MorphGen: Controllable Cell-Image Generation with Biological Representation Alignment

Berker Demirel[1], **Marco Fumero**[1,3], **Theofanis Karaletsos**[2] **and Francesco Locatello**[1]
[1]Institute of Science and Technology Austria (ISTA)
[2]Achira
[3]Kempner Institute, Harvard University

Abstract

Simulating in silico cellular responses to interventions is a promising direction to accelerate high-content image-based assays, critical for advancing drug discovery and gene editing. To support this, we introduce MorphGen, a state-of-the-art diffusion-based generative model for fluorescent microscopy that enables controllable generation across multiple cell types and perturbations. MorphGen is trained with an alignment loss that matches its representations to phenotypic embeddings from a biology foundation model, encouraging meaningful morphological patterns consistent with real cell images. Unlike prior approaches that compress multichannel stains into RGB images, sacrificing organelle-specific detail and focusing on a single cell type, MorphGen generates the complete set of fluorescent channels jointly, preserving per-organelle structure and enabling post-generation interpretation. We demonstrate biological consistency with real images via CellProfiler features, and MorphGen attains an FID score over 35% lower than the prior state-of-the-art MorphoDiff. Finally, in a compositional generalization test that holds out cell type–perturbation combinations during training, MorphGen achieves in-distribution quality on 55% of unseen pairs under a seen-calibrated criterion. Code is available at https://github.com/czi-ai/MorphGen.

1 Introduction

Deep generative models are emerging tools for simulating cellular behavior in computational biology, with early works in modeling gene expression profiles [11, 2] and more recently synthesizing microscopy images [23, 25], which can be easily collected at scale. These models offer the potential to create *in silico* surrogates of biological experiments – virtual systems that capture cellular morphology and its response to perturbations. This is a critical step in the vision of *Virtual Cells* [6]: a generative instrument capable of populating diverse cellular contexts and emulating the effects of genetic or chemical interventions. Realizing such a system could accelerate biological discovery by producing high-quality hypotheses without the time and cost constraints of exhaustive wet-lab experiments. As a practical step toward this vision, we focus on phenotypic image generation under experimentally defined perturbations.

In fact, among the current experimental platforms, microscopy-based high-content screening (HCS) provides a particularly fitting setting for generative models of cellular phenotypes under different perturbations. Automated microscopes now capture hundreds of thousands of single-cell images per experiment, and image-analysis pipelines distill them into high-dimensional phenotypes that reveal subtle biological variations invisible to bulk assays [5, 9]. These rich readouts make HCS an attractive target for *in silico* modeling, where generative models could simulate phenotypic outcomes across perturbations that would be costly to assay exhaustively.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

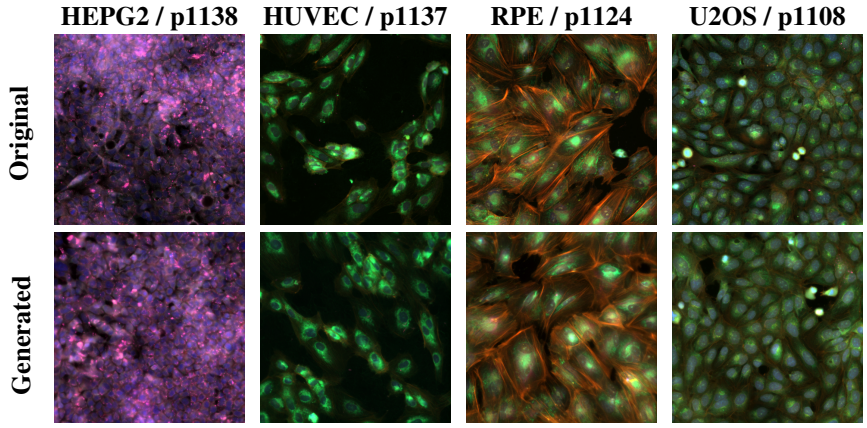


Figure 1: Original (top row) and generated (bottom row) images for various cell type / perturbation ID pairs from the RxRx1 dataset [31]. Unlike existing models, our MorphGen is capable of generating crisp, high-resolution images across different cell-types and perturbations. Generated images are not cherry-picked, and we selected original images that are neighbors of the generated ones for visualization. See Appendix S for additional examples.

A standout HCS protocol is Cell Painting, which stains eight cellular compartments across six fluorescence channels. The resulting morphological fingerprints have proved versatile: they cluster small molecules by mechanism of action [34], map gene function [28], and capture disease-specific signatures for phenotypic drug discovery [33]. With its scale and diversity, Cell Painting enables generative models to learn and generate realistic and biologically faithful images.

However, current image generators fall short of these goals: (i) they operate at low resolution and rely on outdated architectures [25]; (ii) they collapse six-channel fluorescence stacks into lossy RGB, discarding biological detail [27]; and (iii) they are restricted to a single cell type and modest-sized datasets [23]. As a result, they miss out on both fine-grained morphological analysis and realism. Instead, we posit that a generative model should maintain local biological information, even at the individual fluorescence level. Further, restricting generation to a single cell type limits the model's generality, posing a key obstacle toward applications.

We present **MorphGen**, a generative model that addresses these gaps and supports virtual phenotyping across many perturbations and four cell lines. Our contributions are:

- **High-fidelity conditional generation.** MorphGen achieves state-of-the-art fidelity for conditional Cell Painting generation across cell types and perturbations, while jointly synthesizing all six organelle channels.
- **Domain-aligned features.** We repurpose representation alignment from perceptual to semantic alignment, using biology foundation model as a guidance to inject morphology-relevant semantics.
- **Compositional conditioning.** MorphGen supports controllable generation across cell types and perturbations (e.g., swapping cell type while keeping the perturbation fixed), and generalizes to *unseen* cell type–perturbation combinations in a compositional holdout setting.
- **Downstream validation.** Synthetic images reproduce realistic population-level statistics, which we measure as conditional average treatment effects (CATEs) of morphological features across perturbations and controls, confirming suitability for in-silico screening.

Figure 1 illustrates the visual fidelity of our model across four representative cell-type/perturbation pairs. To best of our knowledge, MorphGen is the first generator that delivers high-resolution, organelle-aware, and biologically faithful Cell Painting images at scale for practical *in silico* screening.

2 Related Work

We discuss recent work on controllable Cell Painting image generation for perturbation-induced morphological responses and compare it with MorphGen.

Fully generative diffusion models. MorphoDiff [23] adapts a Stable-Diffusion [29] latent DDPM [13] to Cell Painting. Perturbations are encoded with scGPT embeddings [11]. Six fluorescence channels are projected into RGB through an irreversible compression that merges organelle-specific cues [27] to remain compatible with the pretrained Stable Diffusion VAE [17]. The model trains on full-resolution images from a single cell type in RxRx1 [31], and since unannotated perturbations lack scGPT indices the authors discard those images, limiting general applicability. Consequently, the model considers only annotated perturbations as factors of variation, and its generation quality may not transfer beyond this annotated subset.

Image-to-image translation models. A separate line of work studies Cell Painting generation in an *image-to-image translation* setting [25, 36, 19, 4]: given a *control* cell image (or a batch of controls), the goal is to generate its *perturbed counterpart* by learning a control→perturbation mapping, rather than sampling perturbed cells from scratch as in our fully generative setting. These methods are commonly evaluated under the Image Perturbation Autoencoder (IMPA) pipeline, which applies illumination correction and crops 512×512 Cell Painting fields to 96×96 nuclei-centered patches, with RxRx1 results typically reported on U2OS. Within this regime, IMPA [25] serves as an early baseline, while CellFlux [36] improves substantially using conditional flow matching under the same IMPA preprocessing for comparability. Consequently, these results reflect performance in the same low-capacity crop setting and translation formulation, rather than native-resolution, multi-cell-type and fully generative modeling. We include them for cross-benchmark alignment, not as direct comparisons to our setting.

Comparison to MorphGen. Prior work on Cell Painting generation is largely constrained either by (i) *translation-style* formulations that condition on a control image under heavy preprocessing, or (ii) generative setups that sacrifice biological information through lossy channel compression or low-resolution crops and focus on a single cell type. MorphGen closes these gaps by targeting the native, fully generative problem directly: it models the joint six-channel Cell Painting distribution at 512×512 resolution, without collapsing fluorescence channels into RGB. Concretely, MorphGen wraps each grayscale channel into a three-channel latent and runs diffusion jointly across all six channels, then splits and decodes per channel to preserve organelle-specific structure. We further incorporate an alignment loss to a biological foundation model during training to encourage morphology-relevant structure. Finally, because conditioning is learned directly from images via perturbation and cell-type embeddings, MorphGen can generate *all* cell types and perturbations, whereas MorphoDiff discards unlabeled perturbations and is restricted to HUVEC. Together, these design choices support multi-cell-type generation and compositional generalization to unseen cell type–perturbation pairs while preserving biologically consistent morphology.

3 MorphGen

MorphGen is a generative model that synthesizes high-resolution, biologically meaningful cell images under diverse perturbation and cell-type conditions. It uses a pretrained variational autoencoder (VAE) to map multi-channel images to a latent space, where a conditional diffusion model performs generation. To inject biology-relevant structure during training we leverage guidance from a biological foundation model, using representation alignment. The complete training and test pipeline is depicted in Figure 6 in the Appendix.

We consider a conditional latent diffusion setting for high-resolution, multi-channel fluorescence microscopy images. Let $\mathcal{X} \subset \mathbb{R}^{6\times H\times W}$ denote the image space of six-channel Cell Painting images, where each channel corresponds to a distinct biological signal.

Organelle-aware processing. Since the pretrained VAE encoder is designed for three-channel RGB images, we adapt each grayscale input channel independently by stacking it three times along the channel dimension. Let $\mathbf{x}^{(c)} \in \mathbb{R}^{1\times H\times W}$ be the c-th channel of an image $\mathbf{x} \in \mathcal{X}$. We define its RGB-stacked version as $\tilde{\mathbf{x}}^{(c)} \in \mathbb{R}^{3\times H\times W}$. This design allows us to encode each organelle-specific channel separately, preserving its biological specificity. Moreover, we retain the benefits of the latent-space diffusion framework [29] to process high resolution images. By independently encoding each organelle channel while concatenating them into a shared latent for joint modeling, we enable organelle-level generation and analysis that maintain post-generation channel-wise interpretability and capture cross-organelle dependencies.

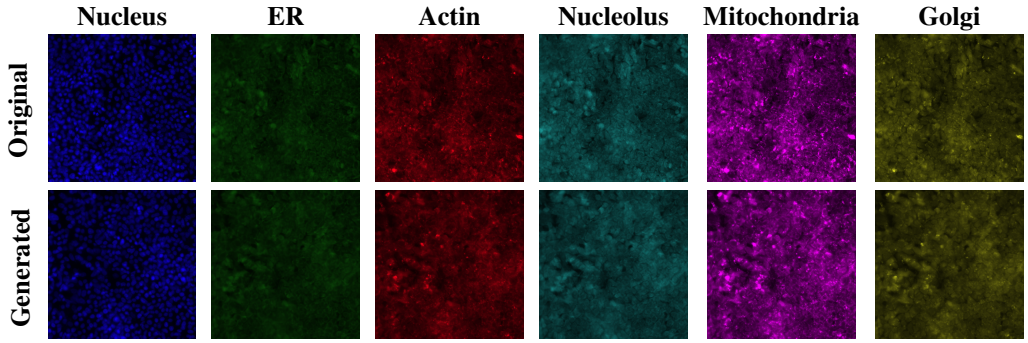


Figure 2: Comparison of original and generated fluorescence images for each organelle in a control HEPG2 cell. Our model reconstructs the six distinct fluorescent channels using RxRx1-recommended colormaps, preserving morphology across subcellular structures. Generated images are not cherry-picked, and we selected original images that are neighbors of the generated ones for visualization.

Each stacked channel $\tilde{\mathbf{x}}^{(c)}$ is passed through a frozen pretrained VAE encoder E_{VAE} to obtain a latent representation:

$$\mathbf{z}^{(c)} = E_{\text{VAE}}(\tilde{\mathbf{x}}^{(c)}) \in \mathbb{R}^{4\times H'\times W'}.$$

where H' and W' denote the spatial resolution of the VAE latent space. We then concatenate the six latents along the channel dimension to form the full latent representation:

$$\mathbf{z} = \text{concat}(\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(6)}) \in \mathbb{R}^{24\times H'\times W'}.$$

Joint diffusion process. The concatenated latent $\mathbf{z}$, encoding all organelle channels, serves as the input to a latent diffusion model parameterized by a Scalable Interpolant Transformer (SiT) [22]. Unlike U-Net-based [30] architectures traditionally used in diffusion models, SiT operates directly on flattened token sequences and excels at modeling complex spatial relationships through global self-attention. This is particularly advantageous for Cell Painting data, where biological signals are distributed across channels and spatially separated structures. Moreover, transformer-based architectures like SiT offer built-in conditioning mechanisms via class tokens and cross-attention layers, in contrast to the more rigid feature-wise modulation used in U-Nets [26, 14]. SiT's transformer backbone, combined with its flexible interpolant framework, enables stable training while preserving morphological detail.

Conditioning is achieved through the combination of perturbation, cell type, and diffusion timestep embeddings. Let $p \in \mathcal{P}$ and $ct \in \mathcal{CT}$ denote the perturbation and cell type labels, respectively, which are mapped to learnable embeddings $\mathbf{e}_p, \mathbf{e}_{ct} \in \mathbb{R}^d$. With the timestep embedding $\mathbf{e}_t$, the conditioning vector $\mathbf{c} = \mathbf{e}_p + \mathbf{e}_{ct} + \mathbf{e}_t$ is used as the cross-attention context in SiT. This formulation attributes distinct variations to perturbation and cell-type factors, enabling factorized control.

Following the EDM formulation [15], the forward diffusion process generates noisy samples by interpolating clean latents $\mathbf{z}_0$ with Gaussian noise. For a timestep $t \in [0, T]$, the interpolation is given by $\mathbf{z}_t = \alpha_t \mathbf{z}_0 + \sigma_t \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, where α_t and σ_t are scaling factors with boundary conditions $\alpha_0 = \sigma_T = 1$ and $\alpha_T = \sigma_0 = 0$. The model predicts the velocity $\mathbf{v}_t$ of the diffusion trajectory, defined as the time derivative of the latent:

$$\mathbf{v}_t = \frac{d\mathbf{z}_t}{dt} = \dot{\alpha}_t \mathbf{z}_0 + \dot{\sigma}_t \boldsymbol{\epsilon}.$$

Given the noisy latent $\mathbf{z}_t$ and conditioning vector $\mathbf{c}$, SiT f_θ estimates this velocity: $\hat{\mathbf{v}}_t = f_\theta(\mathbf{z}_t, \mathbf{c})$, and is trained via mean squared error against the ground-truth:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{z}_0, t, \boldsymbol{\epsilon}} \left[\| f_\theta(\mathbf{z}_t, \mathbf{c}) - (\dot{\alpha}_t \mathbf{z}_0 + \dot{\sigma}_t \boldsymbol{\epsilon}) \|_2^2 \right].$$

This loss ensures the SiT learns a robust velocity prediction, stabilizing training across noise scales.

Transferring biological features. To improve semantic consistency and biological fidelity, we adopt representation alignment regularization loss (REPA) [35] during training. Originally, REPA was proposed to align intermediate representations of SiT with general purpose foundation models such as DINOv2 [24], to accelerate convergence. In contrast, we change both the supervision signal and

Table 1: FID and KID (lower is better) across two datasets. RxRx1/HUVEC uses 50 randomly sampled perturbations. Rohban results are reported for the 5-gene and 12-gene subsets.

Dataset	Subset	Method	FID ↓	KID ↓
RxRx1 (HUVEC)	50 perturbations	Stable Diffusion	115	0.11
		MorphoDiff	78	0.05
		MorphGen	50.2 ± 2.45	0.01 ± 0.000
Rohban dataset	5 genes	Stable Diffusion	326	0.45
		MorphoDiff	251	0.33
		MorphGen	100.24 ± 1.53	0.05 ± 0.014
Rohban dataset	12 genes	Stable Diffusion	317	0.45
		MorphoDiff	277	0.38
		MorphGen	123.93 ± 3.51	0.08 ± 0.017

the objective of representation-geometry alignment: instead of matching generic perceptual features, we align to a biology foundation model whose embeddings encode phenotypic semantics, thereby *transferring* morphology-relevant structure for downstream analysis.

Given an input image $\mathbf{x}$, we extract reference patch-level embeddings: $y^\star = F(\mathbf{x}) \in \mathbb{R}^{N \times d'}$, using the domain-expert as the feature extractor F. Here, N is the number of patches and d' is the embedding dimension. Let $h_t^{(k)} \in \mathbb{R}^{N \times d}$ denote the hidden representation at layer k of the SiT at timestep t. This hidden representation is projected through an MLP h_ϕ into dimension d' to align with $y^\star$. The REPA loss encourages patch-level alignment via cosine similarity:

$$\mathcal{L}_{\text{REPA}} = -\frac{1}{N} \sum_{n=1}^{N} \text{sim}\left(y_n^\star, h_\phi(h_{t,n}^{(k)})\right).$$

The total training objective is $\mathcal{L} = \mathcal{L}_{\text{diff}} + \lambda \mathcal{L}_{\text{REPA}}$, where λ balances their relative importance. We fix $k = 8$ and $\lambda = 0.5$ in all experiments following [35].

Sampling process. At inference, we generate new six-channel images by running a fixed-step Euler–Maruyama sampler in latent space. Starting from Gaussian noise in concatenated latent space, we step through a sequence of noise levels (by default 50 steps), and at each step use the trained SiT to predict the instantaneous "drift" that moves the latent toward a clean signal. We add a small noise term scaled to the current noise level, then repeat. Once the final step is reached, we split the $24 \times H' \times W'$ tensor back into six channel-specific latent representations. Each of these is decoded separately through the VAE decoder, yielding six RGB stacks of size $3 \times H \times W$. We then collapse each stack to single grayscale channel by averaging its three color planes, and recombine all six to form the final $6 \times H \times W$. This inverse procedure enables six-channel fluorescent Cell Painting image generation. Importantly, after training, we do *not* use alignment during sampling; generation is independent of the domain-expert used.

Altogether, our approach enables *high-resolution*, *organelle-aware* synthesis of all six Cell Painting channels by jointly modeling distinct fluorescence signals with *biology-guided* alignment. It further supports flexible, compositional conditioning, allowing *controllable generation* across cell types and perturbations, including *unseen* combinations.

4 Experiments

We evaluate MorphGen along five axes to highlight its strengths and capabilities. We first benchmark the visual fidelity of our generations against prior methods on the standard RxRx1 [31] and Rohban [28] datasets (Section 4.1), demonstrating state-of-the–art results. We then extend this fidelity evaluation to capabilities that existing models do not cover, including organelle-specific and multi-cell-type generation results (Section 4.2). Because visual fidelity does not necessarily imply biological faithfulness, we study the morphology of generated samples in CellProfiler feature space on the most frequent HUVEC perturbations (Section 4.3). Next, we attribute the observed gains by ablating the aligned representations by comparing against alternative alignment choices (Section 4.4). Finally, we stress-test generalization with a compositional holdout of cell type–perturbation combinations (Section 4.5). Dataset and implementation details are provided in Appendix C and D.

Table 2: FID and KID scores for 50 random perturbations across all cell types. Our method supports generation for all four cell types (HEPG2, HUVEC, RPE, U2OS) and provides channel-wise control. MorphoDiff* only supports RGB generation for HUVEC cells.

Method	Metric	RGB	Nucleus	ER	Actin	Nucleolus	Mito	Golgi
Ours	FID↓	**50.2**	27.6	48.1	57.6	49.6	43.6	59.0
	KID↓	**0.008**	0.010	0.011	0.015	0.013	0.012	0.012
	rFID↓	1.411	1.691	1.774	1.562	1.612	1.489	1.564
MorphoDiff*	FID↓	78	—	—	—	—	—	—
	KID↓	0.05	—	—	—	—	—	—

*MorphoDiff is trained only on the HUVEC cell type and does not support channel-wise generation.

Metrics. We report Fréchet Inception Distance (FID) and Kernel Inception Distance (KID), computed *per condition* (perturbation and cell type) and then averaged. All experiments are repeated with three random seeds. When comparing results across conditions with different target distributions, we use *Relative FID (rFID)*. This is defined as $\mathrm{rFID} = \mathrm{FID}(\mathrm{gen}, \mathrm{real}_2)/\mathrm{FID}(\mathrm{real}_1, \mathrm{real}_2)$, $\mathrm{real}_1, \mathrm{real}_2$ are two mutually exclusive real subsets of the same data. A score of 1 indicates generation quality as good as the real distribution.

4.1 Comparison with state-of-the-art

Setting. To evaluate MorphGen's generation quality under a fair comparison with MorphoDiff, we follow the experimental protocol of [23]. Metrics are independently computed for 50 random siRNAs; then averaged across perturbations. For RxRx1, although MorphGen natively generates full 6-channel images across multiple cell lines, we restrict it to HUVEC-only generation—matching MorphoDiff's capacity—and convert our outputs to the RGB space using Recursion's script [31].

Results. Table 1 demonstrates that, even under MorphoDiff's constrained HUVEC-only, RGB-mapped RxRx1 evaluation, MorphGen achieves substantial improvements: it reduces FID by 64.8 and 27.8, and KID by 0.10 and 0.04, compared to Stable Diffusion and MorphoDiff, respectively. Such large gains under a constrained setting highlight MorphGen's fidelity. Moreover, Figure 1 provides complementary qualitative evidence: across multiple cell-type/perturbation pairs, MorphGen faithfully reproduces the morphology, visually corroborating our quantitative results.

To demonstrate the performance beyond the RxRx1 dataset, we evaluated it on the Rohban dataset under two settings: 5-gene and 12-gene subsets. Similarly, MorphGen achieves state-of-the-art performance in both cases, outperforming prior baselines, as shown in Table 1. On the 5-gene subset, FID drops from 251 to 100.24 ± 1.53 ($\sim 60\% \downarrow$), on the 12-gene subset, it falls from 277 to 123.93 ± 3.51 ($\sim 55\% \downarrow$). Moreover, Appendix G reports the cross pipeline comparison with low-resolution models, while Appendix H provides matched comparisons with the released MorphoDiff checkpoint, architectural controls, and computational-cost analysis.

4.2 Additional Capabilities

Setting. To demonstrate MorphGen's additional capabilities beyond prior work, we evaluate (i) organelle-specific generation by measuring fidelity separately for each of the six channels, and (ii) multi-cell-type generation by measuring fidelity per cell type without perturbation conditioning. In the organelle-specific setting, we use the same 50 siRNAs (as in Section 4.1) and report results for each channel; in the cell-type-level setting, we report results per cell type without perturbation conditioning.

Organelle-Specific Generation. To demonstrate MorphGen's fine-grained control, following a similar procedure, we sample 50 siRNAs at random but this time across all four cell types (HEPG2, HUVEC, RPE, U2OS), while matching the original cell-type distribution. For each of the six fluorescence channels (nucleus, ER, actin, nucleolus, mitochondria, and golgi), we extract real and generated single-channel images, replicate them three times to form 3-channel inputs, and compute FID and KID. As shown in Table 2, the mitochondrial channel yields the lowest rFID (1.489), while ER is comparatively harder to model (rFID 1.774). Moreover, Figure 2 provides a qualitative comparison of original versus generated single-channel images in a control HEPG2 cell. The generated outputs closely mirror the morphology and texture of the originals, further

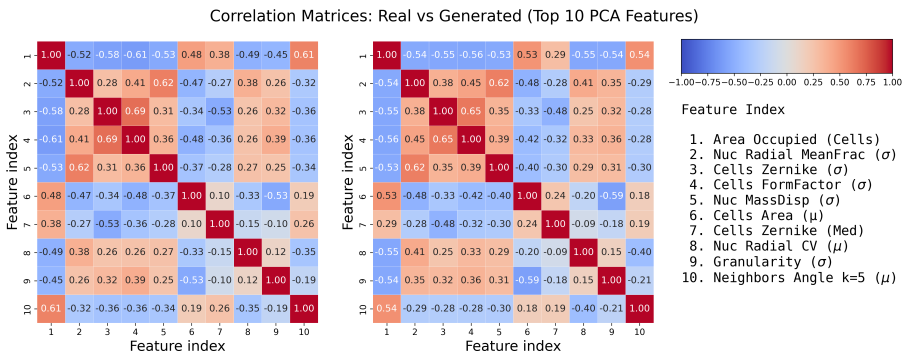


Figure 4: **CellProfiler morphology analysis (HUVEC).** Correlation matrices for the top-10 PCA-selected features in real and generated data, shown side-by-side with a shared scale, indicate that MorphGen preserves key morphological relationships.

demonstrating MorphGen's ability to faithfully reproduce subcellular observations. Appendix N reports the full table with confidence intervals.

Cell-Type-Specific Generation. To assess MorphGen under more natural data distributions, we randomly sample images by cell type (HEPG2, HUVEC, RPE, U2OS) without conditioning on perturbations, avoiding any data augmentation to reach a fixed sample count.Table 3 reports FID, KID and Relative FID on the resulting RGB-converted images.

MorphGen achieves its best scores on HUVEC (rFID 1.136) and maintains strong performance across the other cell types (e.g., RPE: rFID 1.185). These cell-type-specific results outperform perturbation-level experiments, which required aggressive augmentation to inflate small-perturbation sets—particularly those with fewer than 50 unique samples—introducing bias into the real data distribution. By contrast, when following the natural data distribution without artificial augmentation, performance increases further, demonstrating MorphGen's superior fidelity. We refer to Appendix O for the full results with confidence intervals.

Table 3: Per-Cell type specific results. MorphGen is capable of generating high-fidelity images for different cell types.

Cell Type	FID ↓	KID ↓	rFID ↓
HEPG2	41.1	0.016	1.529
HUVEC	28.7	0.006	1.136
RPE	34.4	0.007	1.185
U2OS	38.2	0.017	1.492

4.3 Morphology Analysis

Setting. We evaluated morphological fidelity using CellProfiler [10], extracting features for real and generated images of the four most common perturbations: 1108, 1124, 1137 and 1138 (control) for HUVEC cells. After variance-thresholding, standardization, and removing highly collinear features, we visualize the feature space with PCA and compute correlation matrices over the top-10 PCA features for both real and generated sets.

Biological Validity. From the Figure 3, we observe that (i) real and generated embeddings largely overlap within the same perturbation class, indicating that generated images reproduce morphology faithfully and (ii) perturbation-specific deviations from the control are similarly captured in both real and generated distributions, evident from the color separation. In addition, the pairwise correlation structure of key CellProfiler features closely matches between real and generated data shown in Figure 4. This suggests MorphGen captures biologically meaningful relation-

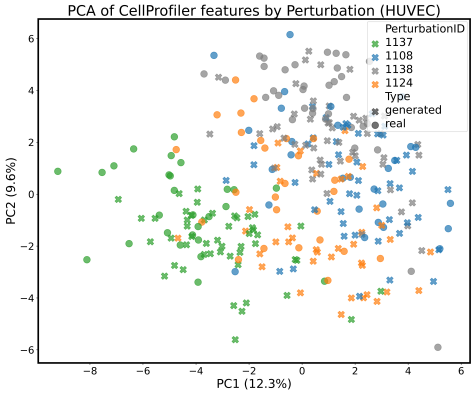


Figure 3: **PCA of CellProfiler features.** Color denotes perturbation (1108, 1124, 1137, 1138); marker style denotes data type (circle: real, cross: generated). Generated samples align with real clusters while maintaining perturbation separation.

ships rather than only marginal statistics. See Appendices J F, L and R for analyses on downstream classification performance (quantitative) and additional visualizations (qualitative).

CATE: Conditional Average Treatment Effect. To quantitatively validate that our generated images capture biologically meaningful perturbation effects [2] at the population level, we compute the Conditional Average Treatment Effect (CATE) between control and perturbed samples using CellProfiler features to represent cellular morphology, denoted as Y. Given a perturbation p, we define CATE as associational difference between a treated population and a control group for a specific cell type:

$$\mathrm{CATE}(p) = \Big\| \mathbb{E}[Y \mid P = \text{control},\ ct = \text{HUVEC}] - \mathbb{E}[Y \mid P = p,\ ct = \text{HUVEC}] \Big\|^2.$$

This metric captures the squared Euclidean distance between the average feature vectors of the control group (perturbation 1138) and a perturbed group p. We compute the CATE separately for real and generated samples across the three most common perturbations: 1108, 1124, and 1137.

Therefore, we estimate the Average Treatment Effect in feature space, as the associational difference will carry over to any downstream prediction-powered (morphological) predictor [8, 7]. We remark that the goal of this experiment is to show that the conclusions drawn from population-level statistics of morphological features match between real and generated samples. Specific causal conclusions may still be invalid, depending on whether hidden confounding or consistency, exchangeability, and overlap assumptions hold on the real data.

Table 4: **CATE agreement in CellProfiler feature space (HUVEC).** Average treatment effects (control p1138 vs. perturbation) computed from real and generated images. Lower ΔCATE indicates closer agreement.

Comparison	$\mathrm{CATE}_{\mathrm{real}}$	$\mathrm{CATE}_{\mathrm{gen}}$	ΔCATE$\downarrow$	Ratio
p1138 vs p1108	16.28	14.76	1.52	0.91
p1138 vs p1124	22.86	25.56	2.70	1.12
p1138 vs p1137	42.07	42.14	0.07	1.01

As shown in Table 4, perturbation 1137 results in the largest morphological deviation from the control, while perturbation 1108 has the smallest effect. These magnitudes align well with the spatial patterns observed in the PCA projections (Figures 3 and 7). The close agreement between CATE values computed from real and generated images further indicates that our model reliably captures the perturbation effects. This experiment, conclusively shows that they do, and that conclusions from statistical associations at the population level between real and generated images match well. For organelle and cell-type specific visualizations and CATE results, refer to Appendices Q and P.

4.4 Impact of aligning to biological features

Setting. To test whether representation alignment transfers useful domain structure into the generator, we train MorphGen variants with (i) no alignment, (ii) alignment to a generic vision encoder (DINOv2), and (iii) alignment to biology-domain expert models(CellCLIP [21] and OpenPhenom). We evaluate fidelity on HUVEC perturbations (Table 5) and assess population-level morphological consistency in CellProfiler via CATE ratios (Table 6).

Results. Table 5 shows aligning to DINOv2 yields only a modest improvement over the non-aligned baseline on HUVEC perturbations (FID $56.87 \rightarrow 55.26$), indicating that generic representation guidance alone is not sufficient. In contrast, alignment to biology-domain experts leads to larger gains: both CellCLIP and OpenPhenom substantially improve fidelity, reducing FID to 53.70 and 50.20, respectively. This demonstrates that the alignment loss allows to transfer relevant morphologies from domain-specialized feature spaces, rather than acting as a generic regularizer. In addition, OpenPhenom alignment improves population-level morphological consistency: the generated CATE ratios are closer to 1 than under DINOv2

Table 5: **Transferring biological features improves generation quality.** We compare MorphGen without guidance and with guidance from different pretrained encoders. Domain-specific guidance (OpenPhenom) yields the best FID/KID on 50 randomly sampled HUVEC perturbations.

Method	FID$\downarrow$	KID$\downarrow$
MorphGen w/o align.	56.87 ± 3.35	0.023 ± 0.001
MorphGen w/ DinoV2	55.26 ± 1.54	0.020 ± 0.001
MorphGen w/ CellCLIP	53.70 ± 2.11	0.022 ± 0.002
MorphGen w/ OpenPhenom	$\mathbf{50.20 \pm 2.45}$	$\mathbf{0.018 \pm 0.000}$

alignment (Table 6), indicating stronger agreement with treatment effect estimates using real images in CellProfiler feature space. Overall, biology-expert alignment consistently improves generation quality compared to both no alignment and alignment with a generic visual encoder. To our knowledge, these results provide early quantitative evidence in microscopy that representation matching can transfer domain expertise from a discriminative, pretrained encoder into a generative model.

4.5 Compositional Virtual Instrument

Setting. To test whether MorphGen can generalize compositionally to unseen cell type–perturbation pairs, we evaluate it in a controlled *virtual instrument* setting on RxRx1. To this end, we train (i) an *oracle* model that observes all cell type and perturbation combinations during training and (ii) a *compositional* model where we hold out 100 randomly selected perturbations on the HUVEC cell type, while still exposing these perturbations on other cell types. This construction enforces a purely compositional generalization requirement: the model must generate unseen (HUVEC, p) pairs using the knowledge of p learned from other cell types.

Table 6: **Biological feature alignment yields better CATE consistency (HUVEC, CellProfiler space).** We report generated CATE and its ratio to the real. Ratios closer to 1 indicate better agreement. Across perturbations, semantic guidance matches real CATE better than perceptual guidance.

Guide	Metric	p1108	p1124	p1137
DINOv2	CATE_{gen}	11.67	18.64	48.92
	Ratio (gen / real)	0.72	0.81	1.16
OpenPhenom	CATE_{gen}	14.76	25.56	42.07
	Ratio (gen / real)	**0.91**	**1.12**	**1.01**

To quantify compositional generalization, we compare the compositional model against the oracle per perturbation using rFID measured on HUVEC. We convert this comparison into a *seen-calibrated* acceptance test. For each perturbation p, define the rFID gap:

$$\Delta(p) = \text{rFID}_{\text{comp}}(p) - \text{rFID}_{\text{oracle}}(p),$$

setting δ to the 95th percentile of the gap over HUVEC perturbations *seen* by both models, $\delta = Q_{0.95}(\{\Delta(p)\}_{p \in \mathcal{P}_{\text{seen}}})$. A held-out perturbation is accepted if the compositional model stays within the oracle–compositional gap observed in-distribution.

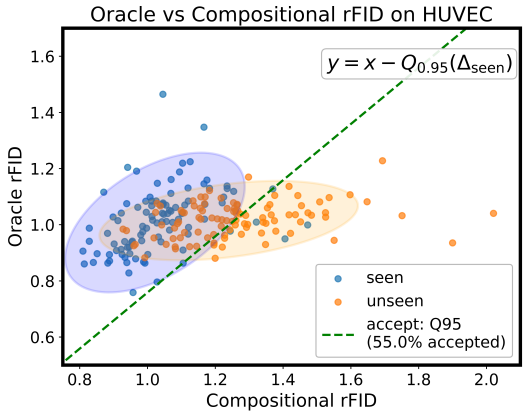


Figure 5: **Oracle vs. compositional rFID on HUVEC.** Each point corresponds to a perturbation p evaluated on HUVEC. The x-axis shows the compositional model rFID and the y-axis shows the oracle model rFID. Blue points are *seen* HUVEC perturbations, while orange points are *unseen* HUVEC perturbations (held out only for the compositional model). The dashed green line shows the $Q_{0.95}$ seen-calibrated acceptance boundary (δ); 55% of unseen perturbations are accepted.

Results. Figure 5 compares the oracle and compositional models in terms of rFID on HUVEC. Using the seen-calibrated threshold at $Q_{0.95}$, the compositional model is accepted on 55% of held-out (HUVEC, p) combinations. To contextualize this result, we evaluate two naive baselines under the same criterion: a *perturbation-agnostic* baseline, which ignores perturbation identity, and a *label-shuffled* baseline, which conditions on an incorrect perturbation. These are accepted only 11% and 4% of the time, respectively, showing that the compositional model is not merely generating generic HUVEC-like images or relying on arbitrary perturbation labels, but captures perturbation-specific structure. See Appendix M for further details on compositional generalization and Appendix M.1 for additional biological validation on unseen compositions, including treatment effect agreement, perturbation classification and failure cases.

5 Conclusion

We introduced MorphGen, a diffusion model that synthesizes high-resolution, six-channel Cell Painting images across perturbations and cell types. We demonstrated that representation alignment

enables the transfer of biologically meaningful features from domain-expert models, encouraging morphology-relevant structure for scientific use. In this way, MorphGen improves fidelity over prior work while supporting compositional generalization, generating unseen cell type–perturbation pairs within a tolerance calibrated on seen data and preserving biologically consistent morphology, as validated by CellProfiler analyses and CATE agreement. One open challenge is extrapolation to entirely novel conditions beyond the training distribution. Furthermore, although it is helpful, aligning with biological foundation models may inherit their biases to the generator, potentially emphasizing certain morphological features over others. Future work will explore instance-based conditioning that learns perturbation representations directly from example images, reducing reliance on perturbation labels, as well as strategies to mitigate biases transferred from the aligned models.

Acknowledgements

This work was supported by the Chan Zuckerberg Initiative (CZI) through the AI Residency Program. We thank CZI for the opportunity to participate in this program and the CZI AI Infrastructure Team for support with the GPU cluster used to train our models.

We are grateful to Bianca M. Dumitrascu and Beini Wang for guidance on our CellProfiler analysis; in particular, we thank Beini for giving us a CellProfiler tutorial. We also thank Christoph Bock for fruitful discussions on experimental design.

References

[1] Stability AI. sd-vae-ft-mse. https://huggingface.co/stabilityai/sd-vae-ft-mse, 2022. Fine-tuned VAE decoder for Stable Diffusion, optimized with MSE + LPIPS loss.

[2] Michael Bereket and Theofanis Karaletsos. Modelling cellular perturbations with the sparse additive mechanism shift variational autoencoder. *Advances in Neural Information Processing Systems*, 36, 2024.

[3] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[4] 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 *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 358–367. Springer, 2024.

[5] Michael Boutros, Florian Heigwer, and Christina Laufer. Microscopy-based high-content screening. *Cell*, 163(6):1314–1325, 2015.

[6] Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B Burkhardt, et al. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25):7045–7063, 2024.

[7] Riccardo Cadei, Ilker Demirel, Piersilvio De Bartolomeis, Lukas Lindorfer, Sylvia Cremer, Cordelia Schmid, and Francesco Locatello. Causal lifting of neural representations: Zero-shot generalization for causal inferences. *arXiv preprint arXiv:2502.06343*, 2025.

[8] Riccardo Cadei, Lukas Lindorfer, Sylvia Cremer, Cordelia Schmid, and Francesco Locatello. Smoke and mirrors in causal downstream tasks. *Advances in Neural Information Processing Systems*, 37:26082–26112, 2024.

[9] Juan C Caicedo, Sam Cooper, Florian Heigwer, Scott Warchal, Peng Qiu, Csaba Molnar, Aliaksei S Vasilevich, Joseph D Barry, Harmanjit Singh Bansal, Oren Kraus, et al. Data-analysis strategies for image-based cell profiling. *Nature methods*, 14(9):849–863, 2017.

[10] 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*, 7:1–11, 2006.

[11] Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, 21(8):1470–1480, 2024.

[12] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[14] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[15] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[16] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024.

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv e-prints*, pages arXiv–1312, 2013.

[18] Oren Kraus, Kian Kenyon-Dean, Saber Saberian, Maryam Fallah, Peter McLean, Jess Leung, Vasudev Sharma, Ayla Khan, Jia Balakrishnan, Safiye Celik, et al. Masked autoencoders for microscopy are scalable learners of cellular biology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11757–11768, 2024.

[19] 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.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[21] Mingyu Lu, Ethan Weinberger, Chanwoo Kim, and Su-In Lee. Cellclip–learning perturbation effects in cell painting via text-guided contrastive learning. *arXiv preprint arXiv:2506.06290*, 2025.

[22] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024.

[23] 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 *The Thirteenth International Conference on Learning Representations*, 2025.

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, pages 1–31, 2024.

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

[26] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019.

[27] Lawrence Phillips. Cellrep: A multichannel image representation learning model. In *1st CVPR Workshop on Computer Vision For Drug Discovery (CVDD): Where are we and What is Beyond?*, 2025.

[28] 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*, 6:e24060, 2017.

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[31] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4285–4294, 2023.

[32] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[33] Fabien Vincent, Arsenio Nueda, Jonathan Lee, Monica Schenone, Marco Prunotto, and Mark Mercola. Phenotypic drug discovery: recent successes, lessons learned and new directions. *Nature Reviews Drug Discovery*, 21(12):899–914, 2022.

[34] Mathias J Wawer, Kejie Li, Sigrun M Gustafsdottir, Vebjorn Ljosa, Nicole E Bodycombe, Melissa A Marton, Katherine L Sokolnicki, Mark-Anthony Bray, Melissa M Kemp, Ellen Winchester, et al. Toward performance-diverse small-molecule libraries for cell-based phenotypic screening using multiplexed high-dimensional profiling. *Proceedings of the National Academy of Sciences*, 111(30):10911–10916, 2014.

[35] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations*, 2025.

[36] 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 *Forty-second International Conference on Machine Learning*, 2025.

A Training and Inference Pipeline

In Figure 6 we illustrate the training and inference stages of our method.

B Ethics Statement and Broader Impact

All datasets used in this work [31, 28] are publicly available and were commonly used and processed by prior works [25, 28, 31, 36, 23]. While the generated samples from MorphGen are partially validated using Cell Profiler and OpenPhenom, the samples, especially for combinations of cells and perturbations not in the training data should not be interpreted as definitive biological ground truth without further experimental validation. To mitigate potential misuse, we emphasize that the work is intended for advancing computational methodology in machine learning, not for direct clinical application.

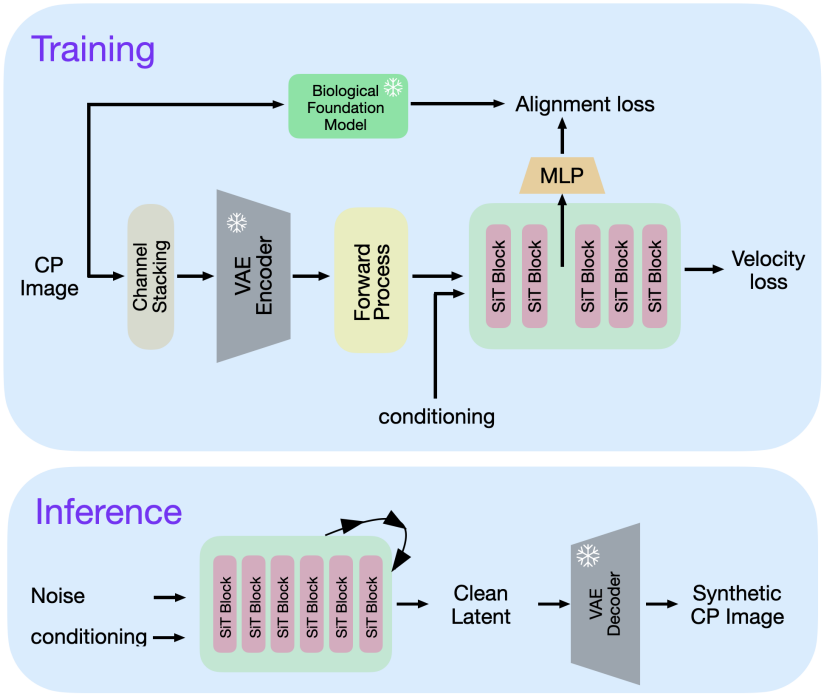


Figure 6: **Overview of MorphGen. Top (training).** A Cell Painting (CP) image is first processed with channel stacking and encoded by a frozen VAE encoder into a multi-channel latent. The latent is perturbed by the diffusion forward process and fed, together with timestep, cell type, and perturbation embeddings (conditioning), into a SiT-based backbone. An MLP projects the SiT features to the OpenPhenom space to compute the alignment loss, while the diffusion head is trained with a velocity loss on the noisy latent. **Bottom (inference).** Starting from noise and the same conditioning embeddings, the SiT backbone iteratively denoises the latent to a clean latent, which is decoded by the frozen VAE decoder to obtain a synthetic CP image.

C Dataset

C.1 RxRx1 dataset.

This dataset is a large-scale, high-resolution collection of fluorescence microscopy images designed to support the study of phenotypic cellular responses to gene knockdowns and to benchmark batch effect correction methods [31]. It comprises $125,510$ images from four human cell types (HUVEC, RPE, HepG2 and U2OS), each exposed to one of $1,108$ siRNA treatments targeting distinct genes, along with 30 non-targeting control conditions. Imaging is performed using a modified Cell Painting assay, generating six-channel 512×512 pixel images that visualize major subcellular structures including the nucleus, endoplasmic reticulum, actin cytoskeleton, nucleoli, mitochondria and golgi apparatus. By capturing morphological changes induced by siRNA perturbations, RxRx1 serves as a challenging benchmark for models aiming to generalize across perturbations, cell types and experimental batches.

C.2 Rohban dataset.

This dataset comprises Cell Painting images of U2OS cells after ORF overexpression of 323 genes. Following MorphoDiff [23], we adopted the same preprocessing protocol (including uniform resizing of 512×512). For analyses, we focus on two small gene subsets: (i) 5 genes from pathways reported to affect cellular morphology (2250 images), and (ii) 12 genes selected via clustering in [28] based on morphological features (3150 images).

D Implementation Details

MorphGen is trained as a latent-diffusion model operating on SD-VAE [29] latents of six-channel RxRx1 [31] training set images of 512×512 resolution. Each single-channel grayscale image is stacked into a 3-channel RGB format, scaled to $[-1, 1]$, encoded with the public stabilityai/sd-vae-ft-mse VAE [1], and rescaled by the SD constants $(0.18215, 0)$. The diffusion backbone is a Scalable Interpolant Transformer (SiT XL/2) [22]. During training, OpenPhenom [18] embeddings of the raw image are injected via a REPA-style projection loss (weight = 0.5), mirroring the original REPA formulation [35].

Optimization uses AdamW (β = 0.9/0.999, lr = 1×10^{-4}) [20] in mixed precision (fp16 or bf16) under HuggingFace Accelerate [12]; TF32 kernels can be enabled for additional speed. An EMA [16] shadow network (decay = 0.9999) is maintained for sampling. Sampling uses a 50-step Euler–Maruyama schedule [22].

Training runs with a batch size of 16 for up to 400 k steps using 8 H100 GPUs. Checkpoints are written every 50 k steps, and the best model is chosen by the average FID across distributed workers, computed on 100 real vs. generated images of pre-selected perturbations using Inception-V3 [32] features. Logging and image grids are tracked in Weights and Biases [3], and all hyper-parameters, metrics, and checkpoints are stored for full reproducibility.

During evaluation, we adopted the augmentation policy of [23] when the real dataset for a given perturbation contained < 500 examples, and synthetically expanded it using random flips and $90°$ rotations.

D.1 Architectural Details

KL-regularized VAE. The KL-regularized VAE used in *stabilityai/sd-vae-ft-mse* is a fully convolutional encoder-decoder that compresses a $3 \times 512 \times 512$ RGB image into a $4 \times 64 \times 64$ latent tensor and then reconstructs it. The encoder begins with a 3×3 convolution followed by three residual blocks, each ending in a stride 2 down-convolution, so channels scale $128 \to 256 \to 512$ while the spatial size shrinks $512 \to 256 \to 128 \to 64$. A 1×1 convolution converts the 512-channel map into eight channels holding per-pixel mean and log-variance; sampling 4-channel latent from these statistics is implemented through the reparameterization trick. The decoder mirrors the encoder: a 1×1 post-quant convolution restores 512 channels, and a final 3×3 convolution followed by tanh maps to RGB. The network contains 81 million learnable parameters (42M in the encoder, 39M in the decoder).

OpenPhenom. OpenPhenom [18] is a channel-agnostic masked autoencoder (CA-MAE) built on a *Vision Transformer Small (ViT-S* backbone with 16×16 patching. It tokenizes each image into patch embeddings, and passes them through 12 transformer layers (with hidden size 384 and 6 heads). A channel-wise cross-attention block lets information flow between stains, and after masking most patches during pre-training the lightweight decoder learns to reconstruct them. At inference time the decoder is dropped and the encoder outputs a fixed 384-dimensional embedding for either one vector per image, or one per channel for finer control. The entire network contains around 22 millions parameters, making it compact for downstream transfer learning.

Scalable Interpolant Transformer (SiT). SiT is a diffusion/flow-hybrid generator that swaps the U-Net of classic DDPMs for a Vision-Transformer backbone. It first splits a latent image into 2×2 patches, yielding a sequence of tokens that pass through 28 transformer blocks with 1152-dimensional hidden size and 16-head self-attention. Each block uses adaptive layernorm-zero modulation: the sum of a sinusoidal timestep embedding and a learned class label embedding is projected and applied as per-channel shifts/scales/gates, letting the same weights serve any diffusion step or class. The encoder is followed by a small MLP projectors which outputs intermediate features for multi-scale predictions, while a final layer unpatchifies the sequence back to an image-shaped residual that guides the interpolant sampler. SiT-XL retains 675 million parameters, yet consistently attains lower FID thanks to the more flexible interpolant objective and an Euler-Maruyama sampler tuned post-training.

E Ablations

Channel-wise processing. To evaluate how different preprocessing strategies affect reconstruction fidelity using a pretrained VAE, we compared three settings. In the baseline setup (RGB), all six fluorescence channels are first compressed into an RGB image before encoding, as done in prior work like MorphoDiff. In contrast, our organelle-aware strategy (MorphGen) processes each channel independently by replicating it across RGB channels to match the VAE's input expectations. We then consider two ways of reporting reconstruction loss. First, we compute the per-channel reconstruction loss and average across channels, yielding a remarkably low MSE of 4.93×10^{-5}. Second, we recompose the reconstructed channels into a 6-channel image and apply the same RGB conversion used in the baseline, resulting in an MSE of 4.06×10^{-4}. As shown in Table 7, both variants outperform the RGB baseline, demonstrating that organelle-aware processing enables better use of the pretrained VAE and leads to consistently improved fidelity—even under the same evaluation transformation.

Table 7: Comparison of reconstruction fidelity across different channel processing strategies using a frozen VAE.

Channel Processing	MSE ↓
RGB	7.13×10^{-4}
Organelle-aware	4.93×10^{-5}
Organelle-aware + RGB	4.06×10^{-4}

Channel-wise encoding. Our use of a frozen RGB VAE involves independently encoding and decoding the six fluorescence channels, while their concatenated latents are modeled jointly by the diffusion transformer. Thus, although the VAE itself does not explicitly model cross-channel dependencies, the joint diffusion process can capture interactions between fluorescence channels. The reconstruction results in Table 7 support the effectiveness of this design, while a dedicated multi-channel VAE remains an alternative for future investigation.

F Generation Variance Experiment

The qualitative analysis done on the feature representations of the generated vs real images (Figure 7) shows the spread on the principal components which shows meaningful variance across conditioned classes. Additionally, we conducted an feature variance analysis (on HUVEC, perturbations 1108, 1124, 1137 and 1138) where we extracted the features using OpenPhenom for both real and generated samples and measured their variances to check if the feature variances distribute similarly to the real data.

Table 8: Feature variance analysis comparing features extracted by real and generated images.

Treatment	Real Variance	Gen Variance	Difference
p1108	0.0171	0.0144	−0.0027
p1124	0.0145	0.0155	+0.0010
p1137	0.0173	0.0192	+0.0019
p1138	0.0129	0.0159	+0.0030

Takeaway. It can be seen from the Table 8 the generated variance is similar to (or even slightly higher) than the one of real data, indicating that our model preserves diversity and does not exhibit mode collapse.

G Comparison with Cellflux

CellFlux [36] introduces a generative framework that treats cellular morphology prediction as a distribution-to-distribution mapping from same-batch control images to perturbed cells, solved via conditional flow matching. Evaluated on BBBC021 (chemical), RxRx1 (genetic), and JUMP (combined) datasets, CellFlux reports substantially lower FID/KID following IMPA's evaluation pipeline.

Despite its strong results, like IMPA, it focuses on low-resolution ($6 \times 96 \times 96$) crops and a single cell type (U2OS), whereas MorphGen synthesizes native-resolution $6 \times 512 \times 512$ fields across all four RxRx1 cell types. To make CellFlux comparable with MorphGen, we adopt two strategies:

1. **Relative FID**. We report the ratio $\mathrm{FID}(\mathrm{gen}, \mathrm{real})/\mathrm{FID}(\mathrm{real}_A, \mathrm{real}_B)$, which normalizes target-distribution shifts due to IMPA's preprocessing and differing resolutions (lower is better; values near 1 indicate parity).
2. **IMPA-matched evaluation**. Ee extract 96×96 nuclei-centered crops from MorphGen outputs using Otsu thresholding and compare against IMPA's preprocessed real images. This deliberately conservative protocol disadvantages MorphGen, which was trained on full-resolution, channel-normalized images with different illumination statistics.

G.1 Relative FID comparison

Table 9: Relative FID comparison across *different* target distributions. MorphGen is evaluated on 96×96 crops from native, channel-normalized data; CellFlux uses IMPA's pipeline (U2OS, 96×96).

Method	FID (gen–real)	FID (real–real)	Relative FID
CellFlux	168.797 ± 0.868	80.712 ± 0.444	2.091 ± 0.021
MorphGen	205.470 ± 2.242	144.743 ± 4.608	1.417 ± 0.052

Using Relative FID within the same target pipeline, we normalize away preprocessing and resolution effects. Table 9 shows that MorphGen is significantly closer to its own target distribution than CellFlux is to IMPA's (1.417 ± 0.052 vs. 2.091 ± 0.021. Because absolute FIDs are tied to the chosen target distribution and preprocessing pipeline, they are not directly comparable across methods; Relative FID corrects for this dependency and thus provides a fair basis for cross-pipeline comparison.

G.2 IMPA-matched evaluation

Table 10: CellFlux vs. MorphGen under IMPA-matched evaluation. Means ± 95% CI over n=3 seeds. Lower is better.

Method	FID (gen–real)	FID (real–real)	Relative FID
CellFlux	168.797 ± 0.868	80.712 ± 0.444	2.091 ± 0.021
MorphGen	182.970 ± 0.464	80.660 ± 0.716	2.260 ± 0.025

Under the analysis (IMPA-style 96×96 U2OS crops) that disadvantages MorphGen -trained on native-resolution 6×512×512 channel-normalized images—the methods are close. The real-real baselines are effectively identical (80.712 ± 0.444 vs. 80.660 ± 0.716), confirming matched targets, and MorphGen's Relative FID is within single-digit percent of CellFlux (2.260 vs. 2.091). This suggests MorphGen remains competitive outside its native resolution/preprocessing regime while additionally supporting multi–cell-type, full-resolution synthesis.

Takeaway. Using Relative FID, we enable a fair cross-pipeline comparison. Results show that even when focusing only on the zoomed 96×96 crops of our higher-resolution images, MorphGen is significantly closer to its own target distribution—indicating sensitivity to fine-grained detail at native resolution. Furthermore, when evaluated against a different target distribution using the IMPA pipeline for full comparability, MorphGen performs on par with CellFlux despite this deliberately disadvantaged setting.

H Matched MorphoDiff Comparison and Architectural Controls

Evaluation protocol. To further assess baseline comparability, we independently evaluate the publicly released MorphoDiff checkpoint on RxRx1/HUVEC and introduce controlled architectural variants. All models are evaluated using the same three seed-specific sets of 50 randomly sampled perturbations. We compute FID after applying MorphoDiff's RGB colormap to the real and generated

Table 11: Matched comparison on RxRx1/HUVEC using the same evaluation protocol across three seeds and 50 perturbations per seed. Lower FID is better. Initialization refers to the diffusion backbone.

Method	Backbone	Training representation	Initialization	FID ↓
MorphGen	SiT-XL/2 + REPA	6-channel	Scratch	50.6 ± 2.3
MorphoDiff (released)	U-Net	RGB	SD-pretrained	91.2 ± 3.3
MorphoDiff-SiT	SiT-XL/2	RGB	Scratch	107.7 ± 2.0
MorphoDiff-SiT (4 cell types)	SiT-XL/2	RGB	Scratch	106.7 ± 1.8

images, ensuring a common evaluation space. The reported confidence intervals reflect variability across evaluation seeds, including perturbation sampling and generation.

Evaluation space considerations. For comparability with MorphoDiff, the matched evaluation computes FID after mapping six-channel images into RGB space. Consequently, this metric primarily assesses image fidelity under the common RGB representation and does not directly measure the fidelity of the complete six-channel joint distribution. We therefore complement RGB-based comparisons with channel-wise fidelity measurements and morphology-based evaluations using the native fluorescence channels.

Released MorphoDiff checkpoint. As shown in Table 11, MorphGen reduces FID from 91.2 ± 3.3 to 50.6 ± 2.3 compared with the released MorphoDiff checkpoint. MorphoDiff's originally reported FID of 78 is retained in Table 1 for comparison with the published benchmarks. Here, we report the independently evaluated checkpoint under the same perturbation sampling and evaluation protocol as MorphGen.

Architectural controls. To investigate the contribution of the diffusion backbone, we replace MorphoDiff's U-Net with SiT-XL/2 while retaining its scGPT conditioning, diffusion objective, RGB representation, training data, and published training pipeline. The SiT backbone is initialized from scratch, matching MorphGen's diffusion-backbone initialization. This variant obtains an FID of 107.7 ± 2.0, without improving over the original MorphoDiff architecture.

Our results suggest that MorphoDiff does not benefit from replacing its U-Net with the more recent SiT architecture, potentially because the original U-Net benefits from Stable Diffusion pretraining.

Training-data scope. We additionally train MorphoDiff-SiT on all four RxRx1 cell types, introducing a cell-type embedding while retaining its RGB representation and scGPT conditioning. The resulting HUVEC FID of 106.7 ± 1.8 is comparable to the HUVEC-only variant (107.7 ± 2.0), indicating that expanding the training data to multiple cell types does not substantially improve its HUVEC generation fidelity.

Overall, these controls indicate that neither replacing the diffusion backbone with SiT nor expanding the training data to four cell types is sufficient to reproduce MorphGen's performance.

Computational cost. We additionally compare the computational requirements of MorphGen and MorphoDiff in Table 12. Although MorphGen has fewer parameters, its larger latent representation and six independent VAE decoding passes introduce additional computational overhead compared to MorphoDiff, which decodes a single RGB image. However, these operations can be parallelized across fluorescence channels to mitigate the decoding overhead.

Table 12: Computational comparison of MorphGen and MorphoDiff. FLOPs-based estimates are reported for the inference setting.

Method	Params	GFLOPs/step	VAE calls	TFLOPs/image
MorphGen	687.6M	468	6	30.9
MorphoDiff	859.5M	339	1	18.2

I Performance in low-sample regimes

To complement our analysis on the frequently observed perturbations (p1108, p1124, p1137, p1138), we also evaluate MorphGen's performance in a low-sample regime. We report the class-conditional FIDs with only 16 samples per perturbation in Table 13.

Table 13: Class-conditional FID with 16 samples per set.

Perturbation ID	Num. Samples	FID (real–real)	FID (gen–real)
789	16	105.55	112.65
862	16	105.39	120.66
83	16	121.65	127.53
531	16	104.33	109.25
1048	16	119.61	112.55

Note: With only 16 images per set, class-conditional FID can fluctuate, so some gen–real scores may appear marginally lower (e.g., perturbation 1048) than the corresponding real–real baselines.

Takeaway. Results show that FIDs between generated and real sets closely match those computed between independent, mutually exclusive real subsets, highlighting MorphGen's generation quality even under data scarcity.

J Analysis of downstream performance on the conditioning factors

Beyond generative quality, MorphGen can populate balanced datasets for downstream tasks such as classification, clustering, and causal discovery. To assess the downstream utility and the biologically meaningful distinctions captured by our generator, we conducted the following experiments.

J.1 OpenPhenom Features

We extracted 384-dimensional OpenPhenom embeddings from both real Cell Painting images and MorphGen-generated images using the frozen OpenPhenom encoder. A simple logistic regression probe was then trained under three train $\rightarrow$ test regimes:

- **Real** $\rightarrow$ **Real**: baseline setting where both train and test data are real.
- **Generated** $\rightarrow$ **Generated**: both train and test sets are synthetic MorphGen images.
- **Generated** $\rightarrow$ **Real**: cross-domain setting where the probe is trained on synthetic images and evaluated on held-out real images.

We evaluated the probe on two prediction tasks—cell type (4 classes) and perturbation ID (4 classes)—reporting test accuracy as well as 5-fold cross-validation (CV) mean and standard deviation on the training split.

Table 14: Linear probe experiment with OpenPhenom features on the most frequent 4 perturbations: 1108, 1124, 1137 and 1138.

Train $\rightarrow$ Test	Task	Test Acc.	CV Mean	CV Std	# Train	# Test
Real $\rightarrow$ Real	Cell-type	0.997	0.992	0.0049	760	327
	Perturbation	0.838	0.845	0.0245	760	327
Generated $\rightarrow$ Generated	Cell-type	0.994	0.999	0.0024	840	360
	Perturbation	0.858	0.838	0.0208	840	360
Generated $\rightarrow$ Real	Cell-type	0.983	0.998	0.0020	1200	1087
	Perturbation	0.804	0.848	0.0308	1200	1087

Takeaway. OpenPhenom features enable near-perfect cell-type classification and robust perturbation recognition even when the probe is trained solely on MorphGen images and evaluated on real data, indicating that the generator preserves biologically relevant signal.

J.2 CellProfiler Features

We extracted morphological features for HUVEC cells for the four most frequent perturbations (1108, 1124, 1137, 1138) using a standard CellProfiler pipeline (modules such as IdentifyPrimary/Secondary/TertiaryObjects, MeasureGranularity, MeasureObjectIntensity/Neighbors/SizeShape, MeasureTexture, MeasureImageAreaOccupied and MeasureImageIntensity). Starting from the full feature set (2533 features), we applied the following preprocessing: (i) z-score standardization, (ii) variance thresholding at 1e-5 to remove near-constant features and (iii) correlation filtering at $|r| \geq 0.7$ to eliminate redundancy. This reduced the feature space to 878 and then 70 features, respectively. We then applied PCA and kept 32 components.

Similar to the OpenPhenom setting, we trained a linear classifier to predict perturbation ID (4 classes) under the same three regimes. We report the accuracy and macro-F1 scores in Table 15.

Table 15: Downstream perturbation classification with CellProfiler features (HUVEC, 4 perturbations: 1108, 1124, 1137 and 1138)

Train → Test	**Accuracy**	**Macro-F1**
Real → Real	0.794	0.792
Generated → Generated	0.873	0.873
Generated → Real	0.739	0.734

It can be seen in Table 15 that Generated→Generated yields the strongest scores (accuracy ~ 0.87), confirming that MorphGen produces internally consistent samples with clear perturbation separability. In contrast, Real→Real shows lower scores (~ 0.79) and visibly higher variability, which is expected given natural measurement noise and biological heterogeneity in real data. Importantly, Generated→Real performs *close to* Real→Real (0.74 vs. 0.79), indicating that classifiers trained on generated images recover perturbation-discriminative structure that aligns well with the real measurements.

Takeaway. Overall, this supports MorphGen's success at capturing biologically meaningful, perturbation-specific signal rather than mere visual plausibility.

K Benefit of synthetic data for downstream classification

To further examine the utility of MorphGen for downstream tasks, we tested whether augmenting training sets with synthetic images can improve perturbation classification performance. We extended the probe from the 4 most frequent perturbations to the 10 frequent ones, thereby increasing phenotypic similarity between classes and making the classification problem more challenging. This setting provides a more demanding benchmark for evaluating the benefit of synthetic augmentation.

K.1 OpenPhenom Features

We trained a logistic regression probe on OpenPhenom embeddings under four regimes:

- **Gen → Gen**: both train and test sets consist of MorphGen images (sanity check).
- **Gen → Real**: train only on synthetic images, test on real images.
- **Real → Real**: baseline with real images only.
- **Real+Gen → Real**: same as Real → Real, but with the training set augmented by synthetic images.

In this harder 10-class setting, the Real → Real baseline drops from 83.8% to 67.7%. Augmenting the training set with MorphGen images increases accuracy by +3.6 percentage points ($67.7\% \rightarrow 71.2\%$). Six out of ten classes improve, three remain unchanged, and one decreases slightly. Training exclusively on synthetic images still achieves 54% accuracy, showing that MorphGen-generated data carry discriminative morphological signal despite the domain gap.

Takeaway. Synthetic augmentation with MorphGen improves real-data classification accuracy, while training on generated images alone still yields features that generalize well to real images.

Table 16: Linear probe accuracy on the 10 frequent perturbations using OpenPhenom features.

Perturbation	Gen → Gen	Gen → Real	Real+Gen → Real	Real → Real
1108	0.633	0.607	0.706	0.706
1109	0.813	0.821	0.813	0.813
1110	0.800	0.375	0.647	0.588
1111	1.000	0.750	0.882	0.824
1112	0.808	0.250	0.588	0.529
1113	0.849	0.964	0.824	0.765
1114	0.800	0.429	0.647	0.588
1115	0.813	0.536	0.492	0.529
1119	0.735	0.196	0.625	0.625
1138	0.758	0.470	0.900	0.800
Average	0.801	0.540	0.712	0.677

K.2 ResNet18 Features

Although OpenPhenom embeddings are used only during training for alignment, evaluating downstream performance on the same representation space can raise concerns about circularity. To address this, we repeated the probe experiment using a ResNet18 feature extractor trained independently of MorphGen.

Table 17: Linear probe accuracy on the 10 frequent perturbations using ResNet18 features.

Perturbation ID	Real → Real	Gen → Real	Real+Gen → Real
1108	0.529	0.353	0.353
1109	0.563	0.563	0.500
1110	0.294	0.353	0.353
1111	0.529	0.765	0.706
1112	0.471	0.235	0.529
1113	0.412	0.412	0.529
1114	0.588	0.294	0.529
1115	0.294	0.412	0.412
1119	0.375	0.250	0.375
1138	0.500	0.600	0.700
Average	**0.456**	**0.424**	**0.499**

When evaluated on ResNet18 features, the Real → Real baseline reaches 45.6% accuracy, and augmenting with synthetic images improves it to 49.9%. Training exclusively on generated data achieves 42.4%. Despite overall lower absolute accuracies compared to OpenPhenom (reflecting weaker features), the relative trends remain consistent: synthetic augmentation provides a measurable gain, and generated data alone still capture class-discriminative signal.

Takeaway. This confirms that the observed benefits are not an artifact of the OpenPhenom feature space.

Table 18: Perturbation-level phenotypic signature preservation. For each perturbation, we compute the 70-dim delta-feature vector (treatment minus control, p1138) for both real and generated HUVEC images, and report Pearson correlation between these vectors.

Perturbation	Corr. ↑	p-value	n_{real}	n_{gen}
1137	0.922121	9.44×10^{-30}	126	233
1108	0.895998	1.14×10^{-25}	130	250
1124	0.834478	2.89×10^{-19}	127	243

Table 19: **Feature-level delta agreement.** Top and bottom CellProfiler features ranked by correlation between real and generated delta-values across perturbations (1108, 1124, 1137) relative to control (1138). We also report the mean absolute delta magnitude for real and generated images.

Feature	Corr. ↑	p-value	$\lvert\Delta\rvert_{\text{real}}$	$\lvert\Delta\rvert_{\text{gen}}$
Top 10 features by correlation				
StDev_Cytoplasm_Texture_Correlation_OrigGray_5_03_256	0.999914	0.008358	0.615383	0.385262
StDev_Cells_AreaShape_FormFactor	0.999843	0.011268	0.550508	0.436651
Mean_Cells_AreaShape_Area	0.999537	0.019373	0.603212	0.672867
StDev_Nuclei_Texture_Correlation_OrigGray_10_00_256	0.999419	0.021701	0.730174	0.649439
AreaOccupied_AreaOccupied_Cells	0.999331	0.023281	1.000485	0.959291
Bottom 10 features by correlation				
Granularity_13_OrigGray	0.235079	0.848931	0.682751	0.435022
Median_Nuclei_Granularity_15_OrigGray	-0.395548	0.741110	0.135087	0.164899
ExecutionTime_24OverlayOutlines	-0.426113	0.719765	0.391609	0.364377
ExecutionTime_23MeasureImageIntensity	-0.626717	0.568796	0.345231	0.294323
ExecutionTime_06MeasureImageQuality	-0.989645	0.091696	0.355497	0.256092

L CellProfiler Delta-Feature Agreement

Delta-feature correlation against control. While PCA overlap and correlation matrices (Section R, Figures 3 and 4) assess distributional alignment and feature relationships, they do not directly test whether the generator preserves *perturbation-induced effects* relative to the control. To quantify this, we evaluate whether MorphGen reproduces both the direction and magnitude of morphological changes in CellProfiler feature space.

We focus on the same four most common perturbations in HUVEC cells: 1138 (control) and treatments 1108, 1124, 1137. For each CellProfiler feature f and perturbation p, we compute the treatment effect as a *delta-feature* relative to control:

$$\Delta_f^{\text{CP}}(p) = \mathbb{E}[f \mid P = p,\, ct = \text{HUVEC}] - \mathbb{E}[f \mid P = \text{control},\, ct = \text{HUVEC}]\,.$$

separately for real and generated images. We then compute the correlation between the real and generated delta-values across treatments for each feature, which measures whether MorphGen preserves perturbation effects rather than only matching marginal feature statistics.

Feature-level results. Overall agreement is high (mean correlation 0.794, median correlation 0.953), with 51/70 features achieving correlation > 0.8 and 64/70 exceeding 0.5. Table 19 reports the top and bottom correlated features. The most consistent features correspond to biologically meaningful texture and shape descriptors (e.g., texture correlation statistics, cell area/shape, granularity), supporting that the model reproduces genuine phenotypic changes. In contrast, low-correlation features are dominated by noisy or non-biological measurements (e.g., execution-time features), indicating that disagreements concentrate on unreliable signals rather than systematic morphological artifacts.

Perturbation-level signature agreement. Finally, we form a delta-feature vector per perturbation and compute the correlation between the real and generated vectors. As shown in Table 18, all treatments exhibit strong agreement, indicating that MorphGen preserves the overall phenotypic cues induced by each perturbation.

M Virtual Instrument Experiment Details

We provide additional quantitative evidence supporting the compositional generalization findings reported in the main text. Recall that *seen* and *unseen* refer to whether the compositional model was trained on the corresponding (HUVEC, p) pair, while the oracle model observes all pairs by construction.

Oracle and compositional model performances. Table 22 summarizes the per perturbation rFID gap $\Delta(p) = \text{rFID}_{\text{comp}}(p) - \text{rFID}_{\text{oracle}}(p)$ on HUVEC. On seen pairs, the gap is small (median 0.014, mean 0.015), indicating that compositional model closely matches the oracle quality in-distribution.

Table 20: Treatment-effect agreement on unseen HUVEC compositions with reproducible real effects. We report median cosine similarities between generated and real effects, alongside the real-to-real reference.

Metric	OpenPhenom	CellProfiler
Reproducible perturbations	46%	44%
Generated vs. real (cosine)	0.85	0.66
Real vs. real (cosine)	0.97	0.84
Agreement recovered	88%	78%

On unseen compositions, the gap is larger on average (median 0.222, mean 0.246), yet a majority of held-out pairs satisfy the seen-calibrated oracle-level tolerance (accepted on 55% at $Q_{0.95}$).

Agreement in rFID trends across perturbations. Beyond the gap magnitude, we measure whether oracle and compositional models agree on the perturbation difficulty trends. On seen pairs, oracle vs compositional rFID shows statistically significant monotonic agreement (Spearman $\rho = 0.417$, $p = 1.9 \times 10^{-16}$), suggesting that perturbation that are easy or hard for the oracle tend to remain consistent for the compositional model on the in-distribution pairs. Importantly, this agreement persists on unseen compositions, where we still observe a significant correlation (Spearman $\rho = 0.328$, $p = 8 \times 10^{-4}$). Together, these results indicate that while the generalization gap grows on unseen combinations, the compositional model retains a meaningful alignment with oracle difficulty structure.

Diagnostic: effect of cross cell-type exposure. To probe when compositional transfer succeeds, we correlate the rFID gap on unseen pairs with the amount of training exposure of each perturbation outside HUVEC. We observe a weak but significant negative correlation (Spearman $\rho = -0.258$, $p = 9.2 \times 10^{-3}$), indicating that perturbations supported by more data in other cell types tend to generalize better to the unseen HUVEC. This is consistent with the intended compositional mechanism: transfer quality improves when the perturbation is observed across diverse cellular contexts.

M.1 Biological Feature Validation on Unseen Compositions

We further evaluate whether MorphGen preserves biologically meaningful perturbation effects on the 100 held-out HUVEC compositions. We conduct complementary analyses in OpenPhenom and CellProfiler feature spaces, measuring treatment-effect agreement and perturbation classification performance.

Treatment effect agreement. We first identify perturbations with reproducible effects in real images. Specifically, for each perturbation, we split the real images into two disjoint subsets and independently estimate the morphological effect relative to controls. We consider an effect reproducible when the two estimates agree more closely than effects from different perturbations. Under this criterion, 46% and 44% of the held-out perturbations exhibit reproducible effects in OpenPhenom and CellProfiler feature spaces, respectively.

We then compare the treatment-effect vectors estimated from generated and real images using cosine similarity. Table 20 summarizes the results.

For perturbations with detectable real effects, MorphGen recovers 88% and 78% of the real-to-real agreement in OpenPhenom and CellProfiler spaces, respectively. The agreement across both feature spaces provides complementary evidence that the generated images preserve perturbation-specific morphological information.

Perturbation classification. As an additional validation, we train perturbation classifiers using only real images and evaluate their performance on generated samples from unseen compositions. We restrict the evaluation to perturbations with reproducible real effects, yielding 46 classes in OpenPhenom and 44 classes in CellProfiler space.

Table 21: Perturbation classification on generated images from unseen HUVEC compositions using classifiers trained on real images. The real-data reference reports classification performance on real images.

Features	Classes	Top-1	Top-5	Chance	Real reference
OpenPhenom	46	0.21	0.53	0.02	0.63
CellProfiler	44	0.22	0.54	0.02	0.70

Table 22: Summary of the rFID gap $\Delta(p) = \mathrm{rFID}_{\mathrm{comp}}(p) - \mathrm{rFID}_{\mathrm{oracle}}(p)$ between the compositional and oracle models on HUVEC. Smaller values indicate closer oracle-level fidelity.

Split	n	Median Δ	Mean Δ
Seen (HUVEC, trained)	300	0.014	0.015
Unseen (HUVEC, held-out)	100	0.222	0.246

As shown in Table 21, classifiers trained on real images achieve top-1 accuracies of 0.21 and 0.22 on unseen generations. This further supports the preservation of perturbation specific information, although a gap remains relative to real image classification.

Relationship between rFID and biological agreement. Finally, we examine whether the seen-calibrated rFID criterion also captures biological consistency. Among held-out pairs accepted by the rFID criterion, 13% still show disagreement with the real morphological-effect direction. Thus, satisfying the real calibrated image quality threshold does not necessarily imply that the generated perturbation effect is correct.

Nevertheless, rFID remains informative for identifying severe failures. Cases with strongly incorrect effect directions and exaggerated effect magnitudes (approximately twice the real effect) are largely rejected by the rFID criterion.

These findings suggest that rFID and morphological effect agreement capture complementary aspects of compositional generalization. While MorphGen preserves meaningful biological features for many unseen compositions, image fidelity and biological consistency should be evaluated separately.

N Organelle Specific Full Results

Table 23 reports the same metrics as Table 2, with the addition of 95% confidence intervals.

Table 23: FID and KID scores (mean $\pm$ 95% CI) for 50 random perturbations across all cell types. Our method supports generation for all four cell types (HEPG2, HUVEC, RPE, U2OS) and provides channel-wise control.

Channel	FID $\downarrow$	KID $\downarrow$	Rel. FID $\downarrow$
RGB	50.2 ± 1.3	0.0082 ± 0.0003	1.411 ± 0.085
Nucleus	27.6 ± 0.6	0.0101 ± 0.0008	1.691 ± 0.206
ER	48.1 ± 0.4	0.0116 ± 0.0008	1.774 ± 0.128
Actin	57.6 ± 1.2	0.0155 ± 0.0003	1.562 ± 0.032
Nucleolus	49.6 ± 0.4	0.0132 ± 0.0005	1.612 ± 0.092
Mito	43.6 ± 0.4	0.0123 ± 0.0009	1.489 ± 0.099
Golgi	59.0 ± 2.3	0.0121 ± 0.0018	1.564 ± 0.155

O Cell-type Specific Full Results

Table 24 reports the same metrics as Table 3 with the addition of 95% confidence intervals.

Table 24: FID, KID, and Relative FID scores (mean ± 95% CI) across cell types. MorphGen generates high-fidelity images consistently across diverse cell types.

Cell Type	FID ↓	KID ↓	Rel. FID ↓
HEPG2	41.11 ± 2.46	0.016 ± 0.0005	1.529 ± 0.055
HUVEC	28.65 ± 1.98	0.006 ± 0.0003	1.136 ± 0.086
RPE	34.35 ± 1.67	0.007 ± 0.0003	1.185 ± 0.059
U2OS	38.17 ± 2.27	0.017 ± 0.0004	1.492 ± 0.204

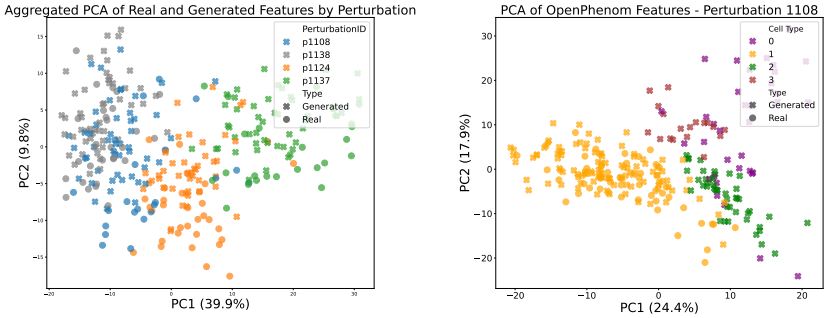


Figure 7: PCA projections of OpenPhenom features from real and generated images. The left panel shows the joint distribution of most frequent perturbations (including the control, p1138) for HUVEC cells, with points colored by perturbation. The right panel visualizes the perturbation 1108 across different cell types. In both panels, marker shapes indicate whether the sample is real or generated.

P OpenPhenom Experiments

OpenPhenom Features. To further assess the biological plausibility of our generated images, we extend our analyses using the microscopy-specific foundation model OpenPhenom. We focus on the same four perturbations in the dataset (1108, 1124, 1137 and 1138) and visualize both real and generated samples in a shared PCA embedding space. The resulting visualizations are shown in Figure 7. Similarly, we observe thatgenerated images reproduce morphology faithfully, showing a strong overlap between real and generated images while preserving perturbation specific deviations. Note that during inference, images are generated without OpenPhenom alignment and processed independently. Moreover, the consistency between Figures 3 and 7 further indicates that OpenPhenom representations are well aligned with CellProfiler features, which was also claimed in the original paper. These patterns suggest that our generative model successfully encodes biologically relevant phenotypic variation while preserving class-level consistency.

Table 25: Conditional Average Treatment Effect (CATE) between control (1138) and perturbed samples, reported per cell type.

Cell Type	p1138 vs p1137			p1138 vs p1108			p1138 vs p1124		
	$\text{CATE}_{\text{real}}$	CATE_{gen}	ΔCATE	$\text{CATE}_{\text{real}}$	CATE_{gen}	ΔCATE	$\text{CATE}_{\text{real}}$	CATE_{gen}	ΔCATE
HEPG2	1.07	1.06	0.01	1.19	0.48	0.71	1.27	0.98	0.29
HUVEC	7.85	7.41	0.44	0.44	0.38	0.06	2.13	2.31	0.18
RPE	1.28	1.09	0.19	1.00	0.65	0.35	1.04	0.83	0.21
U2OS	3.53	2.77	0.76	0.38	0.34	0.04	3.46	2.42	1.04

Table 25 shows the Conditional Average Treatment Effect (CATE) between control (p1138) and perturbed samples, computed using real and generated images across different cell types using OpenPhenom features. Results indicate that images generated by MorphGen preserve treatment-specific cellular features with high fidelity, closely mirroring those from real images. Unsurprisingly, the consistency is strongest for HUVEC cells, likely due to their higher representation in the dataset. The highest treatment effect (i.e., deviation from p1138) in real samples is observed for HUVEC under

treatment p1137, with a CATE of 7.85. MorphGen-generated images closely match this effect with a CATE of 7.41, demonstrating consistency even in cases of strong perturbation response. Overall, the closeness of real and generated CATE values suggests that MorphGen-generated images can support accurate downstream analysis.

Real vs Generated

Figure 8 presents PCA visualizations of OpenPhenom [18] features for four representative perturbations – 1108, 1124, 1137 and the control 1138 – arranged in rows. The left column compares real and generated samples by coloring the points by image type. The strong overlap between real and generated distributions across all perturbations indicates that the generated images faithfully reproduce the morphological feature space of the real data. Whereas the right column colors the same embeddings by cell type, revealing clear separability across cell types. This suggest that OpenPhenom features, even when derived from generated images, retain meaningful cell-type structure and can support downstream analyses such as classification. Together, these results demonstrate that MorphGen produces high-quality, morphologically accurate samples that preserve both perturbation effects and intrinsic cell-type differences.

Figures 9, 10, 11, and 12 show PCA visualizations at the cell-type level (HEPG2, HUVEC, RPE, and U2OS, respectively), allowing a qualitative assessment of color separation across different perturbations. Overall, the results demonstrate that features extracted from MorphGen-generated images exhibit: (i) strong alignment with real features and (ii) clear separability with respect to both cell type and perturbation—the two generative factors we explicitly control.

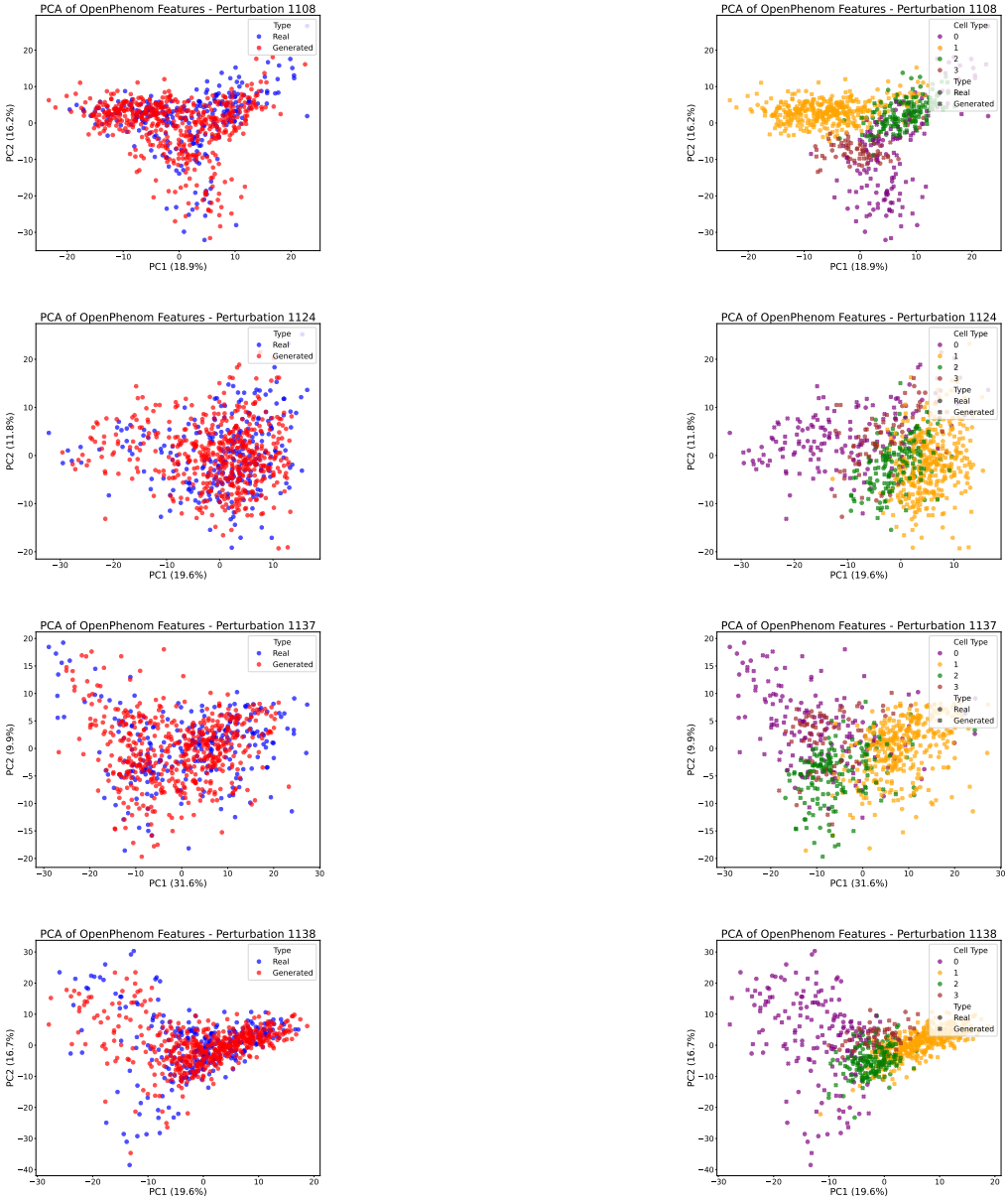
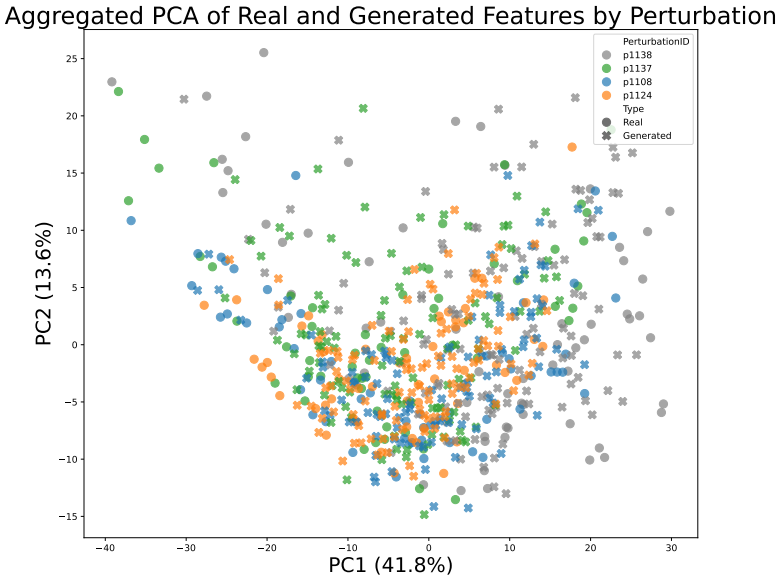
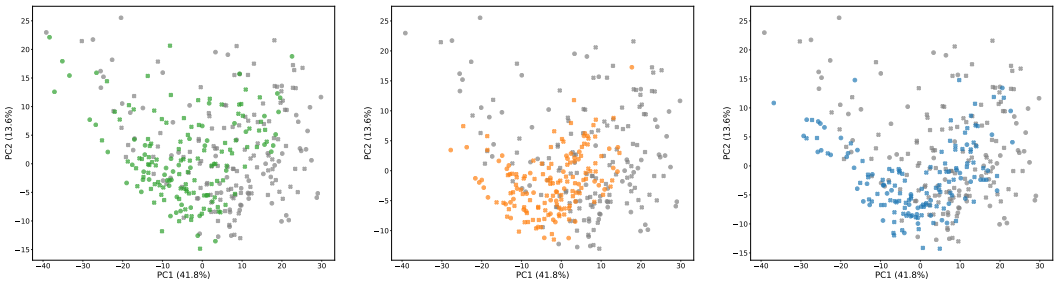


Figure 8: PCA visualizations of OpenPhenom features for four perturbations (rows: 1108, 1124, 1137, and control 1138). Each row compares real and generated embeddings for a single perturbation. *Left column:* points are colored by image type (real vs. generated), revealing strong overlap—indicating that generated images closely match the distribution of real samples. *Right column:* the same embeddings are now colored by cell type, showing that cell-type-specific structure is preserved in both real and generated data.

Perturbation Effects Visualizations HEPG2



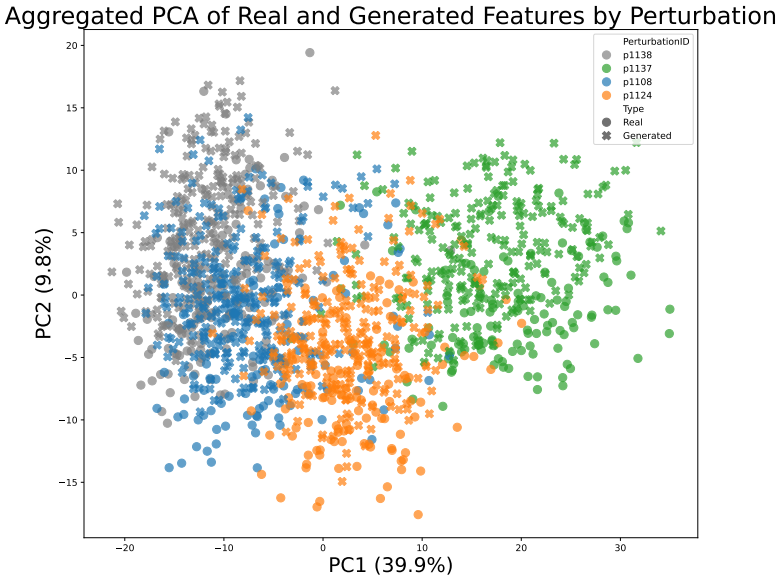
(a) PCA visualization across all four perturbations, including the control (1138).



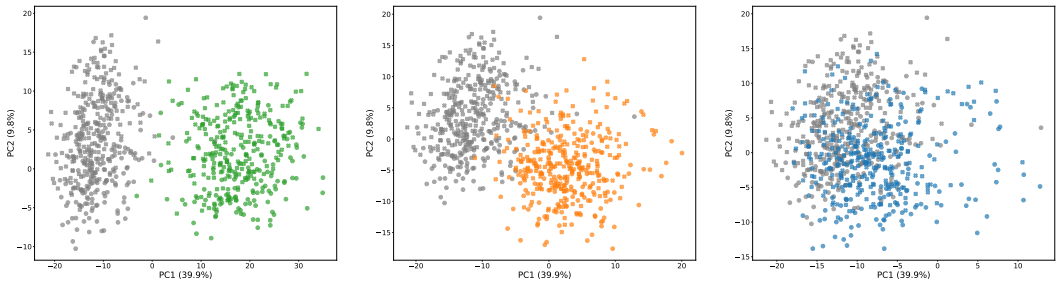
1138 vs 1137 1138 vs 1124 1138 vs 1108

Figure 9: PCA projections of phenotypic embeddings of HEPG2 cells. The top panel shows global variation across all perturbations. The bottom panels show focused pairwise comparisons between the control (1138) and specific perturbations.

Perturbation Effects Visualizatons HUVEC



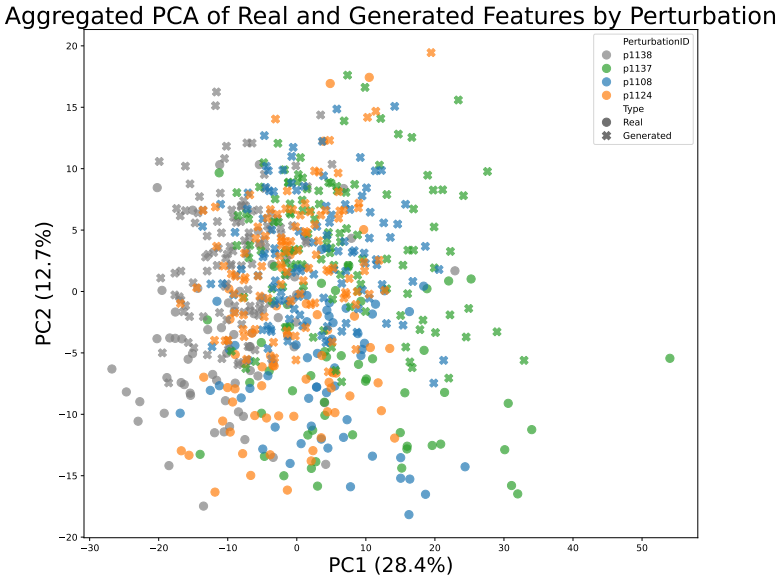
(a) PCA visualization across all four perturbations, including the control (1138).



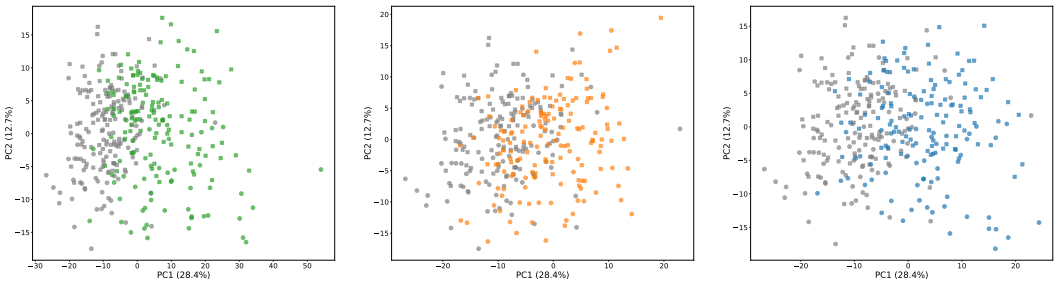
1138 vs 1137 1138 vs 1124 1138 vs 1108

Figure 10: PCA projections of phenotypic embeddings of HUVEC cells. The top panel shows global variation across all perturbations. The bottom panels show focused pairwise comparisons between the control (1138) and specific perturbations.

Perturbation Effects Visualizations RPE



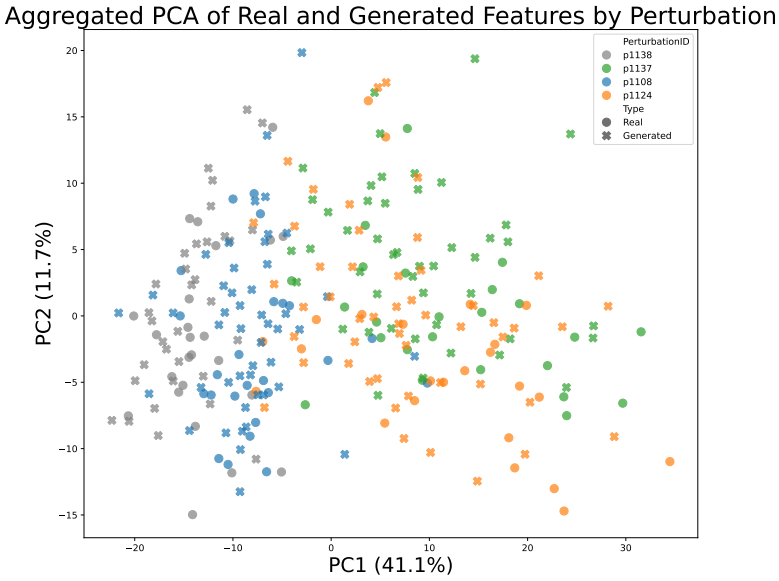
(a) PCA visualization across all four perturbations, including the control (1138).



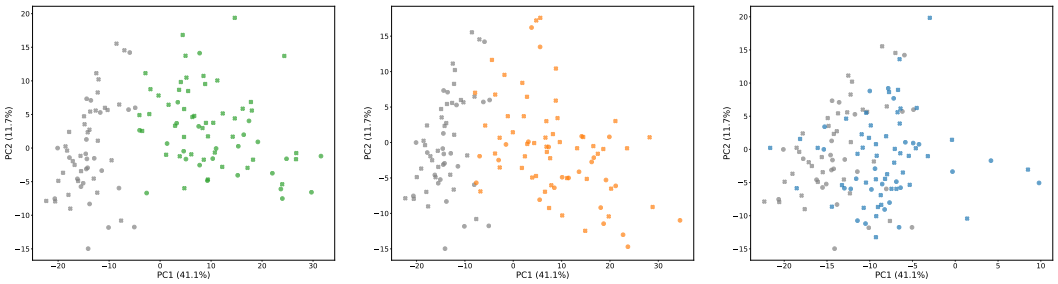
1138 vs 1137 1138 vs 1124 1138 vs 1108

Figure 11: PCA projections of phenotypic embeddings of RPE cells. The top panel shows global variation across all perturbations. The bottom panels show focused pairwise comparisons between the control (1138) and specific perturbations.

Perturbation Effects Visualizations U2OS



(a) PCA visualization across all four perturbations, including the control (1138).



1138 vs 1137 1138 vs 1124 1138 vs 1108

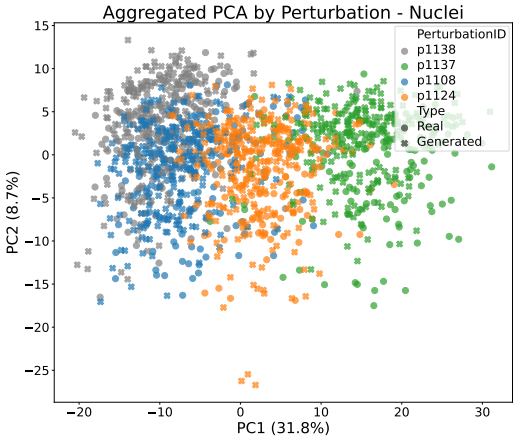
Figure 12: PCA projections of phenotypic embeddings of U2OS cells. The top panel shows global variation across all perturbations. The bottom panels show focused pairwise comparisons between the control (1138) and specific perturbations.

Q Organelle-Specific CATE and Visualizations with OpenPhenom Features

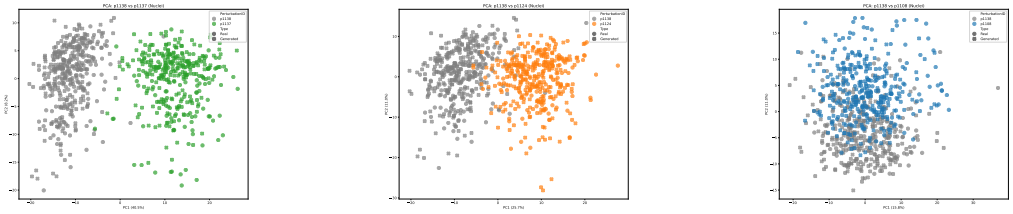
Table 26 reports organelle-specific CATEs, measuring the deviation of perturbations p1108, p1124, and p1137 from the control p1138. Notably, our estimates closely match the real values even at the organelle level, suggesting that MorphGen can accurately capture organelle-specific response patterns. Figure 13 further illustrates this by showing PCA visualizations of Nuclei responses across different perturbations.

Table 26: Conditional Average Treatment Effect (CATE) between control (1138) and perturbed HUVEC cells, reported per organelle.

Organelle	**p1138 vs p1137**			**p1138 vs p1108**			**p1138 vs p1124**		
	$\text{CATE}_{\text{real}}$	CATE_{gen}	ΔCATE	$\text{CATE}_{\text{real}}$	CATE_{gen}	ΔCATE	$\text{CATE}_{\text{real}}$	CATE_{gen}	ΔCATE
Nuclei	9.47	9.50	0.03	0.69	0.59	0.10	2.86	2.97	0.11
ER	9.46	9.54	0.08	0.69	0.60	0.09	2.85	2.97	0.12
Actin	9.49	9.50	0.01	0.69	0.59	0.10	2.86	2.97	0.11
Nucleoli	9.49	9.51	0.02	0.69	0.59	0.10	2.85	2.96	0.11
Mitochondria	9.54	9.51	0.03	0.70	0.59	0.11	2.86	2.97	0.11
Golgi	9.49	9.52	0.03	0.69	0.60	0.09	2.86	2.97	0.11



(a) PCA visualization across all four perturbations, including the control (1138).



1138 vs 1137 1138 vs 1124 1138 vs 1108

Figure 13: PCA projections of Nuclei embeddings of HUVEC cells. The top panel shows global variation across all perturbations. The bottom panels show focused pairwise comparisons between the control (1138) and specific perturbations.

R PCA Visualizations with CellProfiler Features

To further validate the fidelity of generated morphologies, we present PCA visualizations of CellProfiler features for four representative perturbations. Figure 14 shows strong alignment between real and generated samples, highlighting MorphGen's fidelity.

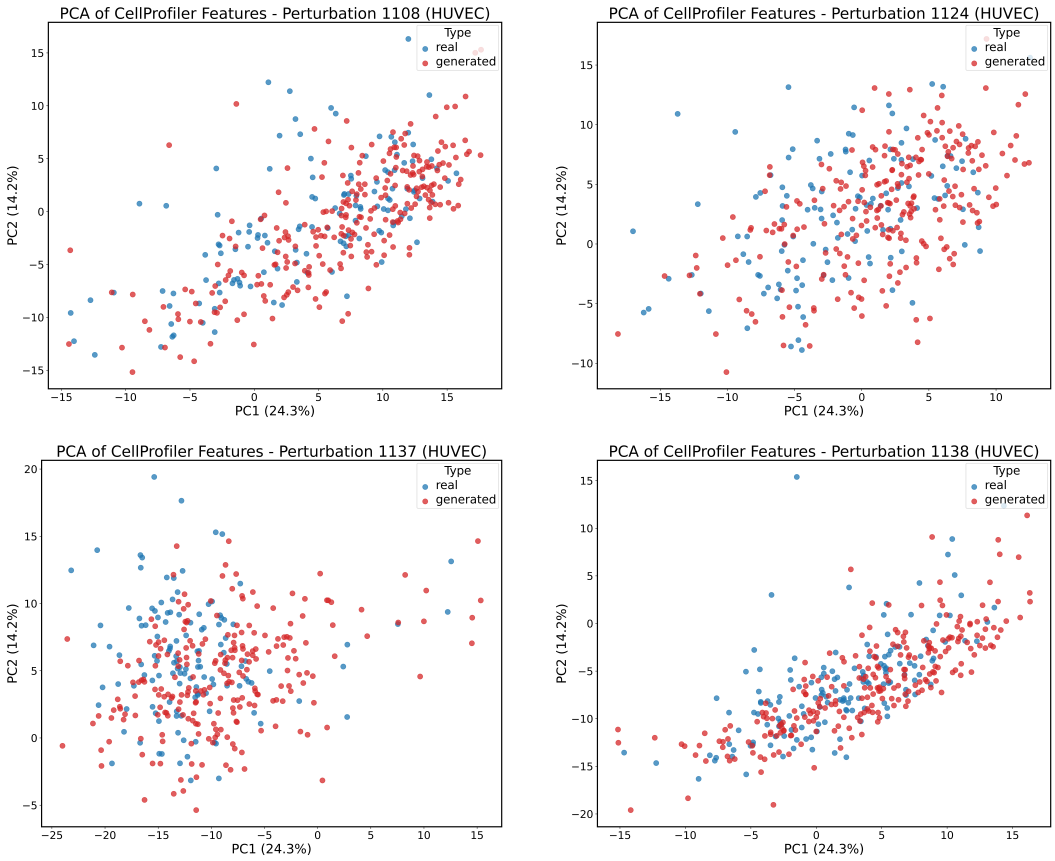


Figure 14: **PCA of CellProfiler features for selected perturbations (HUVEC).** Colors indicate real (blue) and generated (red) samples, the strong overlap highlights that generated images closely match the distribution of real samples.

S Qualitative Examples

In this section, we report additional qualitative evidence of the quality of samples generated with MorphGen. Figures 15, 16, 17 and 18 present representative examples across multiple perturbations and cell types, illustrating that MorphGen consistently produces realistic and diverse cellular morphologies.

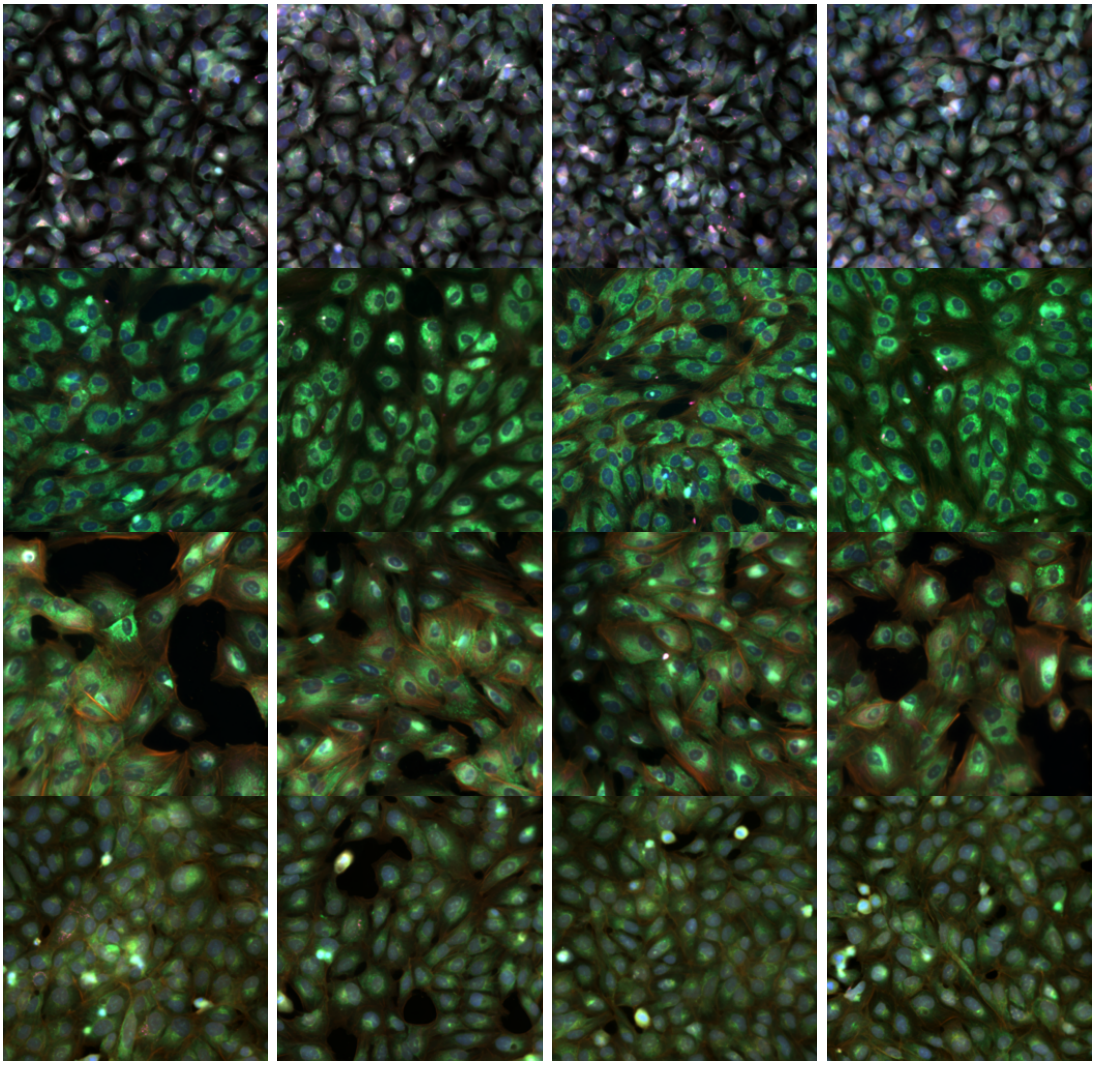


Figure 15: Qualitative examples of generated cell images for perturbation **1108**. Rows correspond to different cell types (HEPG2, HUVEC, RPE, U2OS), and columns show four independent samples per cell type.

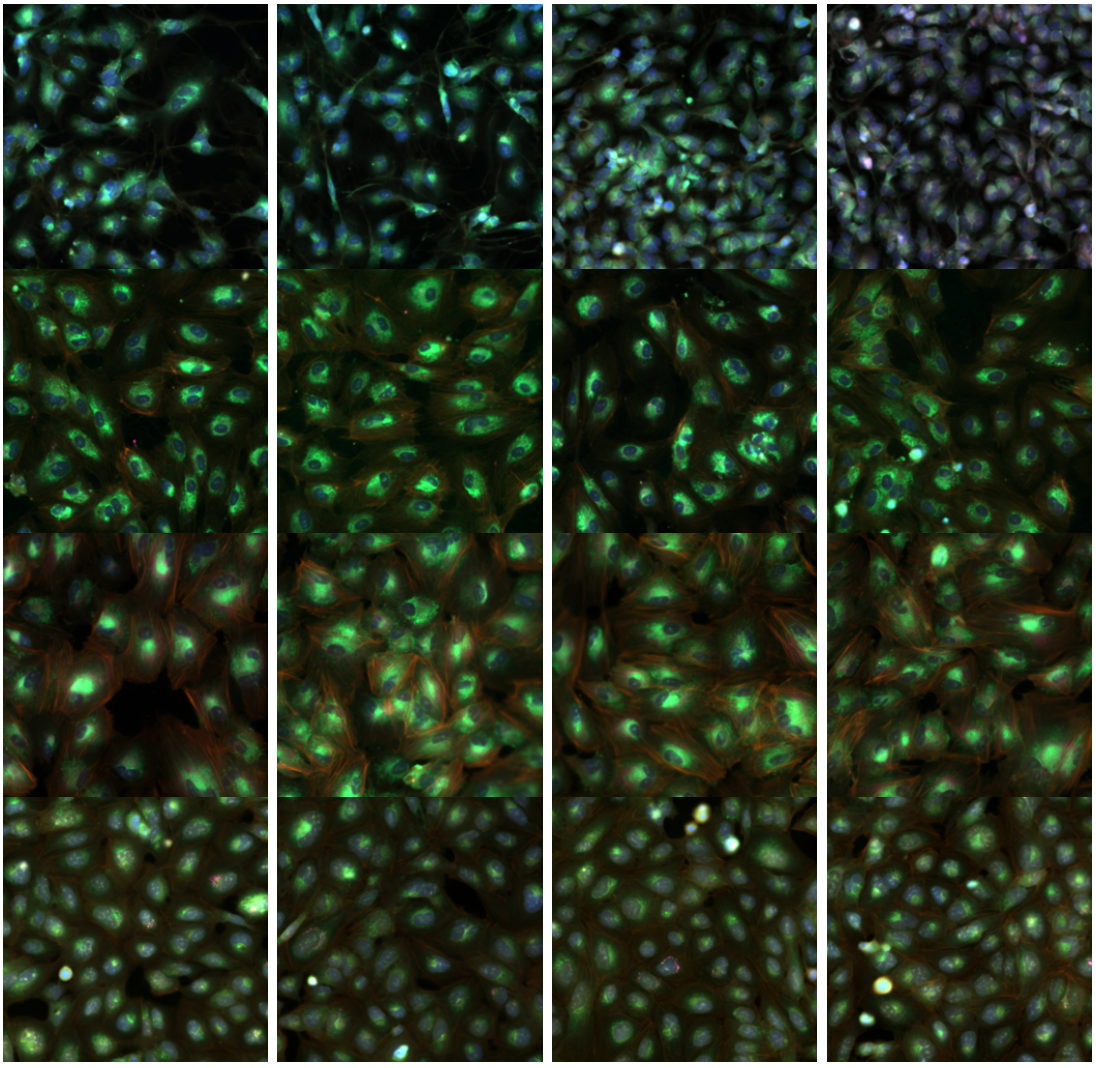


Figure 16: Qualitative examples of generated cell images for perturbation **1124**. Rows correspond to different cell types (HEPG2, HUVEC, RPE, U2OS), and columns show four independent samples per cell type.

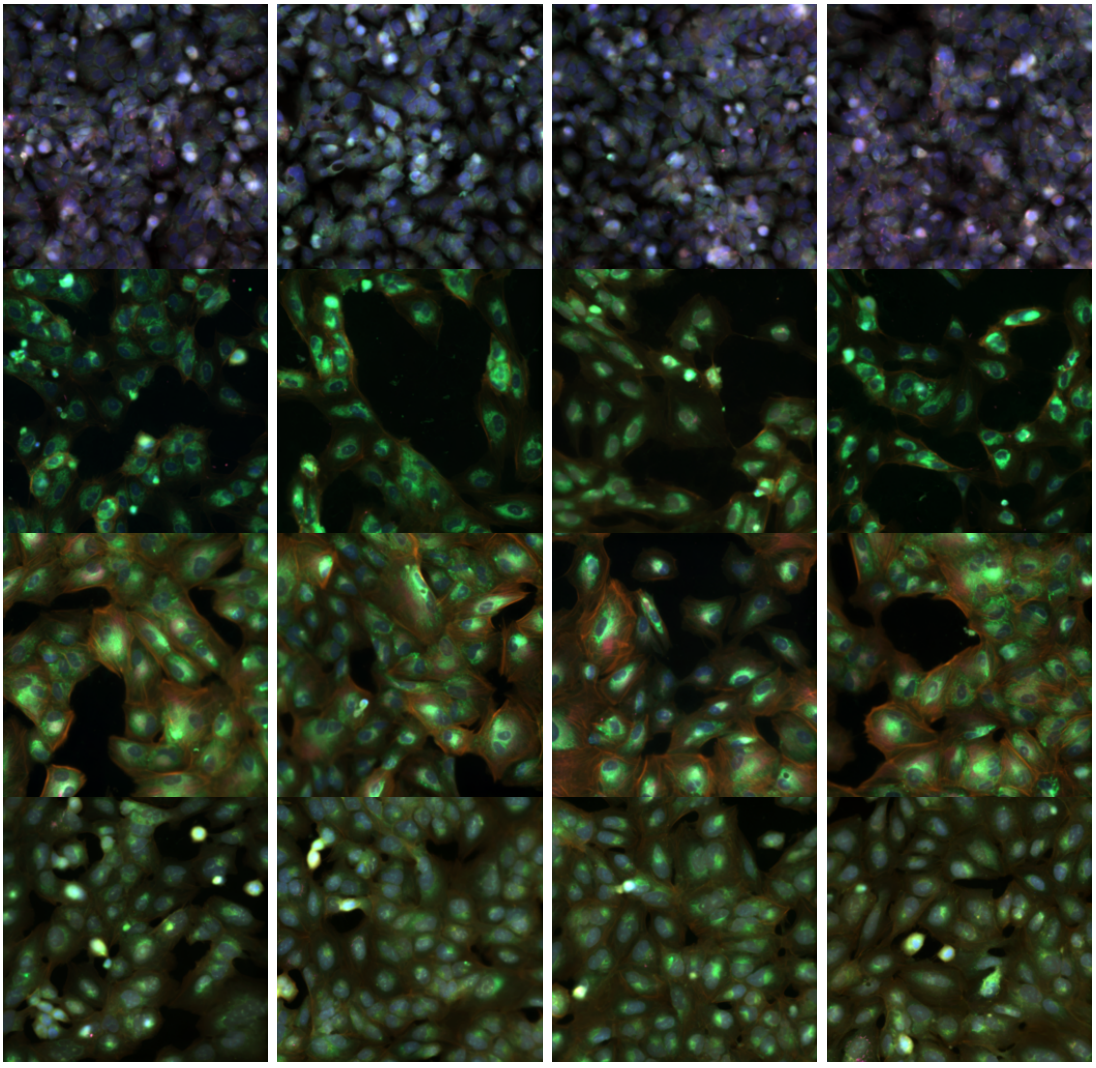


Figure 17: Qualitative examples of generated cell images for perturbation **1137**. Rows correspond to different cell types (HEPG2, HUVEC, RPE, U2OS), and columns show four independent samples per cell type.

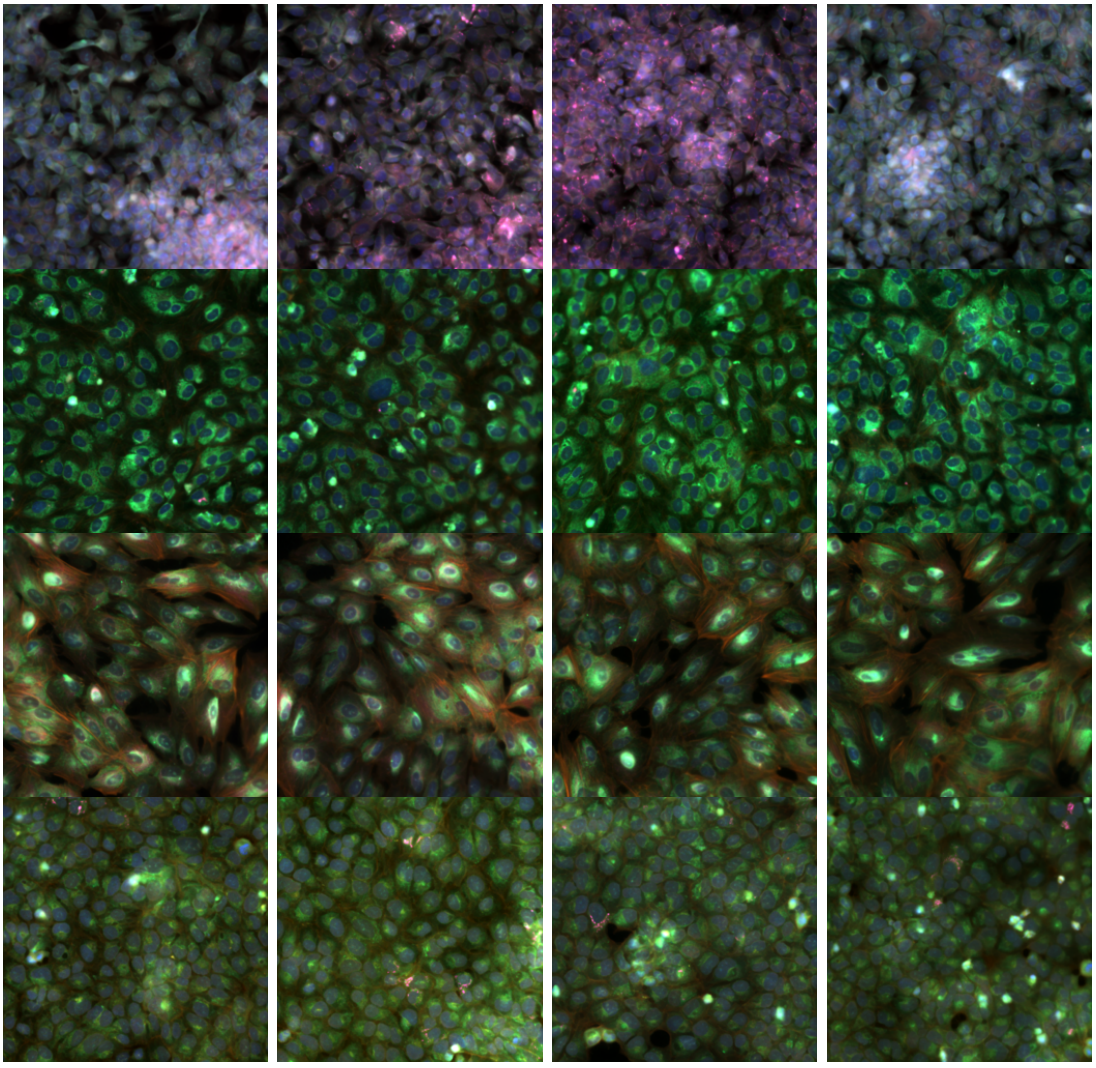


Figure 18: Qualitative examples of generated cell images for perturbation **1138**. Rows correspond to different cell types (HEPG2, HUVEC, RPE, U2OS), and columns show four independent samples per cell type.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Claims made in the abstract are supported through empirical evidence.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations and future work are discussed in the conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include any theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Implementation details and datasets are discussed in the Appendix D. Additionally, supplementary material includes the code that implements the training and evaluation performed in the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We release the code in the supplementary material, and discuss datasets in Appendix C.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix D includes specific training and evaluation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All results produced by the proposed model include confidence intervals when applicable.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources are specified in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: A discussion on ethics and broader impact is provided in Appendix B.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not pose such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper do cite the original owners of assets used in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Training details and the code are released through the Appendix D and the supplementary material.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.