compares generated and real images without condition labels; retrieval tests how highly each generated condition profile ranks real wells of the same siRNA.

Experimental design. For each dataset, we train 35 RGB latent-space generators, one for each pairing of the seven rendering protocols with the five frozen natural-image VAEs, and one six-channel pixel-space generator whose outputs are rendered under all seven protocols. All generators use a DeCo-inspired backbone with representation-specific adaptations (Ma et al., 2026), are trained for 100k updates with one seed per configuration, and are sampled from non-EMA weights using 50 Euler steps without classifier-free guidance. On HUVEC512, training uses batches 01–16 and evaluation covers 1,032 siRNA conditions from held-out batches 17–24 (Sypetkowski et al., 2023). On IMPA96, evaluation uses the 100-perturbation subset released with CellFlux (Zhang et al., 2025), whose test crops share wells and source sites with training. IMPA96 retrieval therefore measures condition correspondence within this benchmark, while the HUVEC512 comparison tests that correspondence

across held-out batches. Paired bootstrap intervals for the retrieval differences below are reported in Appendix Table 10.

Rendering changes the score of the same generator. Changing the evaluation rendering moves the pixel-space generator's gFID from 8.8 to 70.6 on HUVEC512 and from 27.7 to 53.3 on IMPA96 (Fig. 3). For each dataset, the seven Pixel bars use the same generated samples and real images, with both rendered under each protocol in turn. These score differences reflect the evaluation rendering, not differences in the underlying generator.

Shared features, reversed rankings. gFID and siRNA retrieval use the same rendering and a shared frozen Inception feature extractor, yet they can rank generators in opposite orders. Under IMPA on IMPA96, gFID favors the pixel-space generator over the SD3.5 latent generator (27.74 vs. 38.41), but retrieval favors SD3.5 (12.47% vs. 9.44% mAP; Fig. 3b). The retrieval advantage is 3.03 percentage points (paired 95% CI 0.06–6.07). Point-estimate rankings show this pixel–latent reversal under four of the seven IMPA96 protocols. The best latent configurations lead on both criteria on HUVEC512, although retrieval is nearly tied under CELLFLUX. Pooled gFID ignores condition labels, so a closer match to the overall distribution can coexist with weaker perturbation correspondence.

VAE choice mostly agrees. Restricting the comparison to latent generators reveals more agreement: the lowest-gFID VAE also retrieves best in five of seven protocols on each dataset. The exceptions are RECURSION and CELLCLIP. Under RECURSION on IMPA96, gFID favors SD3.5 over SD1.4 (32.08 vs. 33.35; Fig. 3), yet SD1.4 retrieves 3.39 percentage points better (paired 95% CI 0.85–5.89; Appendix Table 10).

Released models show the same reversal. Selecting by gFID among the three released IMPA96 image pools (Zhang et al., 2025) would pick *CellFlux* (33.02), ahead of *IMPA* (36.14). Retrieval reverses this order: *IMPA* reaches 10.35% mAP compared with 2.95% for *CellFlux* (Fig. 3b), a difference of 7.40 percentage points (paired 95% CI 4.98–9.99). Our own generator is competitive under this rendering. Our SD3.5 generator reaches 10.28% mAP under the same CELLFLUX rendering, close to *IMPA*'s retrieval despite a higher gFID of 43.78. On HUVEC512, with the same SD1.4 VAE and RECURSION rendering, our generator has lower gFID (16.88 vs. 32.35) and higher retrieval mAP (4.96% vs. 2.42%) than the released *MorphoDiff* checkpoint, even though *MorphoDiff*'s training included our evaluation batches (Navidi et al., 2025).

Finding

Best gFID, weaker retrieval. Even with the same rendering and feature extractor, choosing the lowest-gFID candidate can favor weaker siRNA retrieval.

5 THE LATENT ADVANTAGE IS MOSTLY TRAINING SPEED

Finally, we ask what RGB latent generation offers over native pixel generation. RGB latent generators reuse pretrained natural-image VAEs but fix the rendering before training; native pixel generators produce all six channels and leave the rendering open afterwards. At 100k updates, the best RGB latent configurations achieve stronger retrieval under most protocols (Section 4). We compare the encoding and training costs of the two routes, test whether longer pixel training closes the retrieval gap, and ask whether the VAE with the lowest rFID also yields the latent generator with the strongest siRNA retrieval.

Latent generators take less time per update. Both routes share the backbone family, learning rate, batch size, and conditioning, and the pixel generator is the larger of the two (134.5M vs. about 108.5M parameters). Including VAE encoding, RGB latent generators reach 100k updates in about a fifth of the native pixel training time on HUVEC512 (6.01–6.08 vs. 27.96 H100 hours) and under a third on IMPA96 (1.30–1.36 vs. 4.43 hours; Fig. 4a). Encoding runs once per VAE and rendering, and later updates reuse the cached latents, so longer latent-space runs add only training time. Most of the pixel cost lies in decoding six channels at full resolution (Appendix D.6). At this budget the retrieval gap is wide: under MORPHGEN, SD ft-MSE reaches 5.59% mAP at a gFID of 13.0, against 0.85%

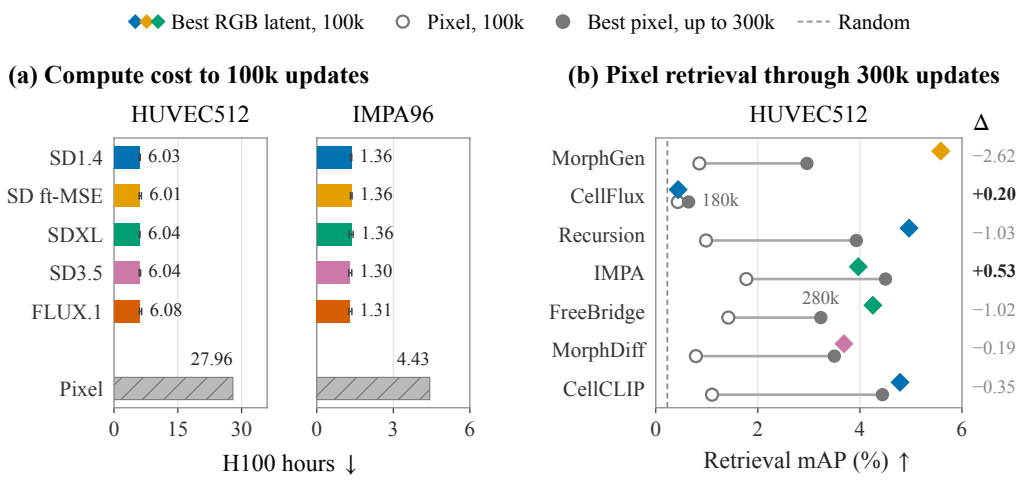


Figure 4: **RGB latent generators reach stronger retrieval with less training.** (a) H100 hours to 100k updates; latent times include VAE encoding (median per VAE). (b) HUVEC512 retrieval: best RGB latent at 100k (diamonds, VAE colors), pixel at 100k (open circles), and best pixel checkpoint per protocol through 300k (filled circles, labeled if before 300k). Dashed line: random. Δ: best pixel mAP through 300k minus best latent mAP at 100k, in percentage points; positive values (bold) mark protocols where pixel training overtakes.

and 68.8 for the pixel generator (Fig. 3). Only under CELLFLUX, whose HUVEC512 renderings are almost entirely black, do all generators retrieve alike, below 0.5%.

Longer pixel training narrows the gap. We extend the HUVEC512 pixel generator to 300k updates and evaluate every checkpoint under all seven protocols (Fig. 4b). Pixel retrieval keeps improving and is still rising under five protocols when training stops. Its best checkpoint overtakes the best 100k latent configuration under CELLFLUX and IMPA and trails under the other five, by less than half a point under MORPHDIFF and CELLCLIP and most under MORPHGEN. Most of the latent lead at 100k thus closes with longer training, but the pixel route needs about 14 times the compute of a latent run to get there.

Finding

A head start at a fraction of the compute. Latent generators retrieve better at matched updates; native pixel generators close most of the gap, but only with far more training.

Reconstruction rankings do not carry over to generation. The efficiency advantage makes RGB latent generation attractive, but leaves open which VAE to use. Reconstruction fidelity offers a ranking before generator training, yet the lowest-rFID VAE never produces the strongest siRNA retrieval at 100k updates in any of the 14 dataset–protocol settings (Table 1; Fig. 3). FLUX.1 leads reconstruction under most protocols on both datasets but produces the weakest latent generator under six of seven HUVEC512 protocols. SD3.5 provides the strongest retrieval under five IMPA96 protocols, so the generation ranking also changes with the data setting. The two 16-channel VAEs take every first place in reconstruction rFID, but neither latent width leads generation on both datasets. RGB latent generation thus offers substantial savings in training time while requiring two choices before training: the rendering and the VAE. Reconstruction fidelity alone cannot identify which combination will deliver the strongest retrieval.

Finding

Reconstructs best, not necessarily generates best. The VAE that reconstructs a rendering most faithfully did not yields the generator that retrieves best.

6 Related Work

Sensitivity of FID. FID estimates depend on preprocessing (Parmar et al., 2022), sample size (Chong & Forsyth, 2020), and the feature representation (Kynkäänniemi et al., 2023; Stein et al., 2023), and work on conditional GANs has proposed evaluations that account for the condition (Miyato & Koyama, 2018; Shmelkov et al., 2018; DeVries et al., 2019). These studies address natural RGB images. In Cell Painting pipelines built on RGB renderings, the rendering protocol determines both the images encoded for training and the reference images for FID, so it can change scores without any change to the generator. We therefore compare generators under a matched rendering and pair gFID with siRNA retrieval (Kalinin et al., 2025), which tests correspondence to each perturbation.

Channel handling and compression. Latent-space generation with pretrained autoencoders (Razavi et al., 2019; Esser et al., 2021; Rombach et al., 2022) reduces the cost of generative training, and Cell Painting pipelines reuse natural-image VAEs after rendering to RGB (Navidi et al., 2025). Other approaches generate native channels in pixel space (Zhang et al., 2025; Jones et al., 2026), encode stains in groups or separately (Cropsal & Mercado, 2025; Demirel et al., 2025), or train Cell Painting-specific VAEs (Zhang et al., 2026). Compressing Biology (Cropsal & Mercado, 2025) evaluates reconstruction by the Stable Diffusion VAE on grouped channels of JUMP images. CPGENeval instead varies both the rendering and the VAE, separates their effects against native targets, and compares latent-space with pixel-space generation under a common training setup.

7 Limitations

Our study covers seen siRNA perturbations in two six-channel RxRx1 settings, one with held-out batches. Our results describe the protocols as applied here rather than within their original pipelines. Extending the benchmark to five-channel assays such as JUMP (Chandrasekaran et al., 2023) and to unseen perturbations would test how far the findings generalize. Our generators condition on perturbation identity and use one backbone family with one seed per configuration, and training-time comparisons reflect our implementations of both routes (Appendix D.6). Generators that map control cells to perturbed cells (Palma et al., 2025; Zhang et al., 2025; 2026), per-stain latent encoding (Demirel et al., 2025), and Cell Painting-specific VAEs remain untested; for CellFluxV2, we found no public checkpoint of its VAE at the time of this study. Native readout targets radial intensity distributions rather than stain-specific signals, and retrieval uses the same Inception features as gFID; microscopy-specific encoders could extend both.

8 Conclusion

We introduced CPGENeval to follow Cell Painting images from their stains to RGB, through VAE reconstruction, and into generative training. One choice recurs at every stage. The rendering determines what a pretrained VAE encodes, what a latent generator can produce, and what FID compares, yet the scores commonly used to choose it measure fidelity to the rendered images rather than to the cells. Low rFID can reward a rendering that has already discarded cellular detail, and the VAE that reconstructs best never yields the generator that retrieves best. Re-rendering the same generated samples changes gFID eightfold, and within one rendering, gFID can favor generators whose samples match their perturbations less well. Latent generation remains the faster route to strong retrieval, although longer pixel training narrows its lead substantially; the latent route also requires the rendering and the VAE to be fixed before training begins.

These results make rendering part of the experimental specification of Cell Painting generation. A rendering should be reported with every FID and held fixed when generators are compared, and it should be checked against native targets on the data at hand before a generator is trained on it. Renderings, VAEs, and checkpoints are better selected by condition-level measures on validation data, since fidelity scores can favor weaker candidates and evaluating only the selected model cannot reveal such a mismatch. The choice between the two routes then follows from whether the rendering is settled: RGB latents repay a validated rendering with speed, and native pixels keep the rendering open while it is not. We release implementations of all seven protocols and the evaluation suite to make these checks routine. The step from stains to RGB belongs to both the model and the metric, and it deserves the same scrutiny as each.

AI USE STATEMENT

We used generative AI tools to provide feedback on research methodology and experimental design. An LLM-based feedback tool made available through ICLR commented on our experimental methodology, including the use of a single training seed per configuration. We also used generative AI tools to draft and revise portions of the manuscript, including Appendix C, identify related work missing from earlier drafts.

We reviewed the AI-assisted contributions. We take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies biological image generation through computational experiments on existing publicly available Cell Painting datasets, including RxRx1 HUVEC images and the U2OS crops released with IMPA. Although these datasets contain images of human-derived cells, our study involves no new wet-lab experiments, biological sample collection, or studies involving human participants. We do not collect new personal or sensitive data or attempt to identify individuals. The datasets are used for research purposes in accordance with their applicable licenses and data-use conditions. Generated images are used for computational evaluation and visualization and should not be interpreted as new experimental observations or validated wet-lab biological evidence.

REPRODUCIBILITY STATEMENT

We provide supplementary code for the seven RGB rendering protocols, core evaluation metrics, pixel- and latent-space model definitions, and feature-readout probes. The accompanying requirements file and README specify the software dependencies and usage. The appendix describes data preparation, evaluation splits, and experimental settings. Datasets and pretrained model weights should be obtained from their original releases. Anyone wishing to reproduce our results must comply with the corresponding licenses, access conditions, and release policies, including any restrictions on use or redistribution, and use these resources responsibly. We estimate the total cost of the CPGENeval experimental campaign at approximately 2,000 GPU-hours, 9,000 allocated CPU core-hours, and 10 TB of storage, covering data preparation, rendering, VAE reconstruction, generator training, sampling, and evaluation. GPU-hours sum device time across H100 and L40S without hardware normalization; CPU core-hours include cores allocated alongside GPU jobs.

REFERENCES

Black Forest Labs. Announcing Black Forest Labs, 2024. URL https://bfl.ai/blog/24-08-01-bfl.

Anis Bourou, Thomas Boyer, Marzieh Gheisari, Kévin Daupin, Véronique Dubreuil, Aurélie De Thonel, Valérie Mezger, and Auguste Genovesio. PhenDiff: Revealing subtle phenotypes with diffusion models in real images. In *MICCAI*, 2024.

Mark-Anthony Bray, Sigrun M Gustafsdottir, Mohammad H Rohban, Shantanu Singh, Vebjorn Ljosa, Katherine L Sokolnicki, Joshua A Bittker, Nicole E Bodycombe, Vlado Dančík, Thomas P Hasaka, et al. A dataset of images and morphological profiles of 30 000 small-molecule treatments using the Cell Painting assay. *GigaScience*, 2017.

Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David A Guertin, Joo Han Chang, Robert A Lindquist, Jason Moffat, et al. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 2006.

Srinivas Niranj Chandrasekaran, Jeanelle Ackerman, Eric Alix, D Michael Ando, John Arevalo, Melissa Bennion, Nicolas Boisseau, Adriana Borowa, Justin D Boyd, Laurent Brino, et al. JUMP Cell Painting dataset: morphological impact of 136,000 chemical and genetic perturbations. *bioRxiv*, 2023.

Srinivas Niranj Chandrasekaran, Beth A Cimini, Amy Goodale, Lisa Miller, Maria Kost-Alimova, Nasim Jamali, John G Doench, Briana Fritchman, Adam Skepner, Michelle Melanson, et al. Three million images and morphological profiles of cells treated with matched chemical and genetic perturbations. *Nature Methods*, 2024.

Min Jin Chong and David A. Forsyth. Effectively unbiased FID and inception score and where to find them. In *CVPR*, 2020.

Télio Cropsal and Rocío Mercado. Compressing biology: Evaluating the Stable Diffusion VAE for phenotypic drug discovery. *arXiv preprint arXiv:2510.19887*, 2025.

Berker Demirel, Marco Fumero, Theofanis Karaletsos, and Francesco Locatello. MorphGen: Controllable and morphologically plausible generative cell-imaging. *arXiv preprint arXiv:2510.01298*, 2025.

Terrance DeVries, Adriana Romero, Luis Pineda, Graham W Taylor, and Michal Drozdzal. On the evaluation of conditional GANs. *arXiv preprint arXiv:1907.08175*, 2019.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *NeurIPS*, 2017.

Charles Jones, Emmanuel Noutahi, Jason Hartford, and Cian Eastwood. Elucidating the design space of flow matching for cellular microscopy. *arXiv preprint arXiv:2603.26790*, 2026.

Alexandr A Kalinin, John Arevalo, Erik Serrano, Loan Vulliard, Hillary Tsang, Michael Bornholdt, Alán F Muñoz, Suganya Sivagurunathan, Bartek Rajwa, Anne E Carpenter, et al. A versatile information retrieval framework for evaluating profile strength and similarity. *Nature Communications*, 2025.

Oren Kraus, Federico Comitani, John Urbanik, Kian Kenyon-Dean, Lakshmanan Arumugam, Saber Saberian, Cas Wognum, Safiye Celik, and Imran S Haque. RxRx3-core: benchmarking drug-target interactions in high-content microscopy. *arXiv preprint arXiv:2503.20158*, 2025.

Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of ImageNet classes in Fréchet Inception Distance. In *ICLR*, 2023.

Alexis Lamiable, Tiphaine Champetier, Francesco Leonardi, Ethan Cohen, Peter Sommer, David Hardy, Nicolas Argy, Achille Massougbodji, Elaine Del Nery, Gilles Cottrell, et al. Revealing invisible cell phenotypes with conditional generative modeling. *Nature Communications*, 14(1): 6386, 2023.

Vebjorn Ljosa, Katherine L Sokolnicki, and Anne E Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature Methods*, 2012.

Mingyu Lu, Ethan Weinberger, Chanwoo Kim, and Su-In Lee. CellCLIP: Learning perturbation effects in Cell Painting via text-guided contrastive learning. *NeurIPS*, 2025.

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. In *CVPR*, 2026.

Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *ICLR*, 2018.

Zeinab Navidi, Jun Ma, Esteban Miglietta, Le Liu, Anne E. Carpenter, Beth A. Cimini, Benjamin Haibe-Kains, and Bo Wang. MorphoDiff: Cellular morphology painting with diffusion models. In *ICLR*, 2025.

Alessandro Palma, Fabian J Theis, and Mohammad Lotfollahi. Predicting cell morphological responses to perturbations using generative modeling. *Nature Communications*, 16(1):505, 2025.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *CVPR*, 2022.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024.

Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *NeurIPS*, 2019.

Mohammad Hossein Rohban, Shantanu Singh, Xiaoyun Wu, Julia B Berthet, Mark-Anthony Bray, Yashaswi Shrestha, Xaralabos Varelas, Jesse S Boehm, and Anne E Carpenter. Systematic morphological profiling of human gene and allele function via Cell Painting. *eLife*, 2017.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Srijit Seal, Maria-Anna Trapotsi, Ola Spjuth, Shantanu Singh, Jordi Carreras-Puigvert, Nigel Greene, Andreas Bender, and Anne E Carpenter. Cell Painting: a decade of discovery and innovation in cellular imaging. *Nature Methods*, 2025.

Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *ECCV*, 2018.

Stability AI. SD-VAE-ft-MSE, 2022. URL `https://huggingface.co/stabilityai/sd-vae-ft-mse`.

Stability AI. Introducing Stable Diffusion 3.5, 2024. URL `https://stability.ai/news-updates/introducing-stable-diffusion-3-5`.

George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *NeurIPS*, volume 36, 2023.

Maciej Sypetkowski, Morteza Rezanejad, Saber Saberian, Oren Kraus, John Urbanik, James Taylor, Ben Mabey, Mason Victors, Jason Yosinski, Alborz Rezazadeh Sereshkeh, et al. RxRx1: A dataset for evaluating experimental batch correction methods. In *CVPRW*, 2023.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Xuesong Wang, Yimin Fan, Yucheng Guo, Chenghao Fu, Kinhei Lee, Khachatur Dallakyan, Yaxuan Li, Qijin Yin, Yu Li, and Le Song. Prediction of cellular morphology changes under perturbations with a transcriptome-guided diffusion model. *Nature Communications*, 2025.

Xurui Wang, Qin Ren, Jun Ma, Haibin Ling, and Chenyu You. FreeBridge: Variational Schrödinger bridges for cellular transition dynamics. *arXiv preprint arXiv:2606.11286*, 2026.

Gregory P. Way, Ted Natoli, Adeniyi Adeboye, Lev Litichevskiy, Andrew Yang, Xiaodong Lu, Juan C. Caicedo, Beth A. Cimini, Kyle Karhohs, David J. Logan, Mohammad H. Rohban, Maria Kost-Alimova, Kate Hartland, Michael Bornholdt, Srinivas Niranj Chandrasekaran, Marzieh Haghighi, Erin Weisbart, Shantanu Singh, Aravind Subramanian, and Anne E. Carpenter. Morphology and gene expression profiling provide complementary information for mapping cell state. *Cell Systems*, 2022.

Dongxia Wu, Shiye Su, Yuhui Zhang, Elaine Sui, Emma Lundberg, Emily B Fox, and Serena Yeung-Levy. CellFluxRL: Biologically-constrained virtual cell modeling via reinforcement learning. *arXiv preprint arXiv:2603.21743*, 2026.

Yuhui Zhang, Yuchang Su, Chenyu Wang, Tianhong Li, Zoe Wefers, Jeffrey J. Nirschl, James Burgess, Daisy Ding, Alejandro Lozano, Emma Lundberg, and Serena Yeung-Levy. CellFlux: Simulating cellular morphology changes via flow matching. In *ICML*, 2025.

Yuhui Zhang, Yuchang Su, Zoe Wefers, Shiye Su, He Li, Tianhong Li, Chenyu Wang, James Burgess, Alejandro Lozano, Linqi Zhou, et al. CellFluxV2: An image generative foundation model for virtual cell modeling. *bioRxiv*, 2026.

A VISUALIZATION

Rendering protocols. Figures 5–8 compare seven rendering protocols on two HUVEC512 sites and two IMPA96 crops, alongside their stain selection, intensity calibration, and RGB mixing. For each dataset, we use the first and last entries of a ten-image set sampled from its treatment training split for the protocol visualizations; neither image intensities nor evaluation scores were used for selection. The six native channels use a fixed grayscale display range of 0–255, and the RGB outputs are shown without additional brightness or contrast adjustment. CELLFLUX produces near-black regions on HUVEC512, whereas RECURSION produces large near-white regions on IMPA96. These comparisons transfer the mappings across input domains: the published RxRx1 use of CELLFLUX relies on IMPA-preprocessed U2OS crops (Zhang et al., 2025; Palma et al., 2025), while RECURSION uses the fixed gains supplied for raw sites in the RxRx1 visualization code.[1] Under these transfers, the fixed offset and clipping in CELLFLUX suppress low raw intensities, and the fixed gains and clipping in RECURSION produce near-white regions in bright preprocessed inputs. Appendix Table 3 reports the dataset-level near-black and near-white pixel fractions over the full HUVEC512 and IMPA96 training sets, including controls.

Corresponding RxRx1 inputs. Figure 9 pairs the two IMPA96 crops above with their raw RxRx1 U2OS sites, matched by experiment, plate, well, and site identifiers. This comparison keeps the cell type and acquisition fixed. In both examples, CELLFLUX shows cellular structure in the released crops and produces near-black renderings of the raw sites. RECURSION shows cellular structure in the raw sites and produces large near-white regions in the crops. The full 512×512 sites and 96×96 crops differ in spatial extent as well as preprocessing; the released metadata inspected here does not provide crop coordinates, so this comparison does not isolate preprocessing alone.

VAE reconstruction. Figures 10–13 compare each protocol's rendered input with reconstructions from five frozen VAEs using the posterior mode. During the paired reconstruction evaluation, we sampled 16 images per dataset from the training split (seed 23) and saved their rendered inputs and VAE reconstructions for visualization. From this pool, we select two treatment images per dataset using seed 20260923, requiring different perturbations and batches within each dataset; selection uses metadata only. These inputs differ from those in the rendering figures. Within each row, all five VAEs reconstruct the same rendered input, and every panel retains its full 512×512 site or 96×96 crop. The panels display the 8-bit RGB arrays supplied as input or saved after reconstruction, without additional brightness or contrast adjustment. This pairing separates differences already present in the rendered input from changes introduced by VAE reconstruction.

Generation. Figure 14 compares held-out HUVEC512 images with outputs from the pixel-space model and five RGB latent generators at 100k updates. We show four siRNA conditions in the CELLCLIP RGB space. Real and generated examples share the same perturbation label but have no spatial correspondence.

[1] https://github.com/recursionpharma/rxrx1-utils

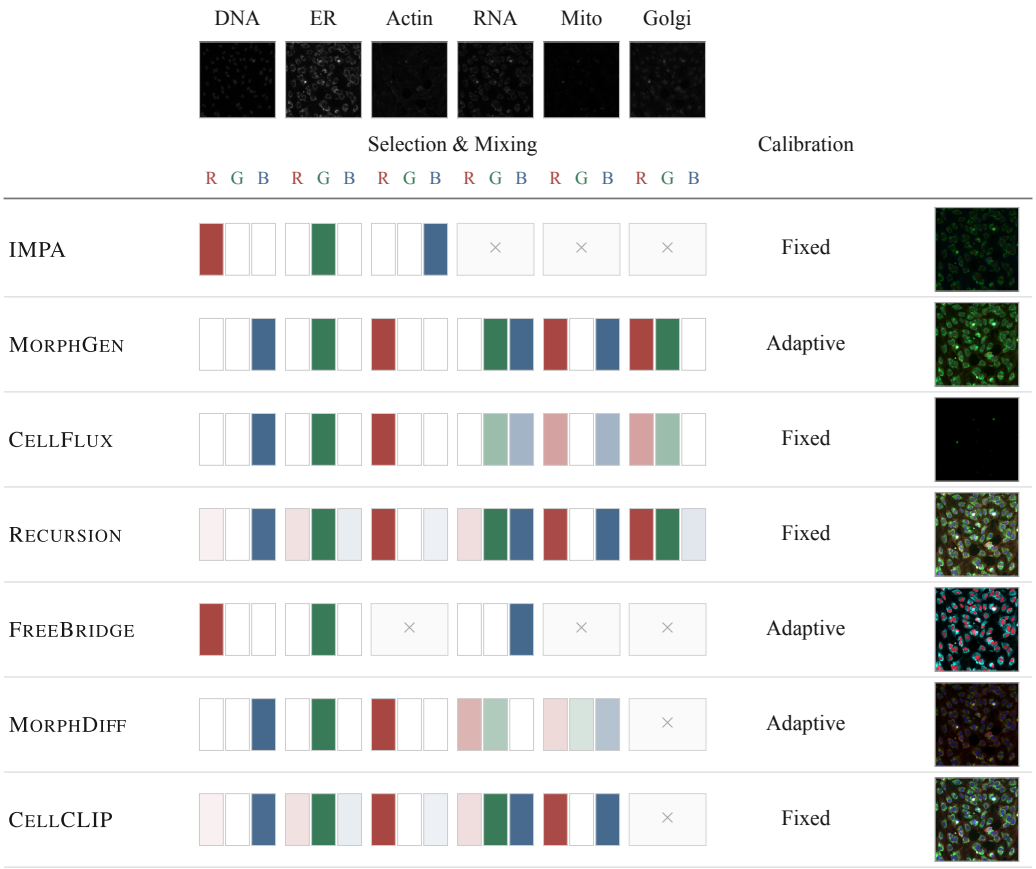


Figure 5: Rendering protocols on HUVEC512. The six stains (top) and seven RGB outputs (right) come from the same 512 × 512 treated training site, HUVEC-01_1_D21_1 (siRNA s29185). Selection: × marks an omitted stain. Calibration: Fixed and Adaptive denote preset and image-dependent intensity scaling. Mixing: the colored blocks show RGB coefficients relative to the largest coefficient within each protocol; white indicates zero. Gains, offsets and clipping are not encoded by the blocks. Native stains use a fixed 0–255 display range; RGB outputs receive no additional contrast adjustment.

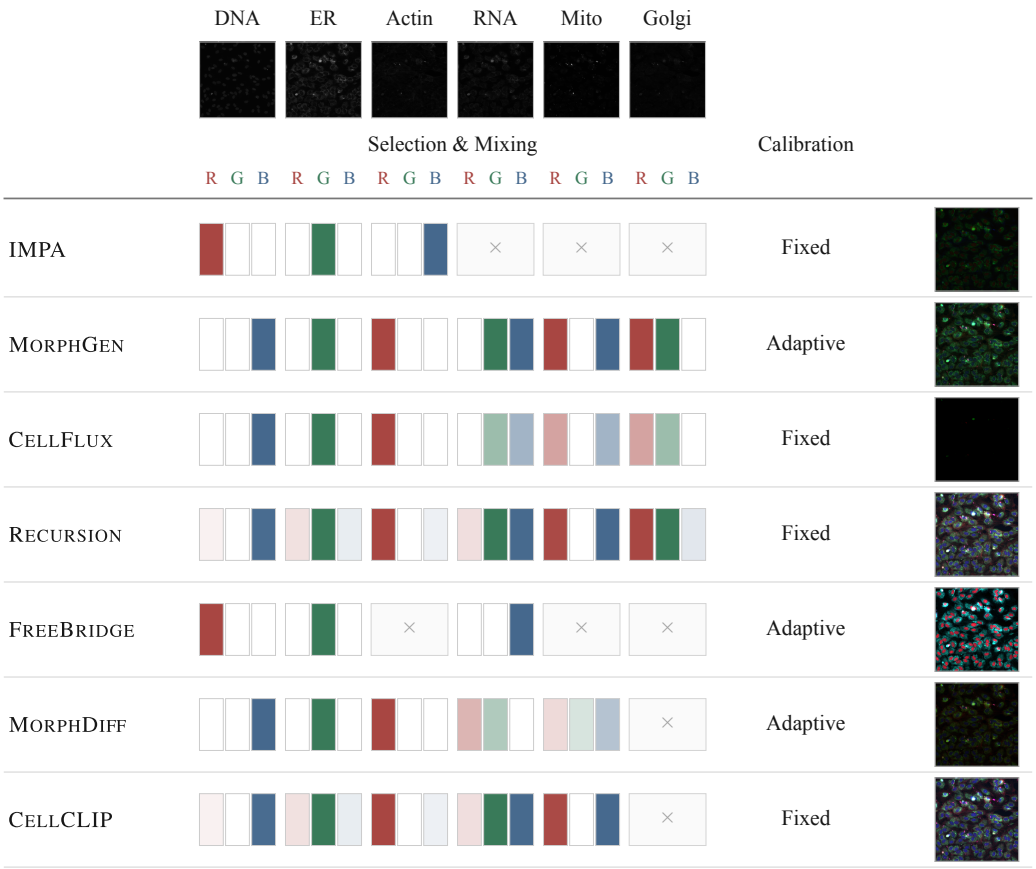


Figure 6: Rendering protocols on HUVEC512. The six stains (top) and seven RGB outputs (right) come from the same 512 × 512 treated training site, HUVEC-16_1_K10_1 (siRNA s27192). Selection: × marks an omitted stain. Calibration: Fixed and Adaptive denote preset and image-dependent intensity scaling. Mixing: the colored blocks show RGB coefficients relative to the largest coefficient within each protocol; white indicates zero. Gains, offsets and clipping are not encoded by the blocks. Native stains use a fixed 0–255 display range; RGB outputs receive no additional contrast adjustment.



Figure 7: Rendering protocols on IMPA96. The six stains (top) and seven RGB outputs (right) come from the same 96 × 96 treated training crop, U2OS-01_1_F08_s1_59 (gene label KIAA0100). Selection: × marks an omitted stain. Calibration: Fixed and Adaptive denote preset and image-dependent intensity scaling. Mixing: the colored blocks show RGB coefficients relative to the largest coefficient within each protocol; white indicates zero. Gains, offsets and clipping are not encoded by the blocks. Native stains use a fixed 0–255 display range; RGB outputs receive no additional contrast adjustment.



Figure 8: Rendering protocols on IMPA96. The six stains (top) and seven RGB outputs (right) come from the same 96 × 96 treated training crop, U2OS-03_4_J05_s1_56 (gene label ALDH1L1). Selection: × marks an omitted stain. Calibration: Fixed and Adaptive denote preset and image-dependent intensity scaling. Mixing: the colored blocks show RGB coefficients relative to the largest coefficient within each protocol; white indicates zero. Gains, offsets and clipping are not encoded by the blocks. Native stains use a fixed 0–255 display range; RGB outputs receive no additional contrast adjustment.

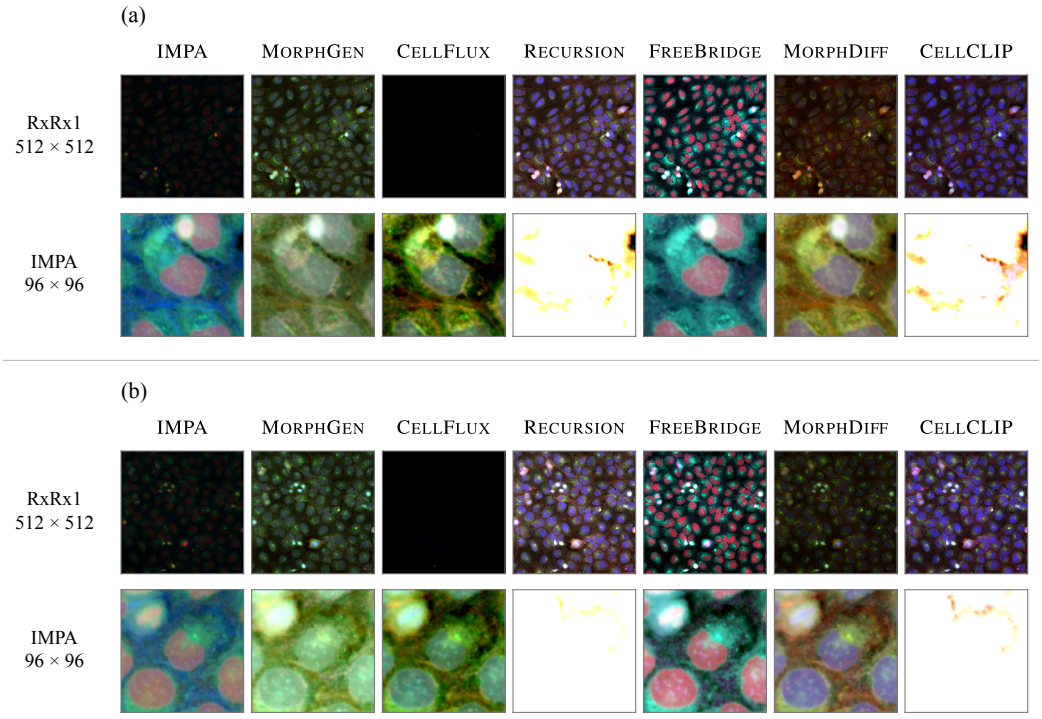


Figure 9: Seven rendering protocols applied to raw RxRx1 U2OS sites and their corresponding IMPA96 crops. Within each pair, the two releases share the experiment, plate, well, and site identifiers. (a) Raw site `U2OS-01_1_F08_1` and crop `U2OS-01_1_F08_s1_59` (target gene KIAA0100). (b) Raw site `U2OS-03_4_J05_1` and crop `U2OS-03_4_J05_s1_56` (target gene ALDH1L1). Each panel shows the complete 512×512 site or 96×96 released crop, without additional brightness or contrast adjustment. This comparison holds the acquisition and cell type fixed, but both preprocessing and spatial extent differ; the crop locations within the sites have not been registered.

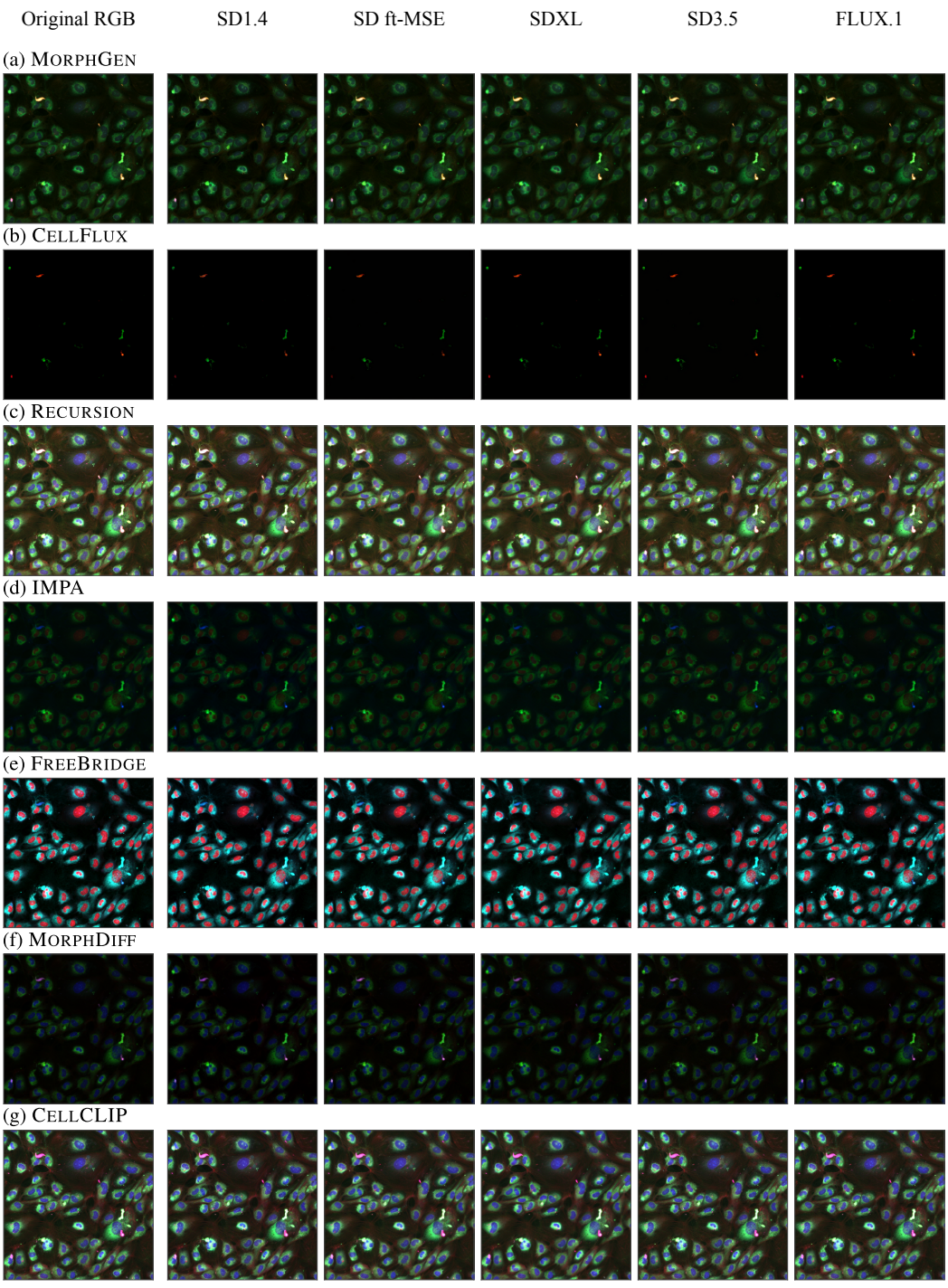


Figure 10: VAE reconstruction of one HUVEC512 treatment image (siRNA s21486; `HUVEC-02_4_E16_1`). Each subfigure applies one rendering protocol to the same source image and shows its original RGB representation followed by five frozen VAE reconstructions. All panels retain the complete 512×512 field and the saved RGB intensities, without additional contrast normalization. Reconstructions use the posterior mode.

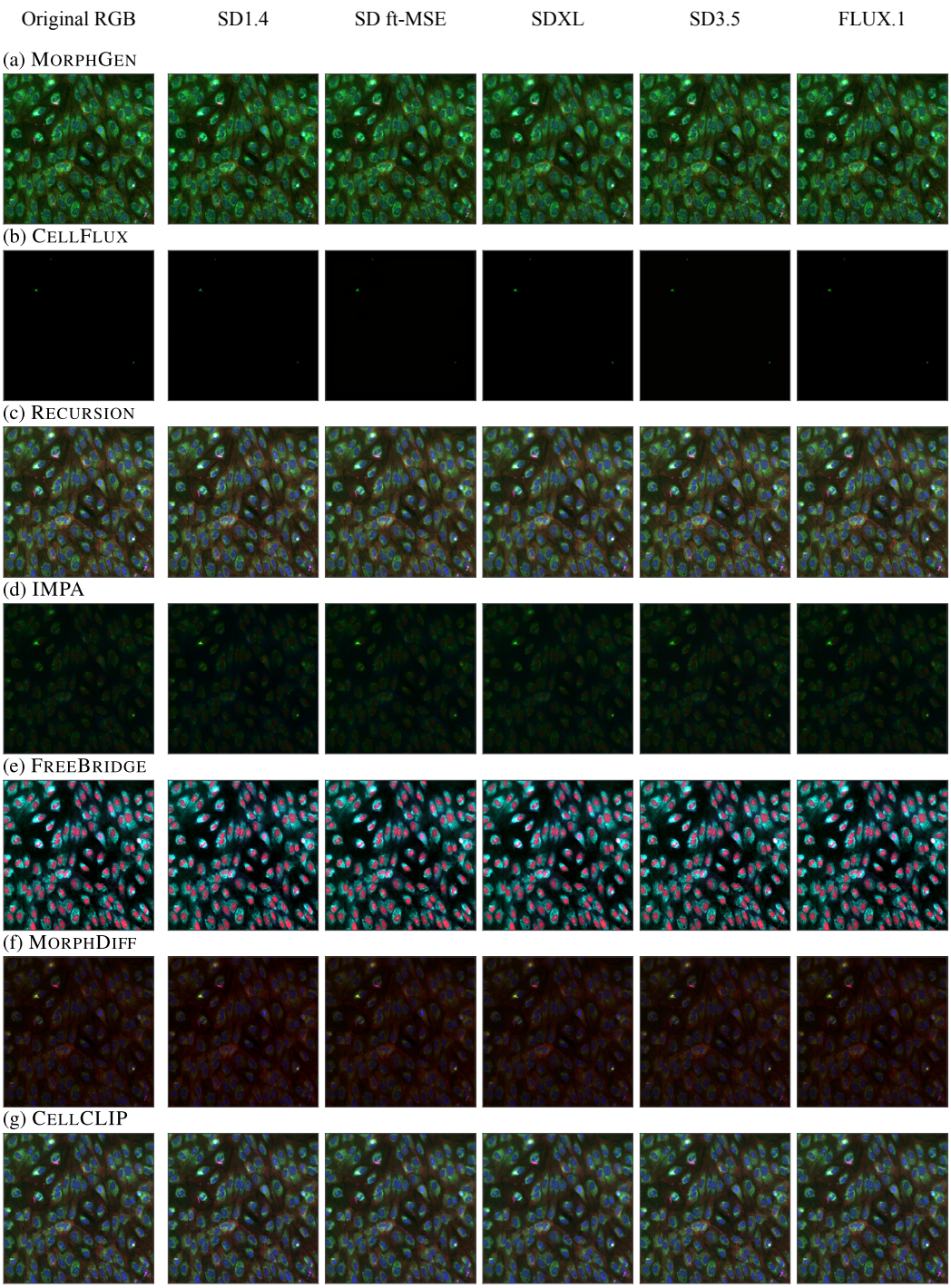


Figure 11: VAE reconstruction of one HUVEC512 treatment image (siRNA s19391; `HUVEC-11_1_O11_1`). Each subfigure applies one rendering protocol to the same source image and shows its original RGB representation followed by five frozen VAE reconstructions. All panels retain the complete 512×512 field and the saved RGB intensities, without additional contrast normalization. Reconstructions use the posterior mode.

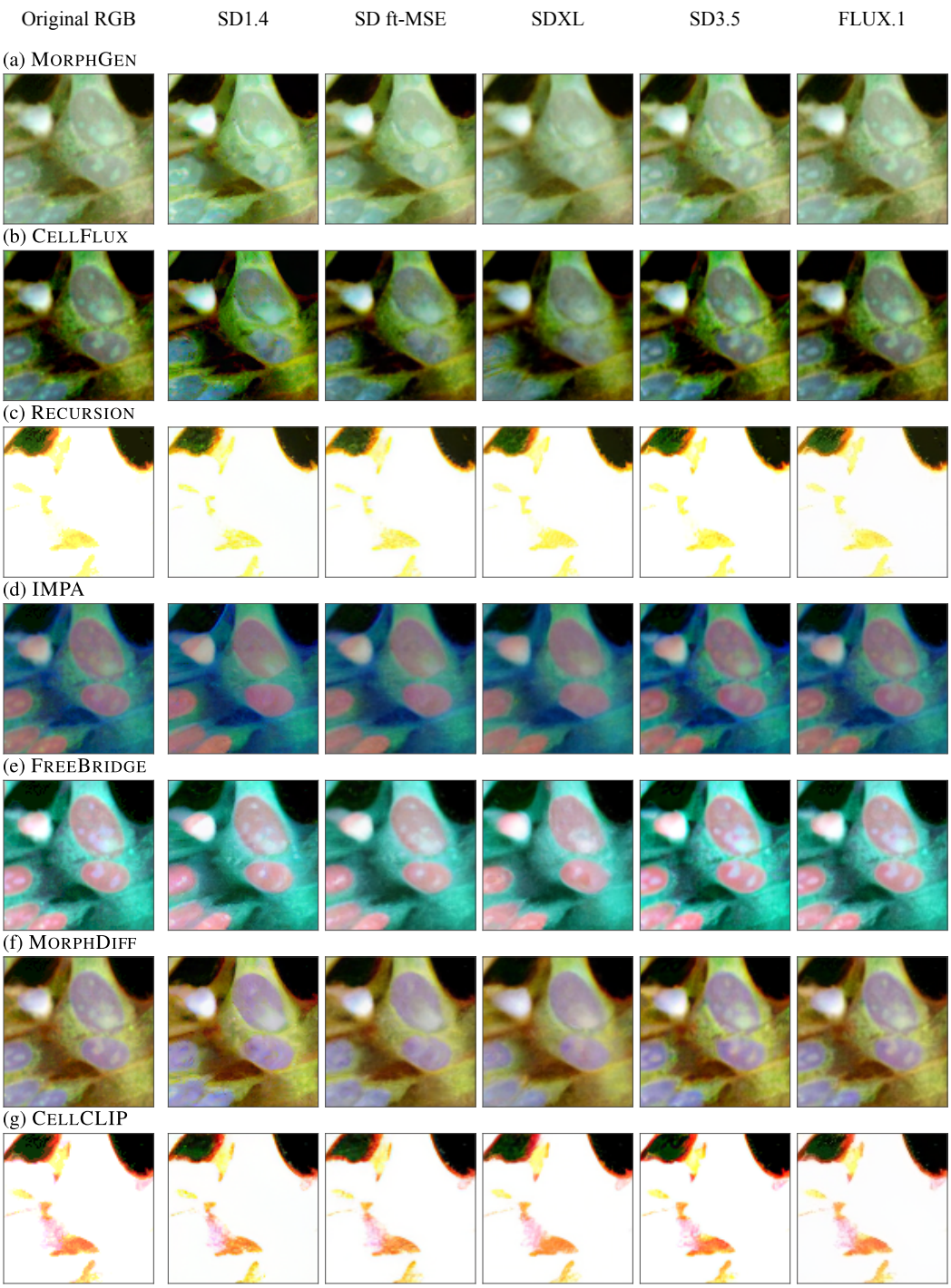


Figure 12: VAE reconstruction of one IMPA96 treatment image (target SDS; `U2OS-01_2_J08_s1_26`). Each subfigure applies one rendering protocol to the same source image and shows its original RGB representation followed by five frozen VAE reconstructions. All panels retain the complete 96×96 field and the saved RGB intensities, without additional contrast normalization. Reconstructions use the posterior mode.

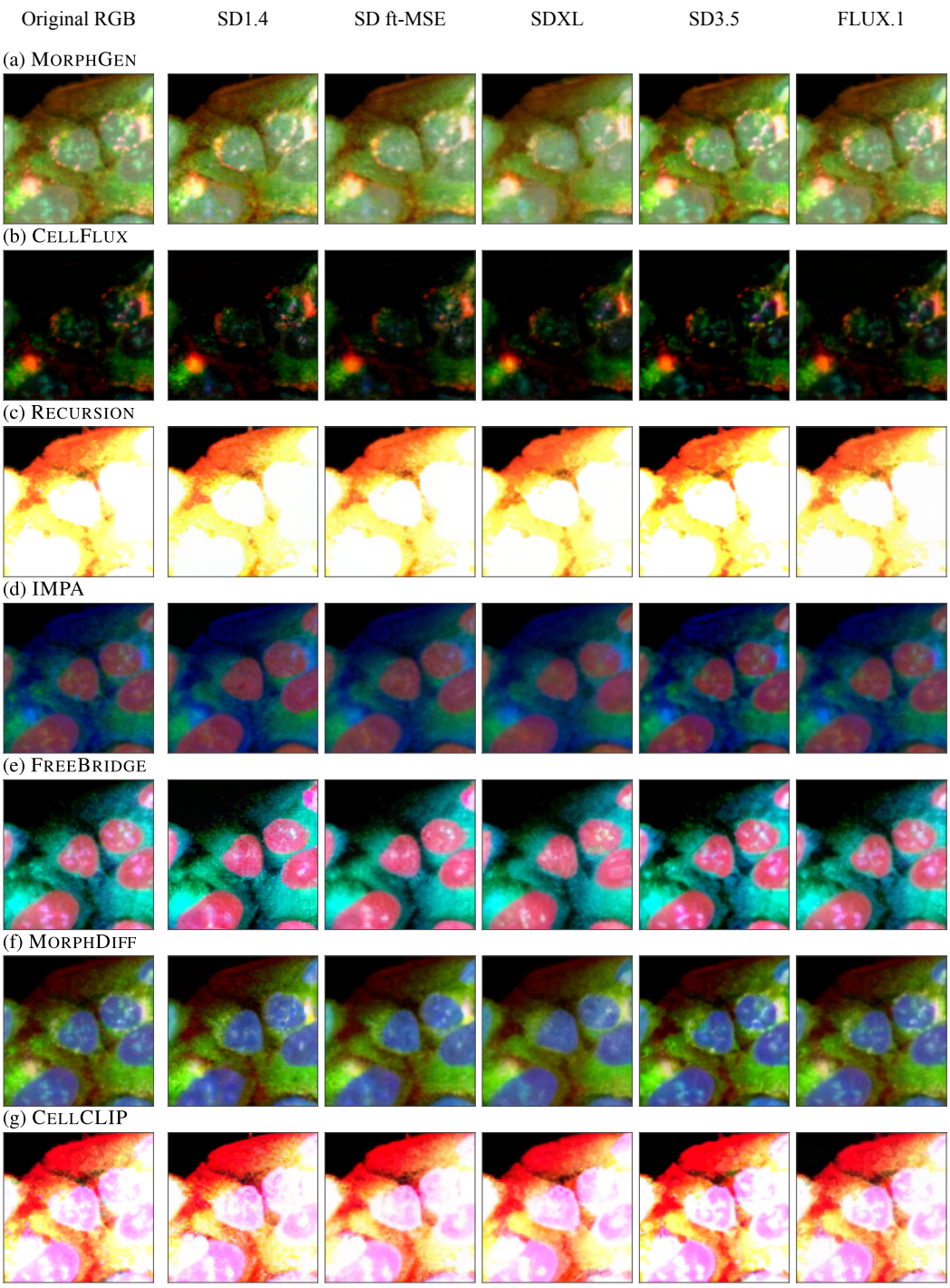
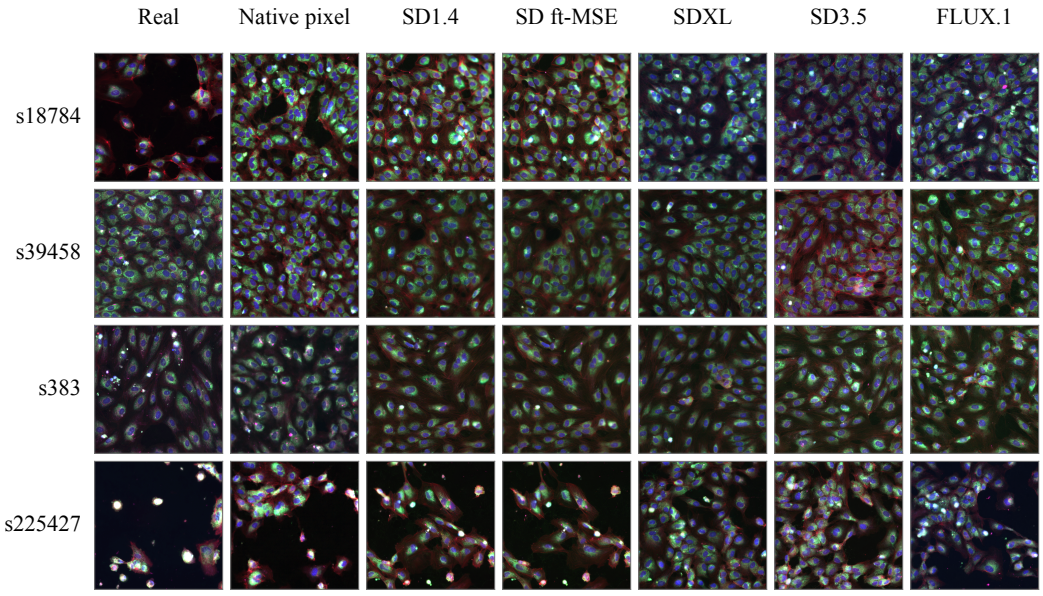


Figure 13: VAE reconstruction of one IMPA96 treatment image (target NPRL2; `U2OS-02_4_E17_s1_41`). Each subfigure applies one rendering protocol to the same source image and shows its original RGB representation followed by five frozen VAE reconstructions. All panels retain the complete 96×96 field and the saved RGB intensities, without additional contrast normalization. Reconstructions use the posterior mode.



(a) CELLCLIP

Figure 14: Visualization of HUVEC512 generation under (a) CELLCLIP. Rows show the same four siRNA conditions. Columns show real held-out images, native pixel generation, and five RGB latent generators at 100k steps. Real and native pixel images use the indicated rendering protocol; latent models are trained with it. Real and generated images are matched by condition, without spatial correspondence.

B RENDERING PROTOCOLS

We apply seven source-derived rendering mappings to the same six-channel, 8-bit inputs: HUVEC512 sites and IMPA96 crops. The stain order is $(1, \ldots, 6) = (\mathrm{DNA}, \mathrm{ER}, \mathrm{Actin}, \mathrm{RNA}, \mathrm{Mito}, \mathrm{Golgi})$. For a pixel, $x = (x_1, \ldots, x_6)^\top$ denotes its stain intensities and $y = (y_R, y_G, y_B)^\top$ its rendered values. Each mixing matrix below has one row per retained stain and one column per output channel. Table 2 summarizes the operations; the protocol descriptions give their original roles, numerical parameters, and adaptations to our inputs.

Table 2: Rendering operations used in the benchmark. Stain numbers follow the six-channel order defined in the text; Q_p denotes the pth percentile. The BGR column indicates a final reversal from BGR to RGB. The matrices and the exact order of clipping and quantization are specified below.

Protocol	Stains used	Pre-mix window	Mixing	Post-mix transform	Quantization	BGR
RECURSION	1–6	Fixed $[0, h_k]$	C_{R}	Clip sum to $[0, 255]$	Floor each stain contribution	No
CELLFLUX	1–6	$2x/255 - 1$	W_{F}	Clip to $[-1, 1]$; map to $[0, 255]$	Floor after mixing	No
MORPHGEN	1–6	None	$W_{\mathrm{G}}/3$	Shared RGB $Q_{0.1}$–$Q_{99.9}$ stretch	Floor after stretch	No
IMPA	1, 2, 3	None	Stack	None	Preserve input bytes	No
FREEBRIDGE	1, 2, 4	Per-stain Q_1–Q_{99}	Stack	None	Nearest, ties up	No
MORPHDIFF	1–5	None	C_{D}	Per-output min–max; multiply by 256	Nearest, ties to even; clip	Yes
CELLCLIP	1–5	Fixed $[0, h_k]$	$C_{\mathrm{R},1:5}$	Clip sum to $[0, 255]$	Floor each stain contribution	No

The mappings operate at the supplied spatial resolution and return 8-bit RGB images. IMPA96 inputs already contain the preprocessing of the released crop dataset; rendering does not repeat that preprocessing. Adaptive bounds are estimated separately for each supplied site or crop, using NumPy's default linear interpolation for percentiles. Intensity and mixing arrays use float64 for RECURSION, CELLCLIP, and MORPHDIFF, and float32 for CELLFLUX, MORPHGEN, and FREEBRIDGE; IMPA copies the selected bytes. The MORPHGEN and MORPHDIFF stretches assume nonzero dynamic range; the experimental renderer applies their divisions without an explicit zero-range branch. The recipe diagrams in Figures 5–8 show stain selection and relative mixing weights; the definitions below also specify gains, offsets, clipping, and quantization.

For each protocol, *Under transfer* reports the two evaluated input settings, including previously reported pairings and transfers. Near-white pixels have all three RGB values at least 250, and near-black pixels have all three values at most 5. We report image-wise mean fractions over the full training sets, including controls: 39,342 HUVEC512 sites and 136,673 IMPA96 crops. Pairs below give (near-white, near-black) percentages, rounded to two decimal places; a value of 0.00% can reflect rounding. Table 3 gives the corresponding means and sample standard deviations.

RECURSION. *Source.* The RxRx1 visualization utility,[2] also adopted by *MorphoDiff* (Navidi et al., 2025) to prepare RGB images for training and FID evaluation. *Original input.* Raw six-channel RxRx1 sites. *Selection.* All six stains. *Calibration.* Fixed per-stain upper bounds $h = (51, 107, 64, 191, 89, 191)$ give $\bar{x}_k = \min(x_k/h_k, 1)$. *Mixing.* Each calibrated stain contributes one row of the integer color table

$$C_{\mathrm{R}} = \begin{bmatrix} 19 & 0 & 249 \\ 42 & 255 & 31 \\ 255 & 0 & 25 \\ 45 & 255 & 252 \\ 250 & 0 & 253 \\ 254 & 255 & 40 \end{bmatrix}. \tag{3}$$

[2] `rxrx1-utils/rxrx/io.py`, commit `31e6f9b`.

Post-processing. Truncate each nonnegative stain contribution before accumulation, then clip the sum:

$$y_j = \mathrm{clip}_{[0,255]}\left(\sum_{k=1}^{6} \lfloor \bar{x}_k C_{\mathrm{R},kj} \rfloor\right), \qquad j \in \{R, G, B\}. \tag{4}$$

The output is stored as uint8 RGB without channel reversal; clipping and quantization make the complete map nonlinear. *Six-channel adaptation.* Stain order, bounds, and colors are retained. We replace the source's fixed 512×512 reshape with the supplied image dimensions, allowing the same pixelwise map to process IMPA96 crops. *Under transfer.* The mean fractions are $(0.13\%, 0.00\%)$ on HUVEC512 and $(63.68\%, 1.32\%)$ on IMPA96 (Table 3). The fixed bounds and clipped sum saturate bright inputs in the preprocessed crop domain.

CELLFLUX. *Source.* The six-to-three-channel converter used for FID evaluation in *CellFlux* (Zhang et al., 2025).[3] *Original input.* IMPA-preprocessed six-channel RxRx1 U2OS crops, normalized to the model's $[-1, 1]$ intensity convention before RGB conversion. *Selection.* All six stains. *Calibration.* The fixed affine transform is $u = 2x/255 - \mathbf{1}$, without image-dependent bounds. *Mixing.* Apply $m = W_{\mathrm{F}}^{\top} u$, with

$$W_{\mathrm{F}} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{bmatrix}. \tag{5}$$

Post-processing. Clip each mixed channel to $[-1, 1]$, map it to $[0, 255]$, and take the floor:

$$y = \left\lfloor 255 \left(\tfrac{1}{2} \mathrm{clip}_{[-1,1]}(m) + \tfrac{1}{2} \right) \right\rfloor. \tag{6}$$

The output is uint8 RGB without channel reversal. Each output column of W_{F} sums to 2. Ignoring floating-point rounding, the same expression is $y_j = \lfloor \mathrm{clip}_{[0,255]}(\sum_k W_{\mathrm{F},kj} x_k - 127.5) \rfloor$; weighted sums at or below 127.5 therefore map to zero. *Six-channel adaptation.* We retain the six-channel order and fixed affine transform, and omit the source loader's stochastic $U[0, 1)$ dequantization noise to obtain a deterministic mapping of the supplied bytes. *Under transfer.* The mean fractions are $(0.00\%, 99.63\%)$ on HUVEC512 and $(0.10\%, 22.38\%)$ on IMPA96 (Table 3). The fixed offset suppresses low raw intensities, explaining the large near-black regions under transfer to HUVEC512.

MORPHGEN. *Source.* The RGB conversion used for visualization and FID evaluation in *MorphGen* (Demirel et al., 2025).[4] *Original input.* Multichannel Cell Painting images, including six-channel RxRx1 sites; the evaluation code invokes the converter separately for each image. *Selection.* All six stains. *Calibration.* Estimate one shared 0.1st–99.9th percentile window from the three mixed output channels of each image. *Mixing.* Compute $m = W_{\mathrm{G}}^{\top} x/3$ using

$$W_{\mathrm{G}} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \tag{7}$$

Post-processing. Set $a = m/255$, flatten the full mixed image in channel–height–width order, and sample every 100th value starting at index zero. Let ℓ and h be the 0.001 and 0.999 quantiles of that sample. For $h > \ell$, return

$$y = \left\lfloor 255 \, \frac{\mathrm{clip}_{[\ell,h]}(a) - \ell}{h - \ell} \right\rfloor \tag{8}$$

as uint8 RGB without channel reversal. Thus the window is shared across mixed RGB channels, whereas FREEBRIDGE stretches selected input stains independently before stacking. *Six-channel*

[3] `CellFlux/training/data_utils.py` and `eval_fid.py`, commit `041c4c0`.

[4] `MorphGen/sc_perturb/dataset.py`, functions `to_rgb` and `rescale_intensity`, commit `cc8ee42`.

adaptation. We retain all six channel positions and estimate bounds on the supplied site or crop. The benchmark computes quantiles with NumPy; the source uses PyTorch. *Under transfer.* The mean fractions are $(0.00\%, 1.69\%)$ on HUVEC512 and $(0.06\%, 1.64\%)$ on IMPA96 (Table 3).

IMPA. *Source.* The default three-channel selection in the FID evaluation of *IMPA* (Palma et al., 2025).[5] *Original input.* IMPA-preprocessed U2OS 96×96 crops, read as normalized six-channel floating-point images for evaluation. *Selection.* Stains 1, 2, and 3: DNA, ER, and Actin. *Calibration.* No additional intensity calibration at the RGB mapping stage. *Mixing.* Stack the selected planes directly as $y = (x_1, x_2, x_3)^\top$. *Post-processing.* Preserve the input uint8 values, with DNA mapped to red, ER to green, and Actin to blue; there is no clipping, requantization, or channel reversal. *Six-channel adaptation.* We retain the source's default channel indices on both datasets and omit RNA, Mito, and Golgi. The benchmark preserves the selected bytes and omits the source loader's stochastic dequantization; the source FID path consumes normalized floats. *Under transfer.* The mean fractions are $(0.00\%, 12.48\%)$ on HUVEC512 and $(0.00\%, 3.80\%)$ on IMPA96 (Table 3).

FREEBRIDGE. *Source.* The image-preparation script released with *FreeBridge* (Wang et al., 2026), which produces RGB single-cell crops for its model pipeline.[6] *Original input.* Full-field channel TIFFs in the released BBBC021 preparation script; it rescales each selected plane before extracting 96×96 crops. *Selection.* The source chooses the first available triple from the one-based priority list $(1, 2, 4)$, $(1, 2, 3)$, $(2, 3, 4)$, $(1, 3, 4)$. On our six-channel inputs, this selects DNA, ER, and RNA. *Calibration.* For each selected stain k, estimate $\ell_k = Q_1(x_k)$ and $h_k = Q_{99}(x_k)$ from all pixels of the supplied image and set $z_k = \mathrm{clip}_{[0,1]}((x_k - \ell_k)/(h_k - \ell_k))$. If $h_k \le \ell_k + 10^{-6}$, return a zero plane. *Mixing.* Stack the three independently calibrated planes in the order $(1, 2, 4)$, without cross-stain mixing. *Post-processing.* Quantize each plane as $\lfloor 255 z_k + 0.5 \rfloor$ and store uint8 RGB, giving nearest rounding with ties upward and no channel reversal. *Six-channel adaptation.* The stain selection follows the source's default priority without an override. Bounds are estimated over each HUVEC512 site or each already-preprocessed IMPA96 crop; the latter changes the support of the percentiles relative to stretching a full field before cropping. We retain the supplied spatial extent. *Under transfer.* The mean fractions are $(0.13\%, 8.17\%)$ on HUVEC512 and $(0.17\%, 8.19\%)$ on IMPA96 (Table 3).

MORPHDIFF. *Source.* The `composite` display function in *MorphDiff* (Wang et al., 2025), called after generating and saving five-channel images.[7] *Original input.* Five planes labeled in the source as `ERSyto`, `ERSytoBleed`, `Hoechst`, `Mito`, and `Ph_golgi`, in that order. *Selection.* Our implementation supplies the first five RxRx1 stains: DNA, ER, Actin, RNA, and Mito. *Calibration.* There is no pre-mix intensity calibration. Each mixed output plane is subsequently scaled using its own spatial minimum and maximum. *Mixing.* Compute $m = C_{\mathrm{D}}^\top x_{1:5}$ with the source's active palette,

$$C_{\mathrm{D}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0.4 & 0.4 \\ 0.4 & 0.2 & 0.2 \end{bmatrix}. \tag{9}$$

Its columns denote the working BGR planes used by the source's OpenCV image writer. *Post-processing.* For each working plane j, compute spatial extrema $\ell_j = \min m_j$ and $h_j = \max m_j$ and, for $h_j > \ell_j$, set $v_j = 256(m_j - \ell_j)/(h_j - \ell_j)$. Apply `np.rint` (nearest integer, ties to even), clip to $[0, 255]$, cast to uint8, and reverse BGR to RGB. The scale is 256, including clipping of the upper endpoint. Independent output scaling can change relative channel intensities and therefore color ratios between images. *Six-channel adaptation.* The five inputs are assigned by position, not matched to the source's stain names; the separate RxRx1 Golgi plane is omitted rather than combined with another stain. This instantiates the source display palette on our inputs. The recipe diagrams show the effective RGB coefficients after the final channel reversal. Because the source planes combine stains differently from RxRx1 (e.g., `Ph_golgi` combines actin, Golgi, and plasma membrane), no complete one-to-one semantic assignment exists; MORPHDIFF here therefore tests the source palette

[5] `IMPA/eval/eval.py` and `gan_metrics/fid.py`, commit `13eb82f`.

[6] `FreeBridge/data_prep/crop_bbbc021_singlecell.py`, commit `f285a22`.

[7] `MorphDiff/scripts/gene_img2img_5c.py`, active `drug` palette, commit `80f79ca`.

Table 3: Near-white and near-black pixel fractions across rendering protocols. Values are image-wise means ± sample standard deviations (%), over the full training sets including controls: 136,673 IMPA96 crops and 39,342 HUVEC512 sites. Near-white: all three RGB values ≥ 250; near-black: all three ≤ 5.

Protocol	**IMPA96**		**HUVEC512**	
	Near-white	Near-black	Near-white	Near-black
MORPHGEN	0.06 ± 0.24	1.64 ± 6.71	0.00 ± 0.01	1.69 ± 7.08
CELLFLUX	0.10 ± 0.43	22.38 ± 23.64	0.00 ± 0.00	99.63 ± 0.60
RECURSION	63.68 ± 25.55	1.32 ± 6.04	0.13 ± 0.19	0.00 ± 0.09
IMPA	0.00 ± 0.00	3.80 ± 10.89	0.00 ± 0.00	12.48 ± 19.34
FREEBRIDGE	0.17 ± 0.27	8.19 ± 13.59	0.13 ± 0.11	8.17 ± 10.47
MORPHDIFF	0.00 ± 0.01	4.07 ± 11.08	0.00 ± 0.00	12.52 ± 17.14
CELLCLIP	54.92 ± 30.07	3.45 ± 10.44	0.07 ± 0.13	0.16 ± 2.23

and its per-plane min–max stretch rather than its intended stain colors. *Under transfer.* The mean fractions are $(0.00\%, 12.52\%)$ on HUVEC512 and $(0.00\%, 4.07\%)$ on IMPA96 (Table 3).

CELLCLIP. *Source.* The RGB export utility released with the *CellCLIP* representation-learning codebase (Lu et al., 2025).[8] *Original input.* Preprocessed Cell Painting arrays; the released RxRx3-core example selects and reorders five planes before RGB export. *Selection.* DNA, ER, Actin, RNA, and Mito; Golgi is omitted. *Calibration.* Use the first five RECURSION bounds, $h_{1:5} = (51, 107, 64, 191, 89)$, with $\bar{x}_k = \min(x_k/h_k, 1)$. *Mixing.* Use the first five rows of C_{R} in Eq. 3. *Post-processing.* Truncate each stain contribution before summation and clip to $[0, 255]$, as in Eq. 4 with $k = 1, \ldots, 5$. Store uint8 RGB without channel reversal. *Six-channel adaptation.* Accounting for the source's one-based input permutation $(2, 4, 5, 1, 3)$ and correspondingly ordered color table gives the same map as RECURSION restricted to stains 1–5. We apply this RGB stage to the supplied bytes; the source's earlier dataset preprocessing is outside this mapping. *Under transfer.* The mean fractions are $(0.07\%, 0.16\%)$ on HUVEC512 and $(54.92\%, 3.45\%)$ on IMPA96 (Table 3).

[8] `CellCLIP/preprocessing/convert_rgb.py`, commit `95d909b`.

C Datasets and Additional Related Work

C.1 Datasets

Table 4: Datasets, image dimensions, and image counts. Panel (a) reports source datasets and their published training and evaluation splits; panel (b) reports surveyed fluorescence image releases. Counts are per image and include controls unless marked as treated.

(a) Datasets used in this study

Dataset	Image size	Channel labels	Channels	Train	Eval	Batches
RxRx1 (Sypetkowski et al., 2023)	512×512	DNA, ER, Actin, RNA, Mito, Golgi	6	81,224	44,286	51
RxRx1 (HUVEC), treated	512×512	DNA, ER, Actin, RNA, Mito, Golgi	6	35,376	17,692	24
RxRx1 (HEPG2), treated	512×512	DNA, ER, Actin, RNA, Mito, Golgi	6	15,500	8,858	11
RxRx1 (RPE), treated	512×512	DNA, ER, Actin, RNA, Mito, Golgi	6	15,506	8,834	11
RxRx1 (U2OS), treated	512×512	DNA, ER, Actin, RNA, Mito, Golgi	6	6,648	4,410	5
IMPA96 (Palma et al., 2025)	96×96	DNA, ER, Actin, RNA, Mito, Golgi	6	136,673	34,270	3
CellFlux eval. subset (Zhang et al., 2025)	96×96	DNA, ER, Actin, RNA, Mito, Golgi	6	12,210	3,062	3

(b) Broader dataset survey

Dataset	Image size	Channel labels	Channels	Total	Batches
BBBC021 (Ljosa et al., 2012)	1024×1280	DNA, Tubulin, Actin	3	13,200	10
JUMP pilot / cpg0000 (Chandrasekaran et al., 2024)	1080×1080	DNA, Mito, AGP, ER, RNA	5	340,466	6
Broad LINCS / cpg0004 (Way et al., 2022)	2160×2160	AGP, DNA, ER, Mito, RNA	5	487,253	2
Rohban / cpg0017 (Rohban et al., 2017)	1080×1080	DNA, Mito, AGP, ER, RNA	5	17,261	1
RxRx3-core (Kraus et al., 2025)	512×512	DNA, ER, Actin, RNA, Mito, Golgi	6	222,601	180
CDRP / Bray2017 (Bray et al., 2017)	520×696	DNA, ER, RNA, AGP, Mito	5	919,265*	–
Golgi scattering (Lamiable et al., 2023)	128×128	DNA, EGFP-CCR5	2	175,020†	–
Nuclear translocation (Lamiable et al., 2023)	256×256	DNA, NF-κB p65	2	15,999	–
LRRK2 (Lamiable et al., 2023)	256×256	DNA, TH, α-synuclein	3	71,520	–

Experiments Our experiments use RxRx1 (HUVEC) 512×512 imaging sites and IMPA U2OS 96×96 single-cell crops, both with six fluorescence channels. IMPA U2OS crops are derived from RxRx1 U2OS batches 01–03. Train and Eval in Table 4 follow the published training and test labels. The IMPA split shares source batches, sites, and wells between training and evaluation; the original RxRx1 U2OS site split uses separate batches for training and testing. The CellFlux eval. subset contains single-cell crops from 100 perturbations and does not represent its full training dataset.

Broader survey Panel (b) reports counts for the surveyed releases. RxRx3-core counts center crops per well; Golgi scattering, Nuclear translocation, and LRRK2 count prepared images in the released Phenexplain PNG archives. The remaining rows in panel (b) count imaging sites. The LINCS count covers 141 plates with publicly indexed image directories. *CellCLIP reports 919,265 CDRP source images and a curated subset of 759,782 images (Lu et al., 2025). †The Golgi count covers only the released single-plate subset. JUMP pilot / cpg0000 denotes CPJUMP1.

Clarifications Image dimensions are in pixels; channel counts refer to fluorescence measurements before RGB packaging. AGP denotes actin, Golgi, and plasma membrane; TH denotes tyrosine hydroxylase. Batches count acquisition groups: experiment IDs for RxRx1 and RxRx3-core, acquisition weeks for BBBC021, and metadata groups for cpg0000, cpg0004, and cpg0017. Repeated imaging sessions are counted separately. CDRP has 406 plates, but its acquisition-batch count is not reported. A dash denotes an unreported acquisition-batch count.

C.2 SCOPE OF PRIOR STUDIES

We survey eleven image-generation studies and two studies of reconstruction or representation learning. Six generators operate directly on image pixels, and five operate in a latent space obtained by encoding the images. GAN style codes alone do not define the latent-space group. Compressing Biology evaluates VAE reconstruction; CellCLIP learns image representations. Table 5 lists each model family and the datasets covered by Table 4. The RGB columns distinguish transforms used in training from those used to compute FID; display-only transforms are excluded.

Table 5: RGB use in training and FID evaluation.

Study	**Image model**	**Dataset**	**RGB transform**	
			Train	**Eval**
(a) Pixel space				
Phenexplain (Lamiable et al., 2023)	GAN	BBBC021; Golgi scattering; Nuclear translocation; LRRK2	✗	✗
PhenDiff (Bourou et al., 2024)	Diffusion	BBBC021; Golgi scattering	✗	✗
IMPA (Palma et al., 2025)	GAN	IMPA U2OS crops; BBBC021; JUMP pilot / cpg0000	✗	✓
CellFlux (Zhang et al., 2025)	Flow matching	IMPA U2OS crops; BBBC021; JUMP pilot / cpg0000	✗	✓
CellFluxRL (Wu et al., 2026)	Flow matching	BBBC021	✗	✗
MiT	Flow matching	IMPA U2OS crops; BBBC021	✗	✓
(b) Latent space				
MorphoDiff (Navidi et al., 2025)	Diffusion	RxRx1 (HUVEC); BBBC021; Rohban / cpg0017	✓	✓
MorphDiff (Wang et al., 2025)	Diffusion	CDRP / Bray2017; Broad LINCS / cpg0004	✗	✓
MorphGen (Demirel et al., 2025)	Diffusion	RxRx1 (HUVEC, HEPG2, RPE, U2OS); Rohban / cpg0017	✗	✓
CellFluxV2 (Zhang et al., 2026)	Flow matching	IMPA U2OS crops; BBBC021; JUMP pilot / cpg0000	✗	✗
FreeBridge (Wang et al., 2026)	Schrödinger bridge	RxRx1 (cell type not reported); BBBC021	✓	✗
(c) Reconstruction and representation				
Compressing Biology (Cropsal & Mercado, 2025)	VAE	JUMP pilot / cpg0000	✗	✓
CellCLIP (Lu et al., 2025)	Contrastive learning	CDRP / Bray2017; RxRx3-core; JUMP pilot / cpg0000	✗	✗
CPGENeval (Ours)	Flow matching	RxRx1 (HUVEC); IMPA U2OS crops	✓	✓

Train covers training inputs, including encoding before latent-space training; FID covers Inception-based evaluation. ✓: an explicit RGB conversion is reported for at least one listed dataset; ✗: none is reported at that stage. Selection and grouping count; native three-channel inputs and per-stain grayscale replication do not. Compressing Biology has no training stage; CellCLIP reports no FID. FreeBridge's training mark follows its released BBBC021 pipeline; its RGB conversion for FID, and CellFluxV2's, are unspecified. Dataset names and release scopes follow the dataset survey.

C.3 VAE Reconstruction Details

VAE reconstruction. We evaluate five frozen RGB VAEs (SD1.4, SD ft-MSE, SDXL, SD3.5, and FLUX.1) under seven rendering protocols using 39,342 512×512 RxRx1 (HUVEC) sites and 136,673 96×96 IMPA U2OS crops from their training splits, including treatment and control images. All protocol–VAE combinations use the same images within each dataset. Rendered RGB inputs are mapped from $[0, 255]$ to $[-1, 1]$, encoded using the posterior mode, and decoded directly without resizing. Encoding and decoding use FP32 with TF32 enabled. Decoded values are clipped to $[-1, 1]$, mapped back to $[0, 255]$, and rounded to 8-bit RGB without further contrast normalization. We report mean per-image PSNR and RGB SSIM against the corresponding rendered inputs (Table 6). Both metrics use intensities rescaled to $[0, 1]$; SSIM uses an 11×11 Gaussian window with $\sigma = 1.5$ and population covariance, without downsampling. Pooled rFID compares the full rendered and reconstructed image sets using 2,048-dimensional Inception-v3-compat features from `torch-fidelity`, with sample means and covariances computed in float64.

Table 6: Frozen RGB VAE reconstruction on HUVEC512 sites (left) and IMPA96 crops (right). All training images, including controls, are included. PSNR (dB) and RGB SSIM are averaged over paired images; higher is better.

		HUVEC512		IMPA96	
Protocol	VAE	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
MorphGen	SD1.4	31.69	0.8176	22.45	0.5293
	SD ft-MSE	32.45	0.8337	24.56	0.6101
	SDXL	33.15	0.8522	25.87	0.6504
	SD3.5	35.53	0.8904	27.01	0.7661
	FLUX.1	37.75	0.9313	30.82	0.8544
CellFlux	SD1.4	49.66	0.9968	24.15	0.5672
	SD ft-MSE	49.16	0.9482	26.00	0.6185
	SDXL	55.38	0.9970	26.45	0.6820
	SD3.5	43.04	0.7007	28.03	0.7971
	FLUX.1	56.39	0.9968	31.48	0.8235
Recursion	SD1.4	28.53	0.7647	27.13	0.8647
	SD ft-MSE	29.38	0.7876	28.72	0.8703
	SDXL	30.01	0.8105	28.95	0.8914
	SD3.5	32.52	0.8747	30.68	0.9204
	FLUX.1	34.29	0.9087	30.96	0.9170
IMPA	SD1.4	37.89	0.8711	23.46	0.5421
	SD ft-MSE	40.11	0.9361	25.89	0.6179
	SDXL	40.95	0.9450	27.14	0.6629
	SD3.5	40.70	0.9209	27.70	0.7509
	FLUX.1	44.51	0.9680	31.34	0.8229
FreeBridge	SD1.4	26.68	0.7544	21.86	0.5570
	SD ft-MSE	27.34	0.7608	23.43	0.6040
	SDXL	27.97	0.7715	23.79	0.6351
	SD3.5	30.46	0.8223	25.39	0.7702
	FLUX.1	31.94	0.8879	28.66	0.8184
MorphDiff	SD1.4	34.49	0.8344	21.90	0.5235
	SD ft-MSE	35.43	0.8817	23.88	0.5973
	SDXL	36.48	0.8983	25.08	0.6444
	SD3.5	38.06	0.9016	26.44	0.7648
	FLUX.1	40.58	0.9519	29.96	0.8266
CellCLIP	SD1.4	28.93	0.7798	25.12	0.7949
	SD ft-MSE	29.77	0.8012	26.43	0.8049
	SDXL	30.43	0.8225	26.46	0.8321
	SD3.5	32.93	0.8798	28.24	0.8825
	FLUX.1	34.61	0.9110	29.13	0.8827

D GENERATOR TRAINING AND EVALUATION

This appendix describes generator training, sampling, and evaluation on HUVEC512 and IMPA96.

D.1 DATA AND INPUT REPRESENTATIONS

HUVEC512. We train six-channel pixel-space and RGB latent-space generators on 35,376 treated 512×512 sites from RxRx1 experiments 01–16, covering 1,108 siRNA conditions (Sypetkowski et al., 2023). The latent models combine seven rendering protocols with five frozen VAEs. Evaluation uses the 1,032 conditions supported by the released MorphoDiff conditioning tables. Of the 17,692 treated sites in experiments 17–24, 16,476 belong to these conditions. This restriction applies only to evaluation; generator training retains all 1,108 conditions.

IMPA96. We train on the released IMPA split of 122,678 treated 96×96 crops with 1,042 perturbation labels (Palma et al., 2025). Evaluation uses the CellFlux subset: 2,942 treated test crops from 100 perturbations and 300 wells (Zhang et al., 2025). Controls are excluded throughout generation evaluation. The published crop split shares wells and source sites between training and testing; retrieval measures condition fidelity within that benchmark, while HUVEC512 provides the held-out-batch comparison for our generators.

Latent inputs. Rendered RGB inputs are mapped to $[-1, 1]$ and encoded once with a frozen VAE. We draw one posterior sample per training image, apply the released shift and scaling constants in Table 7, and cache the result in FP16. Cached latents are loaded in FP32 and reused throughout training; VAE encoding and decoding use FP32 with TF32 enabled. No data augmentation is applied to either representation.

Table 7: Latent channels and released normalization constants. For a posterior sample z, the training representation is $(z - b)s$; this transformation is inverted before VAE decoding.

VAE	Channels	Scale s	Shift b
SD1.4	4	0.18215	0
SD ft-MSE	4	0.18215	0
SDXL	4	0.13025	0
SD3.5	16	1.5305	0.0609
FLUX.1	16	0.3611	0.1159

D.2 GENERATOR ARCHITECTURE

Shared architecture. We adapt DeCo's coarse-resolution transformer and dense, attention-free decoder (Ma et al., 2026), using the dimensions in Table 8. The transformer uses fixed two-dimensional sine–cosine positions, AdaLN-Zero blocks, affine-free LayerNorm ($\epsilon = 10^{-6}$), and GELU MLPs. A 256-dimensional sinusoidal embedding of $1000t$ passes through a $256 \to 768 \to 768$ SiLU MLP and is added to the learned siRNA embedding to condition the transformer. Dense features sum linear projections of the noisy input and 32-dimensional spatial sine–cosine embeddings. Each decoder block uses a SiLU MLP and a separate token-to-dense conditioning projection. Residual modulation is zero-initialized; the final velocity projection uses default linear-layer initialization.

Native pixel decoder. To address visible patch seams with a pointwise pixel decoder, we insert a 7×7 depthwise convolution after adaptive normalization and before each decoder MLP:

$$h_{\ell+1} = h_\ell + \alpha_\ell \odot \mathrm{MLP}_\ell(\mathrm{DWConv}_{7\times7,\ell}[(1 + \gamma_\ell) \odot \mathrm{LN}(h_\ell) + \beta_\ell]) \,. \tag{10}$$

Here h_ℓ has 32 channels; modulation comes from projected transformer features and the time embedding. The convolution spans the full spatial grid across patch boundaries (stride 1, padding 3, 32 groups) and is identity-initialized with a unit center weight and zeros elsewhere, including the bias. The output is a six-channel velocity field.

Attention normalization. The pixel model uses per-head query/key RMSNorm ($\epsilon = 10^{-6}$, learned gains initialized to one), computed in FP32 and cast back to the attention dtype. Latent models omit query/key normalization.

Table 8: Generator configurations for HUVEC512. Parameter counts include the learned siRNA embeddings and exclude the frozen VAE. The two latent counts correspond to 4 and 16 latent channels, respectively.

Setting	Pixel space	RGB latent space
Generator input	$6 \times 512 \times 512$	$C_z \times 64 \times 64$
Input channels	6	$C_z \in \{4, 16\}$
Transformer patch size	16×16	2×2
Transformer tokens	1,024	1,024
Transformer depth / width	10 / 768	10 / 768
Attention heads / head dimension	12 / 64	12 / 64
Transformer MLP width	3,072	3,072
Query / key normalization	RMSNorm	None
Dense decoder grid	512×512	64×64
Decoder depth / width	4 / 32	4 / 32
Decoder MLP width	128	128
Token-to-dense projection	$768 \to 8{,}192$	$768 \to 128$
Decoder spatial convolution	7×7 depthwise	None
Trainable parameters (M)	134.456	108.476 / 108.513

Latent decoder. The latent generator uses a pointwise dense decoder without spatial convolution; the frozen VAE decodes its output to RGB. On HUVEC512, both representations use 1,024 transformer tokens, while the dense grid is 512×512 for pixels and 64×64 for latents.

IMPA96 configuration. Pixel inputs are $6 \times 96 \times 96$ with 16×16 patches; latent inputs are $C_z \times 12 \times 12$ with 2×2 patches, giving 36 tokens in either case. All other architectural settings follow Table 8.

D.3 Training and Sampling

Training objective. Let $x \in \mathbb{R}^{C \times H \times W}$ denote either the normalized native image or the scaled VAE latent, and let c be its siRNA label. We sample $\epsilon \sim \mathcal{N}(0, I)$ and $t = \mathrm{sigmoid}(u)$ with $u \sim \mathcal{N}(0, 1)$, and minimize the mean squared velocity error along the linear interpolation:

$$x_t = (1-t)\epsilon + tx, \qquad \mathcal{L} = \mathbb{E}\left[\frac{\|v_\theta(x_t, t, c) - (x - \epsilon)\|_2^2}{CHW}\right]. \tag{11}$$

This is the only training loss; the siRNA label is replaced by a learned null label with probability 0.1.

Optimization. All CPGENeval models train for 100k updates using AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, zero weight decay, gradient clipping at 10, and seed 42. Both representations use batch size 64 and learning rate 10^{-4} on HUVEC512, or batch size 256 and learning rate 5×10^{-4} on IMPA96, without gradient accumulation. The learning rate increases linearly for 500 updates and then remains constant. Training uses BF16 autocast, FP32 loss computation, TF32, and activation checkpointing. The main comparison uses raw (non-EMA) weights at 100k updates.

Sampling. We integrate from $t = 0$ to $t = 1$ using 50 uniform Euler steps, BF16 network evaluation, and FP32 state updates. For condition c, classifier-free guidance is defined by

$$v_w(x_t, t, c) = v_\theta(x_t, t, \varnothing) + w\big[v_\theta(x_t, t, c) - v_\theta(x_t, t, \varnothing)\big]. \tag{12}$$

We fix $w = 1$, using the conditional prediction alone. Each checkpoint supplies 32 images per HUVEC512 condition (33,024 total), or counts matched to the released IMPA96 test rows (2,942 total). The Gaussian-noise seed is $3{+}1{,}000{,}003j(c){+}r$, where $j(c)$ is the original training-vocabulary index and r is the within-condition replicate index, starting at zero. Each reported result uses one generated pool per checkpoint.

Extended pixel training. We extend the HUVEC512 pixel model to 300k updates with all other training, sampling, and evaluation settings unchanged, saving and evaluating every 20k updates for 15 checkpoints.

Published IMPA96 outputs. We evaluate RGB outputs for CellFlux, IMPA, and PhenDiff distributed by the CellFlux authors (Zhang et al., 2025; Palma et al., 2025; Bourou et al., 2024), matched to the same 2,942 test rows in the released RGB evaluation space. The IMPA and PhenDiff images come from this baseline release, rather than the respective original projects. We use these images directly; our training budget and sampling settings apply only to CPGENeval models. Our CellFlux gFID of 33.02 and PhenDiff gFID of 65.94 match the paper's 33.0 and 65.9 at its reporting precision; IMPA gives 36.14 rather than 41.6, so we report scores recomputed from the released outputs throughout.

MorphoDiff. We sample the released conditional All Batches checkpoint for RxRx1 HUVEC (Navidi et al., 2025) using its perturbation embeddings and DDPMScheduler. Official inference uses the pipeline's 50-step default with classifier-free guidance disabled; setting guidance scale 1 in our stock pipeline preserves this conditional-only path.[9] We use FP16 and batch size 8, with a continuous seed-42 random stream to avoid reusing the same noise sequence across perturbations; the official script resets seed 42 for each perturbation. We evaluate the first 32 images per shared condition against the same RECURSION-rendered real reference. The checkpoint was trained across HUVEC batches; this comparison does not establish held-out-batch generalization for MorphoDiff.

D.4 GENERATION EVALUATION

Features and gFID. HUVEC512 uses the ADM evaluator's (Dhariwal & Nichol, 2021) frozen TensorFlow Inception-v3 graph and its 2,048-dimensional final average-pooling features (`pool_3`).[10] RGB images enter at their native resolution with values in $[0, 255]$; the graph performs its own bilinear resize to 299×299. Pooled gFID (Heusel et al., 2017) uses feature means and sample covariances over all generated and evaluation images, without feature standardization. Real and generated images use the same rendering within each comparison. The same six-channel pixel outputs are rendered under each protocol, with a correspondingly rendered real reference.

For IMPA96, we follow the released CellFlux FID implementation using TorchMetrics 1.7.4 (`normalize=True`) and torch-fidelity 0.3.0. It receives RGB values in $[0, 1]$ and extracts 2,048-dimensional final pooling features using its internal 299×299 bilinear resize. The CELLFLUX real-image loader retains the release's stochastic dequantization, mixing, clipping, and integer conversion (seed 20260924); other protocols replace only the renderer. Reconstruction and readout experiments instead use deterministic CELLFLUX rendering without dequantization noise.

Perturbation retrieval. A query is the mean generated feature vector for one condition; a gallery entry is the mean real feature vector for one evaluation well. For HUVEC512, we retain the first n_c generated draws per condition, where n_c is its real training-site count, giving 32,946 query images and 1,032 query profiles. The gallery contains 8,238 wells, each averaging two sites. For IMPA96, all 2,942 generated crops form 100 queries, and real test crops are averaged into 300 well profiles. Feature means and population standard deviations are fitted only on the protocol-matched real training subset: 32,946 HUVEC512 sites or 11,730 IMPA96 crops. Query and gallery means are standardized and ℓ_2-normalized, then ranked by cosine similarity. Wells with the same perturbation as the query are positives. Average precision groups equal scores at the same threshold, as in `sklearn.metrics.average_precision_score`; mAP averages AP equally across conditions. IMPA96 mAP is our additional analysis, not a metric reproduced from CellFlux.

References and uncertainty. The real-data reference replaces generated queries with condition means from real training images, keeping the gallery and preprocessing fixed; HUVEC512 mAP ranges from 0.865% to 9.640% across protocols. Chance references use 1,000 permutations of query labels with feature scores fixed: globally on both datasets and additionally within the four co-plate condition groups on HUVEC512. Model-wise permutation means range from 0.204% to 0.243% on HUVEC512 (0.204%–0.249% within co-plate groups) and 2.684% to 2.970% on IMPA96 (Table 9). HUVEC512 intervals use 1,000 paired resamples of entire gallery batches and query conditions (seed 20260918), omitting queries without positive wells in a resample. For IMPA96, 10,000 paired resamples of its 100 query conditions hold the gallery fixed (seed 20260925). Intervals use the 2.5th

[9] Official generation script and modified diffusion pipeline.

[10] `https://github.com/openai/guided-diffusion/tree/main/evaluations`

and 97.5th percentiles, with the same draws across models; they condition on the trained checkpoints and generated images and are not adjusted for multiple comparisons.

Table 9: Chance retrieval mAP (%). Entries are ranges of model-wise means over 1,000 label permutations, not confidence intervals. HUVEC512 includes the main pixel and RGB-latent models; IMPA96 additionally includes the CellFlux-distributed outputs in the CELLFLUX row. Co-plate permutations preserve groups of conditions that share training plates.

Protocol	HUVEC512 global	HUVEC512 co-plate	IMPA96 global
MORPHGEN	0.223–0.232	0.228–0.245	2.844–2.925
CELLFLUX	0.204–0.206	0.204–0.207	2.684–2.968
RECURSION	0.224–0.235	0.226–0.246	2.820–2.848
IMPA	0.225–0.232	0.227–0.239	2.853–2.970
FREEBRIDGE	0.231–0.243	0.233–0.249	2.886–2.928
MORPHDIFF	0.217–0.232	0.219–0.240	2.855–2.916
CELLCLIP	0.222–0.233	0.224–0.245	2.811–2.830

Paired mAP differences. Table 10 reports the within-protocol contrasts discussed in Section 4. Both HUVEC512 intervals include zero, so their point-estimate ranking differences do not establish a stable ordering. The IMPA96 query-bootstrap intervals exclude zero, with the SD3.5–Pixel lower bound close to zero.

Table 10: Paired retrieval differences in percentage points: first model minus second. HUVEC512 resamples gallery batches and query conditions; IMPA96 resamples query conditions with the gallery fixed. Intervals are pointwise 95% percentile intervals conditional on the trained models and generated pools.

Dataset	Protocol	Comparison	Δ mAP [95% CI]
HUVEC512	RECURSION	SD1.4 − SD3.5	$0.48\ [-1.33,\ 2.78]$
HUVEC512	CELLCLIP	SD1.4 − SD3.5	$0.97\ [-0.75,\ 3.39]$
IMPA96	RECURSION	SD1.4 − SD3.5	$3.39\ [0.85,\ 5.89]$
IMPA96	IMPA	SD3.5 − Pixel	$3.03\ [0.06,\ 6.07]$
IMPA96	CELLFLUX	Released IMPA − CellFlux	$7.39\ [4.98,\ 9.99]$

D.5 LOGGED TRAINING COST

Training times use a full NVIDIA H100 (80GB), with activation checkpointing for both representations; latent times report the median of available H100 runs. L40S runs contribute only to quality comparisons. Logged time covers data loading, optimization, logging, and intermediate checkpointing and monitoring, excluding queueing, cache preparation, the final checkpoint write, generation, and evaluation. One-time HUVEC512 encoding takes 0.38 hours for SD1.4–RECURSION, in addition to 5.74 training hours; the 300k pixel trajectory takes 83.56 training hours.

D.6 PER-UPDATE COST

Table 11 compares 100k checkpoints on the same H100 with PyTorch 2.10, CUDA 12.9, batch size 64, and the training precision and checkpointing settings above. Each measurement starts from the 100k checkpoint and a fresh AdamW optimizer. After ten warm-up updates, we report the median CUDA-synchronized time over three 20-update blocks, with either the training loader or a batch already on the GPU. Operation counts cover matrix products and convolutions in forward, backward, and checkpoint recomputation, counting each multiply–add as two operations; they exclude elementwise operations, normalization, and optimizer updates. VAE encoding and decoding are excluded.

The pixel model uses $1.43\times$ the counted operations and $4.85\times$ the update time of SD1.4. Removing loading and host-to-device transfer reduces pixel update time by 1.1%. A separate five-update PyTorch trace attributes 88.6% of the pixel–SD1.4 kernel-time difference to the dense decoder, including its projections, normalization, modulation, MLPs, and spatial convolutions. These ratios describe the tested implementations and need not equal the FLOPs ratio.

Table 11: Operation counts and update times on the same H100 at batch size 64. Both latent models use RECURSION; resident inputs exclude loading and host-to-device transfer.

Representation	TFLOPs/update	Loader (ms)	Resident (ms)
Pixel, six channels	66.83	996.4	985.4
SD1.4 latent, four channels	46.71	205.6	205.4
SD3.5 latent, 16 channels	46.72	205.8	205.3

E NATIVE READOUT

E.1 TARGETS AND READERS

Native CellProfiler targets. We compute CellProfiler measurements once from the six native stains, using the library at commit `b611cf0`. Otsu segmentation of the DNA channel defines nuclei, followed by an eight-pixel nearest-nucleus expansion; objects whose expanded region touches the image boundary are excluded. These are fixed nuclear neighborhoods, not whole-cell segmentations. Each stain supplies intensity fractions in four scaled radial bins, giving $6 \times 4 = 24$ targets per object. Zero total native intensity or a nonfinite radial fraction invalidates that target measurement; all representations share these target-specific exclusions. We do not measure CellProfiler features on RGB or reconstructed RGB images.

Spatial inputs and frozen readers. At each native nuclear centroid, we extract the same 64×64 window from every representation, using reflection padding at image boundaries. Channel-wise means and population standard deviations on an 8×8 grid give 384 RGB features or 768 native six-channel features, with intensities scaled to $[0, 1]$. Native features supply an auxiliary readout-capacity reference. For each rendering protocol, a ridge reader predicts the fixed native targets from real RGB features; its coefficients and standardization remain frozen when applied to the five corresponding VAE reconstructions. Only grid features enter the readers. These scores measure predictability under the specified reader; they neither quantify irreversible information loss nor establish an information bound.

E.2 BATCH SEPARATION AND STATISTICAL ANALYSIS

Batch separation and model selection. We use leave-one-experimental-batch-out evaluation, with 16 outer folds for HUVEC512 and three for IMPA96. Each held-out batch includes all of its plates, wells, sites or crops, and objects; all protocols and VAEs use the same split assignments. Within the remaining batches, regularization is selected using three inner batch-grouped folds for HUVEC512 and two for IMPA96. Separately for each input representation, outer fold, and target family, we select one $\lambda \in \{10^{-4}, 10^{-3}, \ldots, 10^{2}\}$ by the mean target-wise inner out-of-fold R^2 and refit on all outer-training batches. Feature centering, scaling, and ridge fitting use only the relevant training fold, with equal well weighting and an unpenalized intercept. VAE reconstructions never enter fitting or hyperparameter selection.

Equal well weighting. For target f, let $\mathcal{W}_f$ contain wells with at least one valid native measurement, and let I_{wf} be the valid objects in well w, with $n_{wf} = |I_{wf}|$. For any object-level quantity u, define the equal-well average

$$\langle u \rangle_f = \frac{1}{|\mathcal{W}_f|} \sum_{w \in \mathcal{W}_f} \frac{1}{n_{wf}} \sum_{i \in I_{wf}} u_i. \tag{13}$$

The same inverse-object-count weighting is used in ridge fitting. After combining the outer-fold predictions, representation s receives the score

$$R^2_{f,s} = 1 - \frac{\langle (\hat{t}_{f,s} - t_f)^2 \rangle_f}{\langle (t_f - \langle t_f \rangle_f)^2 \rangle_f}. \tag{14}$$

Thus, errors are computed for individual objects before averaging within wells, and the numerator and target variance use identical weights. We do not average well-wise or fold-wise R^2 values. Negative scores are retained; a numerically zero target-variance denominator is recorded as undefined rather than replaced by zero.

Within-batch controls. Table 12 gives the overall radial scores and two pooled within-batch summaries for the native and real RGB inputs. The first retains the squared prediction errors in Equation 14 but replaces the global target mean in its denominator with the corresponding batch's equal-well target mean. The second also centers predictions and targets separately within each batch before computing the numerator. The first uses within-batch target variation as its reference; the second removes batch-specific mean prediction errors as well. Both are descriptive evaluations of the existing out-of-fold predictions and do not refit the reader or change the primary score. On IMPA96, batch centering raises the RECURSION RGB score from 0.041 to 0.243, still below half the centered native score of 0.578. With within-batch target variation as the reference, its score before centering is -0.029, showing that batch-specific mean prediction errors contribute to the deficit. Differential saturation under fixed gains is a possible mechanism, but these controls do not isolate its contribution.

Table 12: Radial readout under batch holdout. Overall values are means of 24 target-wise R^2 scores with paired hierarchical 95% confidence intervals. The within-batch reference changes the target-variance denominator; batch centering additionally removes batch-specific mean prediction errors. These controls use the same out-of-fold predictions.

Input	Overall R^2 [95% CI]	Within-batch reference	Batch-centered
HUVEC512			
Native six-channel	0.658 [0.637, 0.670]	0.618	0.640
MORPHGEN	0.621 [0.604, 0.630]	0.573	0.606
CELLFLUX	0.182 [0.147, 0.202]	0.060	0.208
RECURSION	0.652 [0.630, 0.664]	0.612	0.644
IMPA	0.606 [0.578, 0.621]	0.553	0.599
FREEBRIDGE	0.549 [0.525, 0.559]	0.466	0.602
MORPHDIFF	0.584 [0.569, 0.590]	0.523	0.582
CELLCLIP	0.646 [0.622, 0.660]	0.601	0.644
IMPA96			
Native six-channel	0.552 [0.434, 0.631]	0.520	0.578
MORPHGEN	0.496 [0.385, 0.559]	0.460	0.529
CELLFLUX	0.441 [0.273, 0.535]	0.406	0.464
RECURSION	0.041 [-0.198, 0.133]	-0.029	0.243
IMPA	0.558 [0.455, 0.609]	0.530	0.571
FREEBRIDGE	0.512 [0.408, 0.558]	0.480	0.538
MORPHDIFF	0.524 [0.417, 0.573]	0.493	0.542
CELLCLIP	0.149 [-0.057, 0.342]	0.088	0.305

Target aggregation and paired changes. Our primary summary is the arithmetic mean of the 24 radial target-wise R^2 values, equivalently averaging the four bins within each stain and then the six stains with equal weight. The four intensity fractions sum to one and are not treated as independent observations for uncertainty estimation. Table 13 gives the six stain-wise radial means for native inputs and the IMPA RGB protocol. Undefined target-wise scores are excluded from the corresponding mean, without clipping negative values. For each protocol and VAE, the paired change is $\Delta R^2 = R^2_{\mathrm{rec}} - R^2_{\mathrm{RGB}}$, using the same objects, native targets, folds, and frozen reader. Negative values indicate lower readout performance after VAE reconstruction, not necessarily irreversible loss of the target information.

Table 13: Per-stain radial R^2, averaging the four bins within each stain. The IMPA rendering retains DNA, ER, and Actin; RNA, Mito, and Golgi do not enter its RGB channels. These are point estimates under the same batch holdout as the primary scores.

Dataset	Input	DNA	ER	Actin	RNA	Mito	Golgi
HUVEC512	Native	0.644	0.717	0.614	0.704	0.620	0.650
HUVEC512	IMPA RGB	0.630	0.697	0.584	0.660	0.553	0.512
IMPA96	Native	0.607	0.652	0.588	0.662	0.218	0.585
IMPA96	IMPA RGB	0.574	0.653	0.609	0.643	0.330	0.539

Hierarchical paired confidence intervals. We compute 95% percentile intervals from 1,000 hierarchical bootstrap replicates with random seed 885. Each replicate samples experimental batches

with replacement, then plates within each selected batch, and wells within each selected plate, preserving the observed number of units at each level. All objects in a selected well remain together, and the same resampling weights are shared across targets, RGB inputs, and VAE reconstructions. For each replicate, we recompute the weighted target variance and R^2, then form target-family means and paired differences before taking the 2.5th and 97.5th percentiles. Mean and difference intervals are computed jointly, not from marginal confidence bounds. Percentile intervals can be asymmetric around the point estimate under batch resampling. Readers, hyperparameters, masks, and target lists remain fixed throughout resampling, so the intervals do not include uncertainty from refitting or feature selection. They are pointwise, without multiplicity adjustment, and are not standard errors across cross-validation folds. IMPA96 has only three experimental batches; its intervals provide exploratory uncertainty estimates over these observed batches.