HICON: HIERARCHICAL TECHNICAL CONDITIONING FOR CONTRASTIVE CELL PAINTING BATCH CORRECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Batch effects remain a major challenge in analyzing Cell Painting data. They introduce hierarchical and spatially structured technical variations that are entangled with perturbation-induced biological variations. Existing methods rely on global feature alignment or normalization over predefined batch groups. However, these approaches may fail to capture the heterogeneous technical variations, which can disrupt the biological signals. In this paper, we formulate Cell Painting batch correction as a conditional representation learning problem, aiming to maximize biological information given latent technical states. We propose HICON, a hierarchy-aware technical conditioning framework for contrastive biological representation learning. HICON uses a two-branch architecture to disentangle biological and technical representations. The technical branch learns a hierarchical technical embedding with a confidence-aware mechanism to capture sample-specific technical variation beyond metadata labels. The biological branch leverages perturbation replicates to define positive and negative samples and adaptively reweights the contrastive differentiation between the sampled pairs according to their inferred technical similarity, thereby emphasizing perturbation variation under comparable technical conditions. Extensive experiments across JUMP-CP, BBBC, CPG0036, and RxRx1 show that HICON achieves strong batch correction, preserves biological structure, and improves downstream biological utility.

1 INTRODUCTION

Cell Painting is a widely used high-content imaging assay for profiling cellular responses to genetic and chemical perturbations Taylor (2008); Inglese et al. (2007); Blay et al. (2020); Bray et al. (2016). By capturing morphological phenotypes, it enables large-scale inference of biological relationships Seal et al. (2024) and supports a wide range of downstream analyses, such as mechanism-of-action prediction Trapotsi et al. (2022a); Akbarzadeh et al. (2022); Seal et al. (2022), activity prediction Simm et al. (2018); Hofmarcher et al. (2019) and compound toxicity prediction Seal et al. (2021); Trapotsi et al. (2022b); Gerry et al. (2016). Large-scale Cell Painting resources, such as RxRx3 Fay et al. (2023) and JUMP-CP Chandrasekaran et al. (2023), have been constructed to cover a broad range of compound and genetic perturbations. These datasets are generated across diverse experimental conditions. As a result, the extracted profiles inevitably contain technical variation, commonly referred to as batch effects Arevalo et al. (2024); Scherer (2009); Bray et al. (2016). These effects can distort biological signals and degrade downstream analyses.

To address this issue, various batch correction methods Johnson et al. (2007); Ando et al. (2017); Lopez et al. (2018); Yan et al. (2025) have been proposed to mitigate technical variation while preserving biological information. A recent benchmark study Arevalo et al. (2024) systematically evaluates these existing methods for Cell Painting and shows that they still face limitations. First, technical and biological factors are entangled in the original feature space. Most existing methods rely on global alignment or mixing to remove the batch effect, which may disrupt meaningful biological structure. Second, batch effects in Cell Painting are hierarchical and spatially structured, while observed metadata labels provide only coarse proxies for the underlying technical variation. As shown in Fig. 1, Cell Painting features exhibit a triple-effect pattern Yan et al. (2025); Caicedo et al. (2017), including batch- and plate-level clustering and row- and column-level spatial gradients.

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

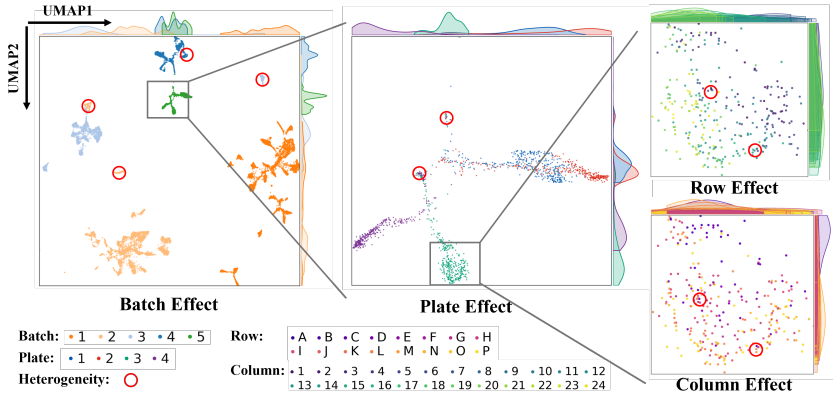

Figure 1: UMAP visualization of JUMP-CP CellProfiler features showing hierarchical technical effects. Global embeddings show batch separation, while zoomed-in regions reveal plate and row/column effects. Red circles highlight metadata-level heterogeneity.

However, these metadata-defined groups may not fully reflect latent technical states Moshkov et al. (2024): technical variation can differ within the same group and overlap across different groups. Therefore, methods that rely on predefined metadata labels may overlook local and sample-specific technical heterogeneity. Third, Cell Painting provides biological anchors Bray et al. (2016), such as negative controls and perturbation replicates, that help identify stable biological structure across technical conditions for contrastive learning. However, pairwise comparisons based on these anchors can be dominated by the batch effect. As a result, contrastive learning without technical conditioning may capture batch-specific artifacts rather than biological structure.

Motivated by these challenges, we formulate Cell Painting batch correction as a conditional representation learning problem, aiming to maximize biological information conditioned on latent technical states. To this end, we propose **HICON**, a hierarchy-aware technical conditioning framework for contrastive biological representation learning. HICON uses a two-branch architecture to disentangle biological and technical representations. The technical branch learns a hierarchical technical embedding as a technical proxy with confidence-aware weak guidance to capture sample-specific technical variation across batch, plate, and spatial levels beyond metadata labels. Rather than treating metadata labels as exact technical domains, HICON uses them as weak guidance to infer sample-specific technical variation. The biological branch leverages compound replicates to define positive and negative samples and reweights contrastive comparisons according to inferred technical similarity. This encourages the biological representation to emphasize perturbation-relevant variation under technical confounding. Extensive experiments across JUMP-CP, BBBC, CPG0036, and RxRx1 show that HICON achieves batch correction, preserves biological structure, and improves downstream utility. HICON also remains effective under metadata corruption and fine-grained technical variation.

Our contributions are summarized as follows: (1) We formulate Cell Painting batch correction as conditional biological representation learning under latent technical confounding. (2) We introduce a hierarchical technical proxy and a confidence-guided learning strategy to capture sample-specific technical variation. (3) We condition replicate-based contrastive learning on the learned proxy, so that supervision emphasizes perturbation-relevant variation under comparable technical states.

2 Related Work

Batch Correction in Cell Painting. Batch correction Li et al. (2026b) is essential for Cell Painting because morphological profiles are affected by experimental conditions. Existing methods include statistical and normalization-based approaches, such as ComBat Johnson et al. (2007), Sphering Kessy et al. (2018), and ZINB-WaVE Risso et al. (2018); Linear projection methods, such as CCA/RPCA Satija et al. (2015), MNN/fastMNN Haghverdi et al. (2018), and Scanorama Hie et al. (2019); graph-based and nearest-neighbor-based methods, such as Seurat integration Stuart et al. (2019) and Harmony Korsunsky et al. (2019); deep learning approaches, such as scVI Lopez et al. (2018), scANVI Xu et al. (2021), DESC Li et al. (2020), CONCORD Zhu et al. (2026), scBCN Wan et al. (2025), and cpDistiller Yan et al. (2025). Although these methods have been adapted or evalu-

ated for Cell Painting Arevalo et al. (2024), most of them use observed metadata labels as covariates and reduce technical variation by adjusting, normalizing, or aligning samples across these predefined groups. Prior work such as cpDistiller has shown that Cell Painting batch effects exhibit a triple-effect pattern and addresses predefined batch, row-column effects using contrastive and adversarial learning. In contrast, HICON treats metadata as weak supervision rather than fixed technical domains, learns a sample-specific hierarchical technical proxy with confidence-aware learning, and directly conditions biological contrastive learning through pairwise technical-similarity weighting.

Contrastive and Biological Representation Learning. Contrastive learning is widely used for representation learning Oord et al. (2018); He et al. (2020); Caron et al. (2020). In Cell Painting, biological anchors such as perturbation replicates can support contrastive learning of biological representations. Recent works have explored self-supervised contrastive learning for Cell Painting Kim et al. (2025); Moutakanni et al. (2025); Lu et al. (2026), and multimodal contrastive learning has been used to align morphological phenotypes with molecular structures or other perturbation modalities Sanchez-Fernandez et al. (2023); Fradkin et al. (2024); Yu et al. (2026); Li et al. (2026a). Despite these advances, pairwise contrastive learning can be technically confounded. Related contrastive learning studies have explored hard-negative mining, debiased objectives, and pair reweighting to control the contribution of sample pairs Robinson et al. (2020); Kalantidis et al. (2020); Chuang et al. (2020); Rao et al. (2025). Different from these general pair-mining strategies, HICON uses the learned technical proxy to estimate pairwise technical similarity and directly condition perturbation-based contrastive learning on latent, sample-specific technical variation.

3 METHOD

3.1 PROBLEM FORMULATION AND OVERVIEW OF THE FRAMEWORK

Let $x_i \in \mathbb{R}^d$ denote the Cell Painting profile of sample i, p_i its perturbation label, and $g_i^{(l)}$ its metadata label at level l (batch, plate, and spatial position). Each profile contains both biological variation and technical variation caused by an unobserved technical state t_i. We use X, P, T, and G to denote dataset-level variables. We formulate Cell Painting batch correction from a conditional information maximization perspective. The goal is to learn a biological representation that preserves perturbation-relevant information conditioned on the technical state T, with target objective $\max_{F_{\text{bio}}} I(Z^{\text{bio}}; P \mid T)$. As T is unobserved, this objective cannot be directly optimized, so we introduce a learnable technical proxy $Z^{\text{tech}} = F_{\text{tech}}(X)$ to approximate this conditioning (Appendix I). As shown in Fig. 2, HICON implements this objective with three components. A two-branch architecture learns biological and technical representations and reduces their mutual information for disentanglement. (Sec. 3.2). Second, the technical branch learns Z^{tech} as a hierarchical proxy for the latent technical state T through confidence-weighted online learning (Sec. 3.3). Third, the biological branch uses this proxy to reweight replicate-based contrastive learning, approximating biological comparisons under comparable technical conditions (Sec. 3.4).

3.2 TWO-BRANCH BIOLOGICAL-TECHNICAL DISENTANGLEMENT

Technical and biological factors are often mixed in the original Cell Painting feature space. To enable hierarchy-aware technical conditioning for robust batch correction, HICON first builds a disentangled two-branch backbone to separate technical variation from perturbation-relevant information. This design promotes a clean conditioning proxy and reduces batch-specific artifacts in the biological representation. Given an input profile x_i, we use two encoders to obtain $z_i^{\text{tech}} = F_{\text{tech}}(x_i)$, $z_i^{\text{bio}} = F_{\text{bio}}(x_i)$ where z_i^{tech} is encouraged to capture technical variation associated with batch, plate, and spatial effects, while z_i^{bio} is encouraged to preserve perturbation-relevant biological information. We describe the learning objectives of the two branches in the following sections. To encourage disentanglement, we minimize the mutual information between Z^{tech} and Z^{bio} using the CLUB upper bound Cheng et al. (2020). We introduce a variational distribution $q_\omega(z_i^{\text{tech}} \mid z_i^{\text{bio}})$ and define

$$\mathcal{L}_{\text{CLUB}} = \frac{1}{N_b} \sum_{i=1}^{N_b} \log q_\omega \left(z_i^{\text{tech}} \mid z_i^{\text{bio}} \right) - \frac{1}{N_b(N_b-1)} \sum_{i=1}^{N_b} \sum_{\substack{j=1 \\ j \neq i}}^{N_b} \log q_\omega \left(z_j^{\text{tech}} \mid z_i^{\text{bio}} \right), \tag{1}$$

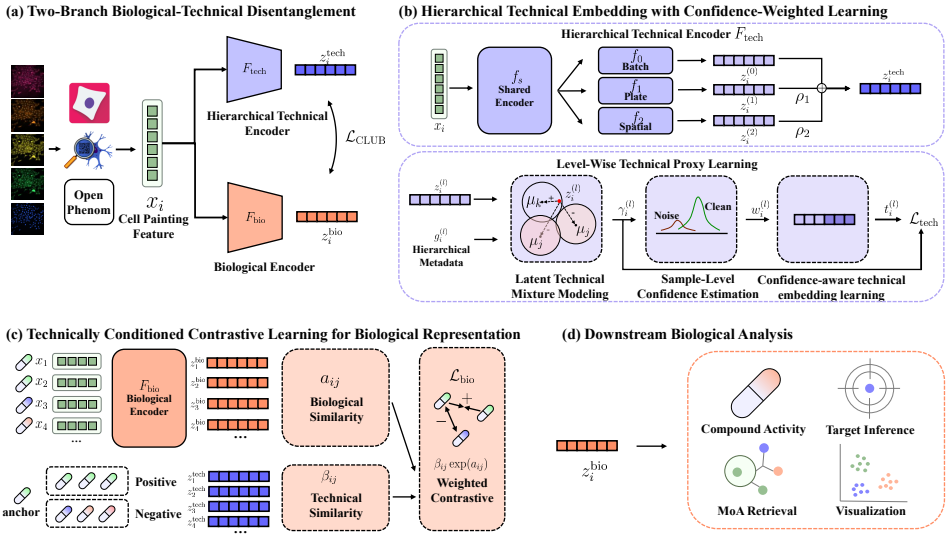

Figure 2: Overview of HICON. (a) A two-branch architecture disentangles biological and technical representations from the Cell Painting feature. (b) The technical branch learns a hierarchical technical proxy through a confidence-aware learning strategy. (c) The biological branch reweights replicate-based contrastive learning according to inferred technical similarity. (d) The resulting biological representation supports downstream biological analyses.

where N_b denotes the mini-batch size. This regularization reduces correlation between branches.

3.3 Hierarchical Technical Embedding with Confidence-Weighted Learning

Since the technical state T is unobserved and the metadata label G provides only coarse and noisy proxies, we design a technical branch to infer a technical embedding z_i^{tech} as a proxy for each sample. This embedding serves as a learnable surrogate of T for conditioning biological representation learning. It is learned by modeling hierarchical technical variation, inferring latent technical assignments from observed features, and adaptively weighting noisy supervision at the sample level.

Hierarchical technical representation. As observed in Fig. 1, technical variation in Cell Painting is not a single global shift, but arises at batch-, plate-, and spatial-scales. A single representation may mix these effects and inaccurately model technical variation. Therefore, we propose a hierarchical structure to capture technical variation across these scales, so that global and local technical effects can be modeled separately while preserving their coarse-to-fine relationship. This is achieved with a shared encoder and level-specific heads. Given an input profile x_i, the shared MLP encoder f_s first extracts a common technical feature. Then, each level-specific MLP head f_l maps this feature to a representation $z_i^{(l)}$ corresponding to batch ($l = 0$), plate ($l = 1$), or spatial ($l = 2$) level technical effect. The final technical representation thus aggregates variations at multiple scales and follows a coarse-to-fine hierarchy: $z_i^{\text{tech}} = z_i^{(0)} + \rho_1 z_i^{(1)} + \rho_2 z_i^{(2)} \quad 1 > \rho_1 > \rho_2$.

Latent technical mixture modeling with confidence-aware learning. As observed in Fig. 1, technical variation exhibits clustered, multi-modal, and heterogeneous patterns within each hierarchy level. Therefore, we model the level-specific technical embedding space with a spherical Gaussian mixture model (GMM) Reynolds et al. (2009). This probabilistic formulation parameterizes latent technical components and estimates the posterior responsibility of each component for each sample. Metadata labels are used as weak initialization for the components, while the learning process is gradually guided by feature-inferred posterior assignments. We then use metadata-posterior consistency to estimate sample-level confidence for weighting technical embedding learning.

(1) Metadata-guided latent technical mixture modeling: At each level l, we model the technical embedding $z_i^{(l)}$ with a spherical Gaussian mixture model: $p(z_i^{(l)}) = \sum_{k=1}^{K_l} \pi_k^{(l)} \mathcal{N}\left(z_i^{(l)} \mid \mu_k^{(l)}, \sigma_l^2 I\right)$, where K_l is the number of latent technical components, $\pi_k^{(l)}$ is the mixture prior, $\mu_k^{(l)}$ is the mean of the k-th component and serves as its technical prototype, and σ_l^2 is the shared isotropic variance.

To obtain a feature-inferred soft assignment of each sample to the latent technical components, we compute the posterior responsibility of each mixture component. We first normalize the level-specific embedding as $\hat{z}_i^{(l)} = \frac{z_i^{(l)}}{\|z_i^{(l)}\|_2}$, and ℓ_2-normalize the technical prototypes accordingly. Under a uniform prior, the posterior responsibility is given by (Appendix H.1)

$$\gamma_{ik}^{(l)} = p(k \mid \hat{z}_i^{(l)}) = \mathrm{softmax}_k \left(\frac{\hat{z}_i^{(l)\top} \mu_k^{(l)}}{\sigma_l^2} \right). \tag{2}$$

Here, $\gamma_{ik}^{(l)}$ is used as the soft assignment of sample i to the k-th latent technical prototype at level l.

(2) Sample-level confidence estimation: Given the feature-inferred posterior assignment $\gamma_i^{(l)}$, we estimate the consistency between each sample and its metadata label.

Specifically, we define the metadata-consistency score as the posterior responsibility assigned to the metadata-indexed component: $s_i^{(l)} = \gamma_{i,g_i^{(l)}}^{(l)} = p\left(k = g_i^{(l)} \mid \hat{z}_i^{(l)}\right)$. A larger $s_i^{(l)}$ indicates stronger consistency between the metadata label and the feature-inferred assignment.

Following prior work Huang et al. (2023), we use the metadata-consistency scores to estimate sample-level metadata confidence. We view samples within each metadata group as a mixture of two latent subsets: metadata-consistent samples and metadata-inconsistent samples. We use these scores to fit a two-component GMM (Appendix H.2): $p\left(s_i^{(l)}\right) = \sum_{c\in\{0,1\}} p(c)\, p\left(s_i^{(l)} \mid c\right)$, where $c = 1$ and $c = 0$ correspond to high- and low-consistency components. We use the posterior probability $q_i^{(l)} = p(c = 1 \mid s_i^{(l)})$ as the sample-level confidence estimate. The final confidence weight is defined as $w_i^{(l)} = \max\left(q_i^{(l)}, \eta_l(\tau)\right)$, where $\eta_l(\tau)$ is a training-dependent lower bound that prevents overly small weights in early training, and τ denotes the training step.

(3) Confidence-aware technical embedding learning: The sample-level confidence weights are used as weights to adapt technical embedding learning. High-confidence samples receive stronger metadata-based supervision, whereas low-confidence samples rely more on posterior assignments.

Component centers are updated using a confidence-weighted responsibility mean:

$$\mu_k^{(l)} \leftarrow m_l(\tau)\, \mu_k^{(l)} + (1 - m_l(\tau)) \frac{\sum_i w_i^{(l)} \gamma_{ik}^{(l)} \hat{z}_i^{(l)}}{\sum_i w_i^{(l)} \gamma_{ik}^{(l)}}. \tag{3}$$

Here, $\gamma_{ik}^{(l)}$ denotes the feature-inferred soft assignment to component k, and $m_l(\tau)$ is a momentum term that stabilizes the update during training. Details are provided in Appendix H.3.

For target construction, we define an adaptive soft target that interpolates between the metadata label and the feature-inferred posterior: $t_i^{(l)} = w_i^{(l)} \,\mathrm{onehot}(g_i^{(l)}) + \left(1 - w_i^{(l)}\right)\mathrm{stopgrad}\left(\gamma_i^{(l)}\right)$.

The overall technical embedding loss is computed by aggregating level-wise cross-entropy losses:

$$\mathcal{L}_{\mathrm{tech}} = \sum_{l=0}^{2} \lambda_l \left[\frac{1}{N_b} \sum_{i=1}^{N_b} \mathrm{CE}\left(t_i^{(l)}, \gamma_i^{(l)}\right) \right]. \tag{4}$$

3.4 Technically Conditioned Contrastive Learning for Biological Representation Learning

The biological branch instantiates the CMI objective using technically conditioned contrastive learning. It learns a corrected representation Z^{bio} that preserves perturbation-relevant information. While this can be done through replicate-based contrastive learning, standard objectives treat all pairwise comparisons as equally consistent and ignore sample-level technical heterogeneity. Therefore, we use the learned technical proxy Z^{tech} to condition the contrastive objective by softly reweighting positive and negative pairs according to inferred technical similarity. This reduces the influence of pairs dominated by technical artifacts.

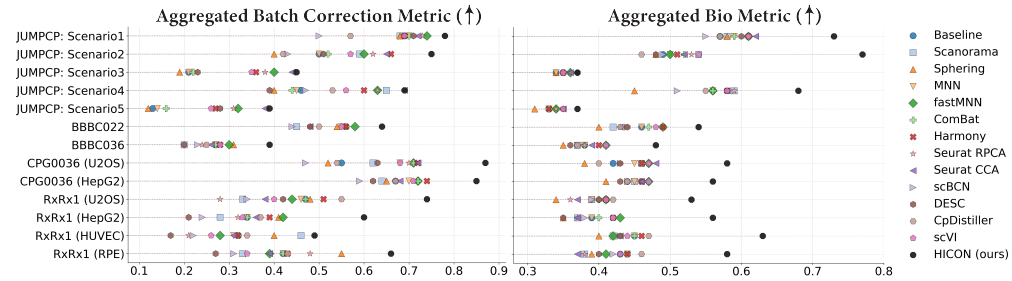

Figure 3: Benchmark evaluation across JUMP-CP, BBBC, CPG0036, and RxRx1. We report aggregated batch-correction and biological preservation metrics; higher is better.

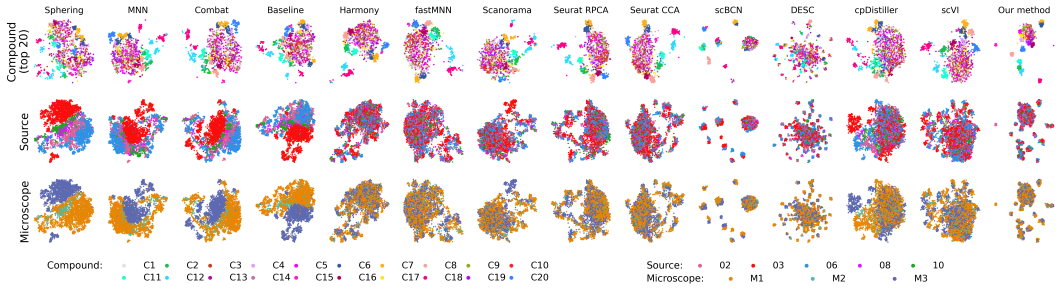

Figure 4: UMAP visualization under JUMP-CP (Scenario 4).

Replicate-based biological supervision. To preserve perturbation-relevant morphology, we exploit perturbation replicates as biological supervision. Replicates of the same perturbation are expected to share consistent phenotypic responses across technical conditions. Contrastive learning leverages these biological anchors through sample-sample comparisons, capturing stable biological structure and approximating the maximization of perturbation-relevant mutual information (Appendix I).

Specifically, given a mini-batch $\{x_i\}_{i=1}^{N_b}$, each sample is treated as an anchor. Replicates that share the same perturbation label define the positive set $\mathcal{P}(i) = \{\, j \neq i \mid P_j = P_i \,\}$, whereas samples with different perturbations define the negative set $\mathcal{N}(i) = \{\, k \mid P_k \neq P_i \,\}$. For anchors without another replicate in the mini-batch, we construct an augmented view of the same sample as an additional positive, following the feature-space augmentation described in Appendix G. These sets define the biological comparisons used in the contrastive objective, where positives encourage perturbation-level consistency, and negatives preserve separability across perturbations.

Soft technical conditioning. To reduce technical confounding in replicate-based contrastive learning, we use the learned technical proxy Z^{tech} to estimate pairwise technical similarity and reweight contrastive comparisons, so that the objective emphasizes biological comparisons under comparable technical conditions and aligns with the conditional information objective (Appendix I).

We define a soft technical similarity kernel: $\beta_{ik} = r\left(\left\langle \tilde{z}_i^{\text{tech}}, \tilde{z}_k^{\text{tech}} \right\rangle\right)$, $\tilde{z}_i^{\text{tech}} = \left[\frac{z_i^{\text{tech}}}{\|z_i^{\text{tech}}\|_2}\right]$, where $r(\cdot)$ denotes a linear rescaling to a bounded interval $[0.6, 1.4]$ and $[\cdot]$ denotes the stop-gradient operation. The resulting score β_{ik} is used as a proxy for pairwise technical similarity.

The resulting technical similarity scores are used to softly modulate pairwise contributions according to their inferred technical comparability. Comparisons under more comparable technical conditions receive greater emphasis, reducing the influence of technical mismatch on perturbation-based contrastive learning. Using these weights, we define the weighted contrastive loss for anchor i as

$$\mathcal{L}_{\text{bio}} = -\frac{1}{N_b} \sum_{i=1}^{N_b} \log \frac{\sum_{j \in \mathcal{P}(i)} \beta_{ij} \exp(a_{ij})}{\sum_{j \in \mathcal{P}(i)} \beta_{ij} \exp(a_{ij}) + \sum_{k \in \mathcal{N}(i)} \beta_{ik} \exp(a_{ik})}. \tag{5}$$

where a_{ij} and a_{ik} denote pairwise similarities computed from z_i^{bio}. By reweighting pairwise comparisons according to technical similarity, this objective encourages the model to focus on perturbation-relevant biological information under comparable technical conditions.

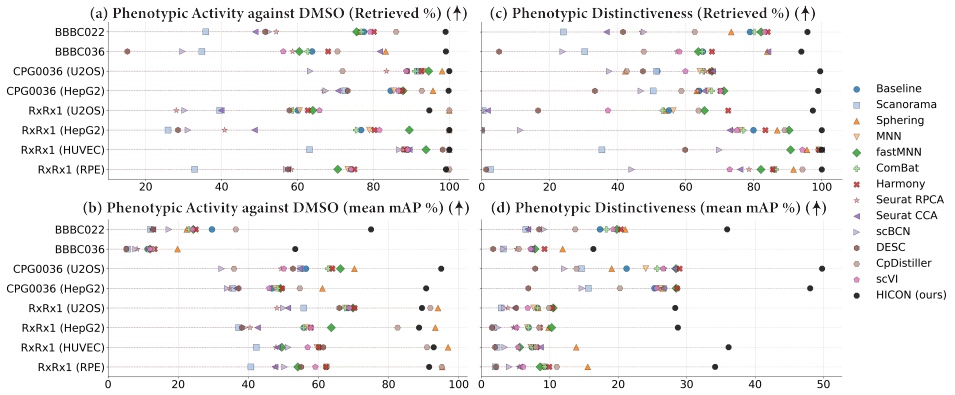

Figure 5: Downstream biological utility measured by phenotypic activity against DMSO and phenotypic distinctiveness (denoted as PSNB), following the evaluation protocol of Kalinin et al. (2025).

3.5 Overall Objective

The full model is trained by jointly optimizing the three objectives: $\mathcal{L}_{\text{total}} = \lambda_{\text{tech}}\mathcal{L}_{\text{tech}} + \lambda_{\text{bio}}\mathcal{L}_{\text{bio}} + \lambda_{\text{CLUB}}\mathcal{L}_{\text{CLUB}}$. Here, λ_{tech}, λ_{bio}, and λ_{CLUB} are hyperparameters for balancing the three objectives.

4 Experiments

4.1 Experimental Setup

Datasets and Features. We evaluate HICON on four Cell Painting benchmarks across multiple scenarios: JUMP-CP, BBBC Ljosa et al. (2012), CPG0036 Wolff et al. (2024), and RxRx1 Sypetkowski et al. (2023). We use three types of features: handcrafted CellProfiler Stirling et al. (2021), learned DeepProfiler Moshkov et al. (2024), and foundation-model OpenPhenom features Kraus et al. (2024). All evaluations use disjoint train/test perturbations, with no test perturbation in training.

Comparison Methods. We compare HICON with representative statistical, alignment-based, and deep learning batch-correction methods. Full method descriptions are provided in Appendix C.

Metrics. Following Arevalo et al. (2024), we compute the full set of batch-correction and biological-preservation metrics. We report aggregated scores and downstream biological utility in the main text. Metric definitions, aggregation details, and complete per-metric results in Appendix D.

4.2 Benchmark Evaluation

We evaluate HICON on a broad benchmark covering JUMP-CP, CPG0036, BBBC, and RxRx1. These settings cover diverse technical and biological conditions and feature representations. Figure 3 summarizes the main results. (Full result can be found in Appendix N.2-N.5) Beyond the real-data benchmarks, we use a controlled synthetic setting to evaluate whether the learned technical proxy can recover known latent technical states under noisy metadata (Appendix E).

Results on JUMP-CP scenarios. For JUMP-CP, we follow the settings in Arevalo et al. (2024) and construct five scenarios, denoted as Scenario 1–5, with increasing levels of technical complexity and batch correction difficulty (see Appendix A). As shown in Fig. 3, the proposed method performs strongly across all five scenarios and remains competitive as technical complexity increases. In Scenarios 1, 2, and 4, where replicate support is sufficient, HICON achieves strong biological preservation and batch correction. In Scenarios 3 and 5, where technical heterogeneity is stronger, all methods degrade to some extent, but our method remains among the strongest overall. These results indicate that HICON achieves effective technical correction and biological preservation across a wide range of CellProfiler-based settings. Qualitative visualization also supports this trend. As shown in Fig. 4, the corrected representation improves mixing across technical sources while preserving clear compound-level structure. This suggests that the hierarchical technical encoder captures fine-grained technical variation beyond coarse batch labels, enabling HICON to reduce technical variation without collapsing meaningful biological structure.

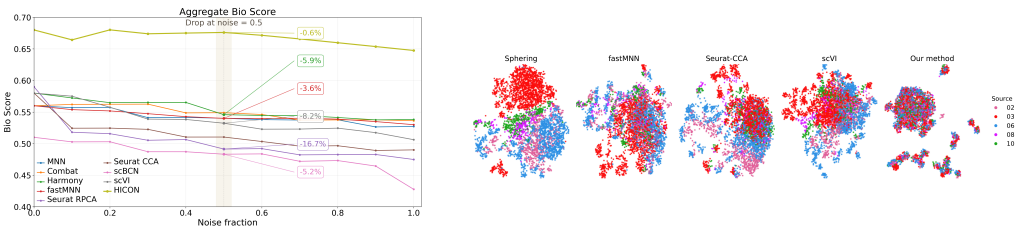

Figure 6: Batch labels are progressively corrupted from 0% to 100% and used as noisy supervision during training. (A) Left: Aggregate biological score under increasing corruption ratios. (B) Right: Embedding visualization at 50% corruption for selected methods. Points colored by the clean labels.

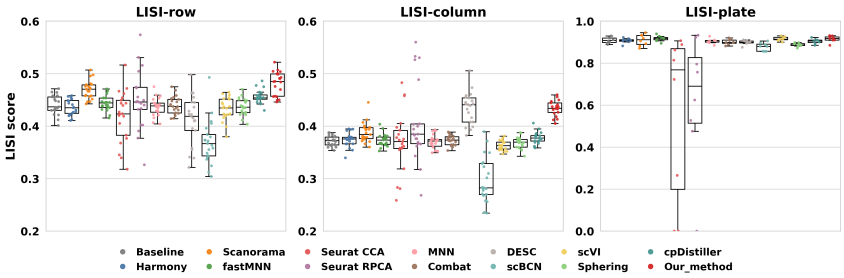

Figure 7: Evaluation of triple-effect correction. LISI scores are reported for plate, row, and column.

Results on BBBC, CPG0036, and RxRx1. HICON also shows strong performance across different datasets and feature representations. As shown in Fig. 3, HICON consistently ranks among the best methods in both aggregated batch correction and biological preservation. The gains are particularly clear on CPG0036 and RxRx1, where technical variation is more complex and methods need to preserve biological structure under stronger experimental shifts. These results further show that HICON is effective across diverse benchmark settings rather than being limited to a single dataset.

Downstream biological utility. Beyond aggregate benchmark metrics, we evaluate whether the corrected features support downstream biological analyses. As shown in Fig. 5, HICON shows clear gains in phenotypic activity against DMSO and phenotypic distinctiveness across BBBC022, BBBC036, CPG0036, and RxRx1. Some baselines perform well on benchmark metrics, but they do not show strong performance in downstream tasks. We further evaluate mechanism-of-action retrieval on BBBC022 and BBBC036 (Results are in Appendix K), where HICON also achieves strong performance. Together, these results demonstrate the downstream utility of HICON.

4.3 Evaluation under Batch Label Corruption

We further examine whether HICON learns technical structure beyond the observed batch labels. Specifically, we progressively corrupt batch labels from 0% to 100% in steps of 10%, thereby decoupling the observed metadata G from the underlying technical factors T. Since the batch annotations are corrupted in this setting, batch-related metrics become unreliable. We therefore focus on biological metrics and use the corrupted labels as noisy supervision during training.

As shown in Fig. 6(A), most compared methods show a clear decrease in biological performance as the corruption ratio increases. This indicates that their correction performance is tied to G. HICON also degrades under severe corruption, but remains stronger than other methods across all corruption levels. Fig. 6(B) visualizes the embeddings under 50% label corruption for selected methods. Other methods show clear separation by source, indicating that their correction remains affected by noisy batch supervision. In contrast, HICON achieves mixing across sources while preserving structure.

4.4 Triple-effect Correction

We further examine whether HICON can correct hierarchical technical effects beyond coarse batch-level variation. Specifically, we construct a within-batch evaluation on JUMP-CP Scenario 4 by selecting a single batch and assessing plate-, row-, and column-level effects. Correction quality is measured using LISI scores computed with respect to the corresponding plate, row, and column

Objective	Batch level			Plate level				Row level				Column level			
	Batch correction	Bio metrics	Overall	Graph connectivity	kBET	LISI plate	Overall	Graph connectivity	kBET	LISI row	Overall	Graph connectivity	kBET	LISI column	Overall
$L^{(0)}$	0.62	0.57	0.60	0.93	0.50	0.30	0.58	0.95	0.80	0.50	0.75	0.94	0.82	0.48	0.75
$L^{(0)}$, $L^{(1)}$	0.62	0.58	0.60	0.95	0.58	0.35	0.63	0.96	0.84	0.53	0.78	0.95	0.86	0.50	0.77
$L^{(0)}$, $L^{(1)}$, $L^{(2)}$	**0.64**	0.58	0.60	**0.96**	**0.58**	**0.36**	**0.63**	**0.97**	**0.92**	**0.56**	**0.82**	**0.96**	**0.91**	**0.54**	**0.80**

Table 1: Impact of hierarchical prototype modeling at different technical levels.

Variant	Batch metrics	Bio metrics	Overall
Vanilla InfoNCE	0.37	0.49	0.43
Replicate-based Contrastive	0.56	0.55	0.56
Hard conditioning (weight)	0.44	0.54	0.49
Hard conditioning (sample)	0.48	0.53	0.51
Two-branch (soft con)	**0.64**	**0.58**	**0.60**

Variant	Batch metrics	Bio metrics	Overall
w/o conf. weight	0.51	0.52	0.51
w/o soft target	0.58	0.56	0.56
w/o proto. update	0.54	0.56	0.55
Fixed proto.	0.41	0.55	0.48
Two-branch (full tech.)	**0.64**	**0.58**	**0.60**

Variant	Batch metrics	Bio metrics	Overall
No disentanglement	0.64	0.58	0.60
GRL	0.66	0.58	0.60
Orthogonality	0.64	0.59	0.60
CLUB (Full HICON)	**0.68**	**0.60**	**0.63**

Table 2: Ablations of (A)biological branch, (B)technical branch and (C)disentanglement module.

labels. Fig. 7 reports the results across these three levels. Compared with plate-level effects, row- and column-level effects are more difficult to remove, as they reflect finer-grained spatial variation within plates. While most methods achieve comparable correction at the plate level, HICON shows clearer advantages on row- and column-level correction. These results suggest that HICON can better model and remove hierarchical technical variation, including fine-grained triple effects.

4.5 Ablation Study

We conduct ablation studies on JUMP-CP Scenario 4 to evaluate the contribution of each component. We use 30% of the training data with fewer perturbation replicates and imbalanced batch distributions, creating a more challenging ablation setting. The details can be found in Appendix M.

Hierarchical technical modeling. We ablate the technical hierarchy by adding plate- and spatial-level objectives to the batch-level objective. As shown in Table 1, adding technical levels improves correction performance, especially for fine-grained technical variation.

Biological branch ablation. We evaluate how technical information should be used in biological learning. Table 2(A) starts from Vanilla InfoNCE, then adds perturbation-replicate positives, and further incorporates technical similarity through hard or soft conditioning. Replicate-based SupCon substantially improves over Vanilla InfoNCE, confirming the importance of biological positives. Hard conditioning brings limited gains, while soft conditioning achieves the best overall scores. This suggests that technical similarity is more effective than a hard selection rule.

Technical branch ablation. We evaluate the design of the technical branch by removing one component at a time. Table 2(B) shows that removing confidence weighting, soft targets, or dynamic prototype refinement decreases performance. This suggests that technical proxy learning requires both weak metadata guidance and adaptation to sample-level technical heterogeneity.

Disentanglement ablation. Finally, we evaluate different strategies for disentanglement, including Gradient Reversal Layer (GRL) Ganin & Lempitsky (2015), Orthogonality Edelman et al. (1998), and CLUB. Table 2(C) shows that CLUB achieves the best performance among these strategies.

5 Conclusion

In this paper, we presented HICON, a hierarchy-aware technical conditioning framework for Cell Painting batch correction. Instead of treating observed metadata labels as exact technical domains, HICON formulates batch correction as conditional biological representation learning under latent technical confounding. It learns a disentangled biological representation and a hierarchical technical proxy through confidence-aware weak metadata guidance, and uses the learned proxy to condition replicate-based contrastive learning. Across multiple Cell Painting datasets, feature representations, and evaluation settings, HICON achieves strong batch correction while preserving biological structure and improving downstream biological utility. Additional analyses under metadata corruption and triple-effect correction further show that HICON can capture technical variation beyond metadata labels. These results suggest that learning latent technical structure and using it to condition biological supervision is a promising direction for robust Cell Painting representation learning.

AI USE STATEMENT

Generative AI tools were used primarily to assist with language editing, translation, and improving the clarity, organization, and presentation of the manuscript. They were also used to provide limited feedback during the discussion of methodological and experimental choices and the interpretation of results. All AI-assisted content was reviewed and revised by the authors. The authors independently verified the technical content, experimental results, and scientific claims, and take full responsibility for the final manuscript.

ETHICS STATEMENT

This work is a computational study based on existing public Cell Painting datasets and does not involve the recruitment of human participants, the collection of personally identifiable information, or new human or animal experiments. HICON is intended as a research method for biological profiling and representation learning rather than for clinical diagnosis or treatment decisions. As model outputs may be affected by limitations or biases in the underlying datasets, biological conclusions should be supported by appropriate downstream validation.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the datasets and evaluation scenarios in Appendix A, the feature representations in Appendix B, and the comparison methods in Appendix C. Detailed implementation information, including the network architecture, training settings, optimization, random seeds, and hyperparameters, is provided in Appendix G. Additional experimental results and computational resource requirements are also reported in the appendix. These details are provided to facilitate reproduction of the reported experiments.

REFERENCES

Mohammad Akbarzadeh, Ilka Deipenwisch, Beate Schoelermann, Axel Pahl, Sonja Sievers, Slava Ziegler, and Herbert Waldmann. Morphological profiling by means of the cell painting assay enables identification of tubulin-targeting compounds. *Cell Chemical Biology*, 29(6):1053–1064, 2022.

D Michael Ando, Cory Y McLean, and Marc Berndl. Improving phenotypic measurements in high-content imaging screens. *BioRxiv*, pp. 161422, 2017.

John Arevalo, Ellen Su, Jessica D. Ewald, Robert van Dijk, Anne E. Carpenter, and Shantanu Singh. Evaluating batch correction methods for image-based cell profiling. *Nature Communications*, 15 (1):6516, August 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-50613-5.

Vincent Blay, Bhairavi Tolani, Sunita P Ho, and Michelle R Arkin. High-throughput screening: today's biochemical and cell-based approaches. *Drug discovery today*, 25(10):1807–1821, 2020.

Mark-Anthony Bray, Shantanu Singh, Han Han, Chadwick T Davis, Blake Borgeson, Cathy Hartland, Maria Kost-Alimova, Sigrun M Gustafsdottir, Christopher C Gibson, and Anne E Carpenter. Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nature protocols*, 11(9):1757–1774, 2016.

Mark-Anthony Bray, Sigrun M Gustafsdottir, Mohammad H Rohban, Shantanu Singh, Vebjorn Ljosa, Katherine L Sokolnicki, Joshua A Bittker, Nicole E Bodycombe, Vlado Dančík, Thomas P Hasaka, et al. A dataset of images and morphological profiles of 30 000 small-molecule treatments using the cell painting assay. *Gigascience*, 6(12):giw014, 2017.

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.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Srinivas Niranj Chandrasekaran, Jeanelle Ackerman, Eric Alix, D Michael Ando, John Arevalo, Melissa Bennion, Nicolas Boisseau, Adriana Borowa, Justin D Boyd, Laurent Brino, et al. Jump cell painting dataset: morphological impact of 136,000 chemical and genetic perturbations. *BioRxiv*, pp. 2023–03, 2023.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*, pp. 1779–1788. PMLR, 2020.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

Marta M Fay, Oren Kraus, Mason Victors, Lakshmanan Arumugam, Kamal Vuggumudi, John Urbanik, Kyle Hansen, Safiye Celik, Nico Cernek, Ganesh Jagannathan, et al. Rxrx3: Phenomics map of biology. *Biorxiv*, pp. 2023–02, 2023.

Philip Fradkin, Puria Azadi Moghadam, Karush Suri, Frederik Wenkel, Maciej Sypetkowski, and Dominique Beaini. Molphenix: A multimodal foundation model for phenomolecular retrieval. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*, 2024.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Christopher J Gerry, Bruce K Hua, Mathias J Wawer, Jonathan P Knowles, Shawn D Nelson Jr, Oscar Verho, Sivaraman Dandapani, Bridget K Wagner, Paul A Clemons, Kevin I Booker-Milburn, et al. Real-time biological annotation of synthetic compounds. *Journal of the American Chemical Society*, 138(28):8920–8927, 2016.

Sigrun M Gustafsdottir, Vebjorn Ljosa, Katherine L Sokolnicki, J Anthony Wilson, Deepika Walpita, Melissa M Kemp, Kathleen Petri Seiler, Hyman A Carrel, Todd R Golub, Stuart L Schreiber, et al. Multiplex cytological profiling assay to measure diverse cellular states. *PloS one*, 8(12):e80999, 2013.

Laleh Haghverdi, Aaron TL Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology*, 36(5):421–427, 2018.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Brian Hie, Bryan Bryson, and Bonnie Berger. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nature biotechnology*, 37(6):685–691, 2019.

Markus Hofmarcher, Elisabeth Rumetshofer, Djork-Arne Clevert, Sepp Hochreiter, and Gunter Klambauer. Accurate prediction of biological assays with high-throughput microscopy images and convolutional networks. *Journal of chemical information and modeling*, 59(3):1163–1171, 2019.

Zhizhong Huang, Junping Zhang, and Hongming Shan. Twin contrastive learning with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11661–11670, 2023.

James Inglese, Ronald L Johnson, Anton Simeonov, Menghang Xia, Wei Zheng, Christopher P Austin, and Douglas S Auld. High-throughput screening assays for the identification of chemical probes. *Nature chemical biology*, 3(8):466–479, 2007.

W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in neural information processing systems*, 33: 21798–21809, 2020.

Alexandr A Kalinin, John Arevalo, Erik Serrano, Loan Vulliard, Hillary Tsang, Michael Bornholdt, Alán F Muñoz, Suganya Sivagurunathan, Bartek Rajwa, Anne E Carpenter, et al. A versatile information retrieval framework for evaluating profile strength and similarity. *Nature Communications*, 16(1):5181, 2025.

Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.

Vladislav Kim, Nikolaos Adaloglou, Marc Osterland, Flavio M Morelli, Marah Halawa, Tim König, David Gnutt, and Paula A Marin Zapata. Self-supervision advances morphological profiling by unlocking powerful image representations. *Scientific Reports*, 15(1):4876, 2025.

Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature Methods*, pp. 1289–1296, December 2019. doi: 10.1038/s41592-019-0619-0.

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*, pp. 11757–11768, 2024.

Mengran Li, Zelin Zang, Wenbin Xing, Junzhou Chen, Ronghui Zhang, Jiebo Luo, and Stan Z Li. Learning cell-aware hierarchical multi-modal representations for robust molecular modeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(1):623–631, 2026a.

Shuang Li, Malte Lücken, John C Marioni, Sarah A Teichmann, and Peng He. Toward informed batch correction for single-cell transcriptome integration. *Nature Computational Science*, pp. 1–11, 2026b.

Xiangjie Li, Kui Wang, Yafei Lyu, Huize Pan, Jingxiao Zhang, Dwight Stambolian, Katalin Susztak, Muredach P. Reilly, Gang Hu, and Mingyao Li. Deep learning enables accurate clustering with batch effect removal in single-cell rna-seq analysis. *Nature Communications*, May 2020. doi: 10.1038/s41467-020-15851-3.

Vebjorn Ljosa, Katherine L Sokolnicki, and Anne E Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature methods*, 9(7):637, 2012.

Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, pp. 1053–1058, December 2018. doi: 10.1038/s41592-018-0229-2.

Mingyu Lu, Ethan Weinberger, Chanwoo Kim, and Su-In Lee. Cellclip-learning perturbation effects in cell painting via text-guided contrastive learning. *Advances in Neural Information Processing Systems*, 38:124505–124537, 2026.

Nikita Moshkov, Michael Bornholdt, Santiago Benoit, Matthew Smith, Claire McQuin, Allen Goodman, Rebecca A Senft, Yu Han, Mehrtash Babadi, Peter Horvath, et al. Learning representations for image-based profiling of perturbations. *Nature communications*, 15(1):1594, 2024.

Théo Moutakanni, Camille Couprie, Seungeun Yi, Michael Doron, Zitong S Chen, Nikita Moshkov, Elouan Gardes, Mathilde Caron, Hugo Touvron, Armand Joulin, et al. Cell-dino: Self-supervised image-based embeddings for cell fluorescent microscopy. *PLOS Computational Biology*, 21(12): e1013828, 2025.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Jiahua Rao, Hanjing Lin, Leyu Chen, Jiancong Xie, Shuangjia Zheng, and Yuedong Yang. Multimodal contrastive learning with negative sampling calibration for phenotypic drug discovery. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30752–30762. IEEE, 2025.

Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663):3, 2009.

Davide Risso, Fanny Perraudeau, Svetlana Gribkova, Sandrine Dudoit, and Jean-Philippe Vert. A general and flexible method for signal extraction from single-cell rna-seq data. *Nature communications*, 9(1):284, 2018.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

Ana Sanchez-Fernandez, Elisabeth Rumetshofer, Sepp Hochreiter, and Günter Klambauer. Cloome: contrastive learning unlocks bioimaging databases for queries with chemical structures. *Nature Communications*, 14(1):7339, 2023.

Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 33(5):495–502, 2015.

Andreas Scherer. *Batch effects and noise in microarray experiments: sources and solutions*, volume 868. Wiley Online Library, 2009.

Srijit Seal, Hongbin Yang, Luis Vollmers, and Andreas Bender. Comparison of cellular morphological descriptors and molecular fingerprints for the prediction of cytotoxicity-and proliferation-related assays. *Chemical Research in Toxicology*, 34(2):422–437, 2021.

Srijit Seal, Jordi Carreras-Puigvert, Maria-Anna Trapotsi, Hongbin Yang, Ola Spjuth, and Andreas Bender. Integrating cell morphology with gene expression and chemical structure to aid mitochondrial toxicity detection. *Communications Biology*, 5(1):858, 2022.

Srijit Seal, Maria-Anna Trapotsi, Ola Spjuth, Shantanu Singh, Jordi Carreras-Puigvert, Nigel Greene, Andreas Bender, and Anne E Carpenter. A decade in a systematic review: The evolution and impact of cell painting. *bioRxiv*, 2024.

Jaak Simm, Günter Klambauer, Adam Arany, Marvin Steijaert, Jörg Kurt Wegner, Emmanuel Gustin, Vladimir Chupakhin, Yolanda T Chong, Jorge Vialard, Peter Buijnsters, et al. Repurposing high-throughput image assays enables biological activity prediction for drug discovery. *Cell chemical biology*, 25(5):611–618, 2018.

David R Stirling, Madison J Swain-Bowden, Alice M Lucas, Anne E Carpenter, Beth A Cimini, and Allen Goodman. Cellprofiler 4: improvements in speed, utility and usability. *BMC bioinformatics*, 22(1):433, 2021.

Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *cell*, 177(7):1888–1902, 2019.

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*, pp. 4285–4294, 2023.

D Lansing Taylor. *High content screening*, volume 356. Springer Science & Business Media, 2008.

Maria-Anna Trapotsi, Layla Hosseini-Gerami, and Andreas Bender. Computational analyses of mechanism of action (moa): data, methods and integration. *RSC chemical biology*, 3(2):170–200, 2022a.

Maria-Anna Trapotsi, Elizabeth Mouchet, Guy Williams, Tiziana Monteverde, Karolina Juhani, Riku Turkki, Filip Miljkovic, Anton Martinsson, Lewis Mervin, Kenneth R Pryde, et al. Cell morphological profiling enables high-throughput screening for proteolysis targeting chimera (protac) phenotypic signature. *ACS Chemical Biology*, 17(7):1733–1744, 2022b.

Lei Wan, Yang Zhou, Xingzhi Wang, Jing Qi, and Shuilin Jin. scbcn: deep learning-based batch correction network for integration of heterogeneous single-cell data. *Briefings in Bioinformatics*, 26(5):bbaf503, 2025.

Christopher Wolff, Martin Neuenschwander, Carsten Jörn Beese, Divya Sitani, Maria C Ramos, Alzbeta Srovnalova, María José Varela, Pavel Polishchuk, Katholiki E Skopelitou, Ctibor Škuta, et al. Morphological profiling dataset of eu-openscreen bioactive compounds over multiple imaging sites and cell lines. *bioRxiv*, pp. 2024–08, 2024.

Chenling Xu, Romain Lopez, Edouard Mehlman, Jeffrey Regier, Michael I Jordan, and Nir Yosef. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular systems biology*, 17(1):MSB20209620, 2021.

Chengwei Yan, Yu Zhang, Jiuxin Feng, Heyang Hua, Zhihan Ruan, Zhen Li, Siyu Li, Chaoyang Yan, Pingjing Li, Jian Liu, and Shengquan Chen. Triple-effect correction for cell painting data with contrastive and domain-adversarial learning. *Nature Communications*, 16(1):6886, July 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-62193-z.

Yemin Yu, Emre Hayir, Neil Tenenholtz, Lester Mackey, Ying Wei, David Alvarez-Melis, Ava P Amini, and Alex X Lu. Integrating chemical structures as treatments improves representations of microscopy images for morphological profiling. *PLOS Computational Biology*, 22(7):e1014483, 2026.

Qin Zhu, Zuzhi Jiang, Binyamin Zuckerman, Leor Weinberger, Matt Thomson, and Zev J Gartner. Revealing a coherent cell-state landscape across single-cell datasets with concord. *Nature Biotechnology*, pp. 1–15, 2026.

A Dataset and Scenario

We evaluate our method across multiple Cell Painting datasets and feature representations, as summarized in Table 3. For JUMP-CP, we follow five benchmark scenarios with increasing technical and biological heterogeneity, varying microscope type, laboratory source, compound diversity, and replicate availability. We further evaluate DeepProfiler features on BBBC022 and BBBC036, CellProfiler and DeepProfiler features on CPG0036, and OpenPhenom foundation-model features on RxRx1 to test robustness across different feature spaces and experimental settings. For all datasets, train/test splits are constructed at the perturbation level, such that all replicates of a perturbation belong to the same split and no perturbation overlaps between training and test sets.

Dataset	Scenario	Feature Type	# Microscope Types	# Laboratories / Sites	# Compounds / Perturbations	# Replicates
JUMP-CP	Scenario1	CellProfiler / DeepProfiler	1	1	301	6,698
JUMP-CP	Scenario2	CellProfiler / DeepProfiler	1	3	301	11,795
JUMP-CP	Scenario3	CellProfiler / DeepProfiler	1	3	82,275	213,120
JUMP-CP	Scenario4	CellProfiler / DeepProfiler	3	5	301	21,074
JUMP-CP	Scenario5	CellProfiler	3	5	82,411	325,338
BBBC022	–	DeepProfiler	1	1	1,539	6,127
BBBC036	–	DeepProfiler	1	1	2,238	17,570
CPG0036	FMP_U2OS	CellProfiler	1	1	2,448	9,817
CPG0036	FMP_HepG2	CellProfiler	1	1	2,461	9,913
CPG0036	IMTM_HepG2	CellProfiler	1	1	2,458	9,884
CPG0036	MEDINA_HepG2	CellProfiler	1	1	2,466	9,951
CPG0036	USC_HepG2	CellProfiler	1	1	2,466	9,963
RxRx1	U2OS	OpenPhenom	1	1	1,138	6,128
RxRx1	HepG2	OpenPhenom	1	1	1,138	13,497
RxRx1	HUVEC	OpenPhenom	1	1	1,138	29,400
RxRx1	RPE	OpenPhenom	1	1	1,138	13,484

Table 3: Summary of datasets and evaluation scenarios used in our batch correction experiments.

JUMP-CP Chandrasekaran et al. (2023) is a large-scale public Cell Painting reference dataset generated by the Joint Undertaking for Morphological Profiling consortium to profile the morphological impact of chemical and genetic perturbations in U2OS cells. The dataset cpg0016 covers over 116,750 unique compounds, 12,602 ORF overexpression genes, and 7,975 CRISPR-Cas9 knockout genes across multiple pharmaceutical and academic partner sites. Overall, JUMP-CP is estimated to contain approximately 115 TB of imaging and profile data, capturing around 1.6 billion cells with single-cell and well-level morphological profiles, making it one of the largest publicly available resources for studying perturbation-induced cellular phenotypes, batch effects, compound–gene relationships, and representation learning in Cell Painting.

BBBC022 Gustafsdottir et al. (2013) is an early Cell Painting-style benchmark dataset based on a multiplex cytological profiling assay designed to capture diverse compound-induced cellular states. The dataset profiles U2OS cells treated with 1,600 commercially available bioactive compounds, typically at 10 μM for 48 hours, with four replicates per compound in 384-well plates. The study showed that these profiles could identify active compounds and cluster compounds with related mechanisms of action or chemical structure, making BBBC022 useful for evaluating morphological profiling, compound similarity, and mechanism-of-action discovery from Cell Painting images.

BBBC036 Bray et al. (2017) is a large-scale Cell Painting dataset of U2OS cells treated with 30,616 small molecules in quadruplicate, designed for image-based morphological profiling and compound mechanism-of-action analysis. The dataset contains 919,265 five-channel microscopy fields of view acquired from 406 384-well plates at 20$\times$ magnification, with six sites imaged per well. The dataset provides raw images, CellProfiler pipelines, illumination correction functions, quality-control metadata, extracted single-cell morphological features, per-well morphological profiles, and chemical annotations such as compound names, SMILES, and PubChem identifiers.

CPG0036 Wolff et al. (2024) is a public Cell Painting dataset profiling the EU-OPENSCREEN Bioactive Compound Set, a curated collection of 2,464 bioactive compounds enriched for approved drugs, chemical probes, annotated targets, and known mechanisms of action. The dataset was generated across four European imaging sites using high-throughput confocal microscopy, primarily in HepG2 cells, with an additional U2OS dataset produced at FMP for comparison with widely used Cell Painting resources. The dataset is useful as a well-annotated reference for studying compound bioactivity, toxicity, mechanism-of-action exploration, cross-cell-line morphological differences, and batch/site normalization in Cell Painting analysis.

RxRx1 Sypetkowski et al. (2023) is a large fluorescence microscopy dataset released by Recursion for evaluating experimental batch-effect correction methods in high-content imaging. It contains 125,510 six-channel Cell Painting-like images of human cells across 51 experimental batches, covering four cell types: HUVEC, RPE, HepG2, and U2OS. Each experiment uses 384-well plates with siRNA genetic perturbations, including 1,108 treatment siRNAs, 30 positive-control siRNAs, and an untreated EMPTY class, giving 1,139 total classes. The dataset is designed around a perturbation-classification task where models must predict the genetic perturbation in held-out experimental batches, making it especially useful for studying batch correction, domain generalization, and robust representation learning in microscopy-based phenotypic profiling.

JUMP-CP is used as our primary large-scale benchmark. We follow the five evaluation scenarios defined in prior batch-correction benchmarking work. These scenarios are designed to progressively increase technical and biological heterogeneity. Scenario 1 contains multiple experimental batches from a single laboratory using one microscope type and a small set of replicated compounds. Scenario 2 extends this setting to multiple laboratories using the same microscope type. Scenario 3 further increases biological diversity by including a large number of compounds with fewer replicates. Scenario 4 introduces multiple microscope types across laboratories while retaining the small replicated compound set. Scenario 5 represents the most challenging setting, combining multiple laboratories, multiple microscope types, a large compound collection, and limited replicate availability. This design allows us to evaluate batch correction under increasingly complex technical sources, including batch, laboratory, microscope, and compound-scale variation.

For BBBC datasets, we evaluate BBBC022 and BBBC036 as two separate dataset-level scenarios. Both datasets profile U2OS cells under compound perturbations, but they differ in scale and experimental design. BBBC022 is an early Cell Painting-style benchmark with bioactive compound treatments, while BBBC036 is a larger-scale compound profiling dataset with richer image and metadata resources. Following their original dataset organization, we report results separately for BBBC022 and BBBC036 rather than merging them into a single benchmark. This setting tests whether the method remains effective across independent compound-profiling datasets.

For CPG0036, we split the evaluation according to the available site and cell-line subsets. Specifically, we consider FMP_U2OS, FMP_HepG2, IMTM_HepG2, MEDINA_HepG2, and USC_HepG2 as separate scenarios. This split allows us to evaluate performance under different site and cell-line conditions while keeping the compound library relatively consistent. This dataset profiles the EU-OPENSCREEN Bioactive Compound Set across multiple European imaging sites, mainly in HepG2 cells, with an additional U2OS subset generated at FMP. Therefore, this split allows us to evaluate performance under different site and cell-line conditions while keeping the compound library relatively consistent.

For RxRx1, we follow its cell-type organization and evaluate four subsets separately: U2OS, HepG2, HUVEC, and RPE. RxRx1 contains Cell Painting-like fluorescence microscopy images across multiple experimental batches and is designed for evaluating batch-effect correction under genetic perturbations. By separating the four cell types, we assess whether the learned correction remains effective across distinct cellular contexts and perturbation distributions.

B Feature Representations

We evaluate our batch correction approach across multiple feature representations, including CellProfiler, DeepProfiler, and OpenPhenom, to demonstrate its robustness and effectiveness under diverse feature spaces.

B.1 CellProfiler

CellProfiler Stirling et al. (2021) is a widely used image analysis tool that converts raw Cell Painting images into quantitative morphological features. These manually defined and interpretable features describe the size, shape, texture, intensity, granularity, and radial distribution of painted cells, providing a comprehensive representation of cellular states for downstream analysis. Owing to their interpretability, reproducibility, and explicit connection to measurable cellular phenotypes, CellProfiler features remain a widely used and transparent representation for biomedical and pharmaceutical research.

B.2 DEEPPROFILER

DeepProfiler Moshkov et al. (2024) is a deep-learning-based profiling framework for extracting morphological representations from high-throughput Cell Painting images. Instead of relying on manually designed CellProfiler measurements, DeepProfiler uses convolutional neural networks to learn image-derived features directly from single-cell crops. These learned representations provide a data-driven alternative to handcrafted features and can capture subtle phenotypic variation that may be missed by classical morphology descriptors, supporting more sensitive and scalable perturbation profiling.

B.3 OPENPHENOM

OpenPhenom Kraus et al. (2024) is a self-supervised foundation-model approach for extracting features from high-content microscopy images. It uses masked autoencoding to learn cellular representations directly from large-scale microscopy data. During training, portions of the input image are masked and the model is optimized to reconstruct the missing regions, encouraging the encoder to capture general morphological structure, texture, and cellular organization. These features are beneficial because they are label-free, scalable to very large datasets, and capable of capturing rich morphological signals that support tasks such as perturbation retrieval and biological relationship prediction.

C BASELINE METHODS

Method	Category	Batch Labels Required
ComBat Johnson et al. (2007)	Statistical	Yes
Sphering Kessy et al. (2018)	Statistical	No
CCA Satija et al. (2015)	Linear Projection	Yes
RPCA Satija et al. (2015)	Linear Projection	No
MNN Haghverdi et al. (2018)	Graph-based / KNN	Yes
fastMNN Haghverdi et al. (2018)	Graph-based / KNN	Yes
Scanorama Hie et al. (2019)	Graph-based / KNN	Yes
Harmony Korsunsky et al. (2019)	Graph-based / KNN	Yes
scVI Lopez et al. (2018)	Deep Learning	Yes
DESC Li et al. (2020)	Deep Learning	Yes
scBCN Wan et al. (2025)	Deep Learning	Yes
cpDistiller Yan et al. (2025)	Deep Learning	Yes

Table 4: Comparison of baseline and competing batch correction methods.

To evaluate the batch correction performance of **HICON**, we consider a comprehensive set of baselines spanning statistical, linear projection, graph-based, and deep learning approaches, as summarized in Table 4.

C.1 STATISTICAL APPROACHES

ComBat Johnson et al. (2007): Removes batch effects by assuming that each batch introduces a systematic shift in the data distribution, mainly changing the mean and variance of each feature. It estimates these batch-specific effects with an empirical Bayes framework and then adjusts the data to a shared distribution, while preserving the biological variation of interest.

Sphering Kessy et al. (2018): Sphering is a covariance-based linear transformation that decorrelates features and rescales them to unit variance. In the batch-correction setting, the whitening transform is estimated from negative control samples and then applied to the full dataset, so that control-derived technical variation is removed more broadly across samples.

C.2 Linear Projection Approaches

Seurat CCA Satija et al. (2015): Aligns datasets by finding shared linear combinations of features that are maximally correlated across batches or datasets. It therefore projects cells into a common low-dimensional space where shared biological structure is emphasized and dataset-specific technical differences are reduced.

Seurat RPCA Satija et al. (2015): Finds integration anchors by projecting each dataset into the other's PCA space, providing a faster and more conservative alternative to CCA that is often preferred for large datasets or settings with greater heterogeneity.

C.3 Graph-based Approaches

MNN Haghverdi et al. (2018): Identifies mutual nearest-neighbor pairs across batches and uses these matched samples as anchors to estimate batch-correction vectors while preserving shared biological structure.

fastMNN Haghverdi et al. (2018): Corrects batch effects by identifying mutual nearest neighbors across batches in a low-dimensional PCA space and using these matched samples to estimate correction vectors. It assumes that shared subpopulations are present across batches, allowing similar samples to be aligned while preserving meaningful biological structure.

Scanorama Hie et al. (2019): Integrates heterogeneous datasets by performing approximate nearest-neighbor matching across all pairs of datasets in a low-dimensional space, then merging overlapping subpopulations into shared panoramas. This design relaxes strict overlap assumptions between individual batch pairs, helps reduce overcorrection, and scales efficiently to large collections of samples.

Harmony Korsunsky et al. (2019): Removes batch effects by iteratively clustering samples in a shared low-dimensional embedding and correcting batch-associated variation within clusters. By encouraging clusters to contain samples from multiple batches, it preserves biological structure while producing an integrated representation that is robust and scalable.

C.4 Deep Learning Approaches

scVI Lopez et al. (2018): Employs a deep generative latent-variable model to represent high-dimensional count data in a low-dimensional space while explicitly modeling technical variation such as batch effects and sampling depth. Built on variational inference and neural networks, it learns a shared latent representation that supports batch correction and downstream analyses in a scalable probabilistic framework.

DESC Li et al. (2020): Adopts a deep embedded clustering framework that first learns a low-dimensional representation with an autoencoder, then iteratively refines cluster assignments through self-training. By emphasizing high-confidence samples, it progressively suppresses batch-associated variation while preserving meaningful structure.

scBCN Wan et al. (2025): Integrates cross-batch similar-cluster identification with a residual neural network for batch correction. It expands mutual nearest-neighbor relationships, constructs a cluster-level similarity graph, and optimizes the embedding with a Tuplet Margin Loss to align similar clusters while separating dissimilar ones.

cpDistiller Yan et al. (2025): Introduces a three-stage framework consisting of feature extraction, joint representation learning, and technical correction to address batch, row, and column effects. It leverages both handcrafted and deep image features, and applies a semi-supervised GMVAE with contrastive and domain-adversarial learning to derive embeddings that preserve biologically relevant variation.

D Evaluation Metrics

We evaluate the corrected Cell Painting representations from three complementary perspectives: technical-effect removal, biological-structure preservation, and downstream biological utility. Unless otherwise stated, all metrics are defined such that higher values indicate better performance.

Following the benchmark protocol, the aggregate batch-correction score is computed from three technical metrics: Graph Connectivity, kBET, and batch LISI. The biological-preservation score is computed from four biological metrics: label LISI, Leiden ARI, Leiden NMI, and perturbation retrieval mAP. Specifically,

$$S_{\text{batch}} = \frac{S_{\text{GC}} + S_{\text{kBET}} + S_{\text{LISI-batch}}}{3},$$

and

$$S_{\text{bio}} = \frac{S_{\text{LISI-label}} + S_{\text{ARI}} + S_{\text{NMI}} + S_{\text{mAP}}}{4}.$$

To give equal weight to each of the seven underlying metrics, the Overall score is computed as

$$S_{\text{overall}} = \frac{3S_{\text{batch}} + 4S_{\text{bio}}}{7}.$$

Thus, the Overall score corresponds to the unweighted average over all three batch-correction metrics and four biological-preservation metrics, rather than an equal-weighted average of the two aggregate groups.

D.1 Batch-Correction and Biological-Preservation Metrics

Graph Connectivity. Graph Connectivity evaluates whether samples sharing the same biological label remain connected in the corrected kNN graph. Let $\mathcal{C}$ denote the set of biological labels and $\mathcal{V}_c$ the set of samples with label c. Let $\mathcal{G}_c$ be the subgraph induced by $\mathcal{V}_c$, and $\text{LCC}(\mathcal{G}_c)$ its largest connected component. The Graph Connectivity score is

$$S_{\text{GC}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{|\text{LCC}(\mathcal{G}_c)|}{|\mathcal{V}_c|}.$$

The score lies in $(0, 1]$, with larger values indicating better connectivity of biologically related samples after correction.

kBET. The k-nearest-neighbor Batch Effect Test (kBET) evaluates whether the local distribution of technical labels agrees with the corresponding global distribution. Let π_b denote the global proportion of samples from technical group b, and let n_{ib} denote the number of samples from group b among the k neighbors of sample i. For each tested sample, kBET uses the statistic

$$\chi_i^2 = \sum_b \frac{(n_{ib} - k\pi_b)^2}{k\pi_b}.$$

The null hypothesis that the local and global technical-label distributions agree is tested using the corresponding χ^2 test. Let r_{kBET} denote the fraction of tested neighborhoods in which the null hypothesis is rejected. We report the acceptance-based score

$$S_{\text{kBET}} = 1 - r_{\text{kBET}},$$

such that higher values indicate better local mixing across technical conditions.

LISI. The Local Inverse Simpson's Index (LISI) measures label diversity in the local neighborhood of each sample. Let q_{ig} denote the local probability of observing label g around sample i, obtained from the distance-weighted neighborhood. The sample-level LISI is

$$\text{LISI}_i = \frac{1}{\sum_g q_{ig}^2}.$$

We aggregate sample-level scores using the median, following the benchmark implementation.

For batch LISI, g indexes the B technical groups. Since stronger technical mixing corresponds to larger raw LISI, the normalized score is

$$S_{\text{LISI-batch}} = \frac{\text{median}_i(\text{LISI}_i^{\text{batch}}) - 1}{B - 1}.$$

For label LISI, g indexes the C biological labels. Here, biologically pure neighborhoods correspond to smaller raw LISI, and the score is therefore reversed:

$$S_{\text{LISI-label}} = \frac{C - \text{median}_i(\text{LISI}_i^{\text{label}})}{C - 1}.$$

Both scores are thus normalized to $[0, 1]$, where higher values indicate better performance.

Leiden ARI. We apply Leiden clustering to the corrected representation and compare the resulting cluster assignments with the biological perturbation labels. Let n_{uv} denote the number of samples assigned to Leiden cluster u and biological class v, with

$$a_u = \sum_v n_{uv}, \qquad b_v = \sum_u n_{uv}, \qquad N = \sum_{u,v} n_{uv}.$$

The Adjusted Rand Index (ARI) is

$$S_{\text{ARI}} = \frac{\sum_{u,v} \binom{n_{uv}}{2} - \frac{\sum_u \binom{a_u}{2} \sum_v \binom{b_v}{2}}{\binom{N}{2}}}{\frac{1}{2}\left[\sum_u \binom{a_u}{2} + \sum_v \binom{b_v}{2}\right] - \frac{\sum_u \binom{a_u}{2} \sum_v \binom{b_v}{2}}{\binom{N}{2}}}.$$

Higher ARI indicates stronger agreement between the learned clustering and the biological labels.

Leiden NMI. Let U denote the Leiden cluster assignment and V the biological perturbation label. Their mutual information is

$$I(U; V) = \sum_{u,v} p(u, v) \log \frac{p(u, v)}{p(u)p(v)},$$

with entropy

$$H(U) = -\sum_u p(u) \log p(u), \qquad H(V) = -\sum_v p(v) \log p(v).$$

We use the arithmetic-mean normalized mutual information

$$S_{\text{NMI}} = \frac{2I(U; V)}{H(U) + H(V)}.$$

The score ranges from 0 to 1, with larger values indicating stronger agreement between the unsupervised clusters and biological labels.

Perturbation Retrieval mAP (nonrep). We evaluate whether replicates of the same perturbation are retrieved before unrelated perturbations. Similarity between two corrected profiles is measured using cosine similarity,

$$s_{ij} = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\|_2 \|\mathbf{z}_j\|_2}.$$

For query i, profiles sharing the same perturbation label form the positive set $\mathcal{P}_i$, while non-replicate profiles from different perturbations form the negative set $\mathcal{N}_i$. Following the benchmark protocol

for mAP(nonrep), negatives are sampled from wells on the same plate with different perturbation identities.

After sorting candidate profiles in decreasing order of s_{ij}, let $r_i(k) \in \{0, 1\}$ indicate whether the item at rank k is positive. Precision at rank k is

$$P_i(k) = \frac{\sum_{t=1}^{k} r_i(t)}{k}.$$

Average precision for query i is

$$\mathrm{AP}_i = \frac{1}{|\mathcal{P}_i|} \sum_{k} P_i(k)\, r_i(k).$$

The perturbation retrieval score is obtained by averaging AP over eligible queries:

$$S_{\mathrm{mAP}} = \frac{1}{|\mathcal{Q}|} \sum_{i \in \mathcal{Q}} \mathrm{AP}_i.$$

Aggregate Scores. The aggregate batch-correction score is the unweighted average of Graph Connectivity, kBET, and batch LISI:

$$S_{\mathrm{batch}} = \frac{S_{\mathrm{GC}} + S_{\mathrm{kBET}} + S_{\text{LISI-batch}}}{3}.$$

The biological-preservation score is the unweighted average of label LISI, Leiden ARI, Leiden NMI, and perturbation retrieval mAP:

$$S_{\mathrm{bio}} = \frac{S_{\text{LISI-label}} + S_{\mathrm{ARI}} + S_{\mathrm{NMI}} + S_{\mathrm{mAP}}}{4}.$$

To assign equal weight to each of the seven underlying metrics, we compute

$$S_{\mathrm{overall}} = \frac{3S_{\mathrm{batch}} + 4S_{\mathrm{bio}}}{7}.$$

Equivalently, S_{overall} is the unweighted average of all seven individual metrics.

D.2 Downstream Biological Utility

For the downstream retrieval evaluations, we use the same cosine-similarity ranking and average-precision formulation defined above, while changing the positive and negative sets according to the biological task.

Phenotypic Activity against DMSO. For each treated profile, replicates of the same perturbation are treated as positives, while DMSO negative-control profiles form the negative set. The activity score is computed as

$$S_{\mathrm{activity}} = \frac{1}{|\mathcal{Q}_{\mathrm{act}}|} \sum_{i \in \mathcal{Q}_{\mathrm{act}}} \mathrm{AP}_i,$$

where AP_i is computed from the cosine-similarity ranking as defined above. A higher score indicates that perturbation replicates can be distinguished more reliably from DMSO controls.

Phenotypic Distinctiveness. Phenotypic distinctiveness evaluates whether replicates of the same perturbation can be retrieved against profiles from other perturbations. For each query i, we define

$$\mathcal{P}_i = \{j \neq i : p_j = p_i\},$$

and

$$\mathcal{N}_i = \{j : p_j \neq p_i\}.$$

The distinctiveness score is

$$S_{\mathrm{distinct}} = \frac{1}{|\mathcal{Q}_{\mathrm{distinct}}|} \sum_{i \in \mathcal{Q}_{\mathrm{distinct}}} \mathrm{AP}_i.$$

When an activity filter is applied, $\mathcal{Q}_{\mathrm{distinct}}$ is restricted to perturbations satisfying the activity criterion.

Mechanism-of-Action Retrieval. For MoA retrieval, profiles are ranked by cosine similarity. For query i, a candidate j is relevant if it shares the same MoA annotation,

$$r_i^{\mathrm{MoA}}(j) = \mathbb{I}[m_i = m_j],$$

where m_i denotes the MoA label. MoA mAP is computed using the same average-precision definition above.

Precision@k and Recall@k are

$$\mathrm{Precision@}k = \frac{\mathrm{TP}_k}{k}, \qquad \mathrm{Recall@}k = \frac{\mathrm{TP}_k}{|\mathcal{P}_i^{\mathrm{MoA}}|}.$$

For EOR@90, let τ_{90} denote the 90th percentile of pairwise cosine similarities and define a pair as highly similar if $s_{ij} \geq \tau_{90}$. Let n_{11} denote the number of highly similar pairs sharing an MoA, n_{10} the number of highly similar pairs with different MoAs, n_{01} the number of lower-similarity pairs sharing an MoA, and n_{00} the number of lower-similarity pairs with different MoAs. The enrichment odds ratio is

$$\mathrm{EOR@90} = \frac{n_{11} n_{00}}{n_{10} n_{01}}.$$

Larger values indicate stronger enrichment of same-MoA pairs among the most similar profiles.

E Controlled Synthetic Study of the Learned Technical Proxy

A central assumption of HICON is that the learned technical representation Z^{tech} can serve as a useful proxy for the unobserved technical state T. On real Cell Painting datasets, however, T is fundamentally unobserved: available batch, plate, and spatial metadata provide only coarse annotations and therefore cannot serve as ground truth for directly evaluating proxy fidelity. We therefore construct a controlled synthetic setting in which both biological and hierarchical technical factors are known by design. This allows us to directly study whether HICON can recover latent technical structure from the observed features when its metadata supervision becomes unreliable.

Synthetic construction. We generate 3,840 samples corresponding to 40 biological perturbations, with 96 replicates per perturbation. Technical variation is organized hierarchically into 4 batches and 12 plates. Biological, batch-level, and plate-level signals are generated from non-overlapping orthonormal subspaces, such that the underlying biological and technical factors are explicitly known while remaining jointly present in the observed feature representation.

Conceptually, each synthetic profile contains additive contributions from biological and hierarchical technical factors,

$$\mathbf{x}_i = \mathbf{x}_i^{\mathrm{bio}} + \mathbf{x}_i^{\mathrm{batch}} + \mathbf{x}_i^{\mathrm{plate}} + \boldsymbol{\epsilon}_i,$$

where the biological, batch, and plate components occupy mutually non-overlapping subspaces and $\boldsymbol{\epsilon}_i$ denotes residual variation.

To explicitly decouple the observed metadata G from the true technical state T, we progressively corrupt the complete batch–plate metadata tuple for a fraction ρ of the samples while leaving the generated features and the underlying technical states unchanged. We consider corruption ratios of 0%, 5%, 10%, 20%, 30%, and 40%. The ground-truth technical states are never provided to HICON during training and are used only for evaluation.

Evaluation protocol. We train HICON using the corrupted metadata as weak technical supervision. For each perturbation, 20 of the 96 replicates are held out for evaluation. Experiments are repeated over three random seeds.

To quantify how much information about the true technical state is retained by the learned proxy, we train linear classifiers on the learned batch- and plate-level technical representations and evaluate their ability to predict the uncorrupted ground-truth batch and plate states. We report balanced accuracy as mean $\pm$ standard deviation across the three runs. For reference, we also report in parentheses the accuracy of the corrupted metadata itself with respect to the corresponding ground-truth technical labels.

Table 5: Controlled synthetic evaluation of learned technical-proxy fidelity under increasing metadata corruption. Values report balanced accuracy (mean $\pm$ standard deviation over three runs) for predicting the ground-truth technical states from the learned proxy representations. Parentheses show the corresponding accuracy of the corrupted metadata used during training.

Corruption	Batch proxy	Plate proxy
0%	0.985 ± 0.003 (1.000)	0.967 ± 0.005 (1.000)
5%	0.963 ± 0.001 (0.951)	0.945 ± 0.003 (0.951)
10%	0.943 ± 0.012 (0.900)	0.918 ± 0.015 (0.901)
20%	0.910 ± 0.009 (0.800)	0.873 ± 0.007 (0.800)
30%	0.866 ± 0.025 (0.700)	0.830 ± 0.040 (0.701)
40%	0.794 ± 0.047 (0.599)	0.772 ± 0.038 (0.599)

Can the learned proxy recover the latent technical state? Table 5 shows that the learned technical proxy retains substantial information about the ground-truth technical states even when the metadata provided during training becomes increasingly unreliable. Under clean metadata, the learned batch and plate representations recover the corresponding technical states with balanced accuracies of 0.985 and 0.967, respectively. As metadata corruption increases, proxy accuracy decreases gradually, but remains consistently higher than the information contained in the corrupted metadata itself across all non-zero corruption levels.

The gap becomes particularly clear under stronger corruption. At 40% corruption, the corrupted metadata agrees with the true technical state only about 0.599 of the time, whereas the learned batch and plate proxies retain balanced accuracies of 0.794 and 0.772, respectively. This indicates that HICON does not simply reproduce the metadata labels used for supervision. Instead, the technical branch exploits the observed sample features together with weak metadata guidance to recover part of the underlying technical structure.

This controlled study provides direct empirical support for the proxy assumption used in the formulation of HICON. When the latent technical factors are known by construction, the learned proxy more faithfully reflects those factors than the corrupted metadata supplied during training. This complements the real-data metadata-corruption analysis, where the true technical state is unavailable and proxy fidelity can therefore only be assessed indirectly.

At the same time, this experiment should be interpreted as a controlled mechanistic study rather than as evidence that the true technical state can be fully recovered in real Cell Painting data. The synthetic factors are deliberately constructed to be identifiable and more regular than real experimental variation, where biological and technical effects may interact and overlap. The purpose of this setting is therefore to isolate and directly test the central proxy-learning mechanism under known ground truth.

F Additional Method Details

F.1 Approximation Error Induced by Proxy Conditioning

In the main text, the ideal objective is to preserve perturbation-relevant information after conditioning on the latent technical state:

$$\max_{F_{\text{bio}}} I(Z^{\text{bio}}; P \mid T),$$

where Z^{bio} denotes the learned biological representation, P denotes the perturbation label, and T denotes the unobserved technical state. Since T is not observed, HICON learns a technical proxy Z^{tech}. In practice, this proxy induces a discrete technical state estimate, denoted by $\widehat{T}$, through the inferred latent technical assignments. The resulting surrogate objective can be written as

$$I(Z^{\text{bio}}; P \mid \widehat{T}).$$

Below, we characterize the approximation gap introduced by replacing T with $\widehat{T}$.

Proposition. For any discrete random variables A, B, C, D,

$$|I(A; B \mid C) - I(A; B \mid D)| \leq H(C \mid D) + H(D \mid C).$$

By setting $A = Z^{\text{bio}}$, $B = P$, $C = T$, and $D = \widehat{T}$, we obtain

$$\left| I(Z^{\text{bio}}; P \mid T) - I(Z^{\text{bio}}; P \mid \widehat{T}) \right| \leq H(T \mid \widehat{T}) + H(\widehat{T} \mid T).$$

This bound shows that the gap between the ideal objective and the proxy-conditioned surrogate is controlled by the discrepancy between the latent technical state T and its learned proxy $\widehat{T}$. When $\widehat{T}$ preserves the technical information in T and avoids introducing unrelated technical states, both conditional entropy terms become small, and the proxy-conditioned objective becomes a close approximation to the ideal objective.

F.2 Proof sketch.

We decompose the difference as

$$I(A; B \mid C) - I(A; B \mid D) = [I(A; B \mid C) - I(A; B \mid C, D)] + [I(A; B \mid C, D) - I(A; B \mid D)].$$

Taking the absolute value and applying the triangle inequality gives

$$|I(A; B \mid C) - I(A; B \mid D)| \leq |I(A; B \mid C) - I(A; B \mid C, D)| + |I(A; B \mid D) - I(A; B \mid C, D)|.$$

By the chain rule of mutual information,

$$I(A; B \mid C) - I(A; B \mid C, D) = I(A; D \mid C) - I(A; D \mid B, C).$$

Therefore,

$$|I(A; B \mid C) - I(A; B \mid C, D)| \leq H(D \mid C).$$

Similarly,

$$|I(A; B \mid D) - I(A; B \mid C, D)| \leq H(C \mid D).$$

Combining the two inequalities yields

$$|I(A; B \mid C) - I(A; B \mid D)| \leq H(C \mid D) + H(D \mid C).$$

This completes the proof.

G Implementation Details

Network architecture. HICON is implemented with two MLP branches for biological and technical representation learning. The biological branch consists of an MLP trunk and a projection head. The trunk has a hidden dimension of 256 and an output dimension of 256, and the projection head has a hidden dimension of 256. The final corrected biological embedding has a dimension of 128. The technical branch uses a shared MLP encoder with level-specific MLP heads. The shared encoder has a hidden dimension of 256 and an output dimension of 128, and each level-specific head has a hidden dimension of 128. The technical embedding dimension is set to 64. For the hierarchical technical representation, we aggregate the batch-, plate-, and spatial-level embeddings with weights 1.0, 0.5, and 0.2, respectively.

Training settings. All models are trained for 500 epochs with a mini-batch size of 512. We use the Adam optimizer with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-4}. The learning rate is decayed using a StepLR scheduler, where the step size is set to one quarter of the total number of training epochs and the decay factor is 0.5. The random seed is fixed to 1234 for PyTorch, NumPy, and Python random initialization. Unless otherwise specified, we use the same training configuration across datasets and scenarios.

Technical hierarchy. The technical branch models three levels of technical variation: batch/source, plate, and spatial position. The batch metadata key is scenario-dependent. The plate-level metadata is defined by `Metadata_Plate`, and the spatial metadata is defined by `Metadata_Row` and `Metadata_Column`. The assignment temperature is set to 0.1 for all three levels. The level-wise technical loss weights are set to 2.0, 1.0, and 0.2 for the batch, plate, and spatial levels, respectively. This places stronger emphasis on coarse batch-level variation while still preserving the ability to model fine-grained spatial effects.

Table 6: Computational cost comparison. Inference time is measured on 4,068 test samples.

Method	Training/Fitting (s)	Inference (s)	Time / 1K samples (s)	Params.
ComBat	0.376 ± 0.002	N.A.	N.A.	N.A.
Scanorama	13.201 ± 0.062	N.A.	N.A.	N.A.
Harmony	463.846 ± 32.965	N.A.	N.A.	N.A.
cpDistiller	100.216 ± 3.313	0.0084 ± 0.0008	0.0021 ± 0.0002	2.445M
scVI	30.860 ± 0.647	0.0223 ± 0.0004	0.0055 ± 0.0001	0.703M
HICON	90.430 ± 0.480	0.0104 ± 0.0009	0.0026 ± 0.0002	1.200M

Confidence-weighted technical proxy learning. For technical proxy learning, we use a two-stage training strategy. During the first 50% of training steps, metadata labels are treated as high-confidence supervision. After this stage, we enable sample-level confidence weighting based on metadata-consistency scores. This allows the model to adaptively adjust the supervision strength according to how consistent each sample is with its metadata label. The one-dimensional two-component GMM used for confidence estimation is fitted with 7 EM iterations. The technical component centers are updated using an EMA schedule, with the momentum increasing from 0.85 to 0.999 for the batch level, from 0.84 to 0.998 for the plate level, and from 0.82 to 0.99 for the row/column level.

Technically conditioned contrastive learning. The biological branch uses `Metadata_JCP2022` as the perturbation label for supervised contrastive learning. The contrastive temperature is set to 0.1. During the first 30% of training steps, pair reweighting is disabled to allow the biological and technical branches to form stable initial representations. After this warm-up stage, technical-similarity-based pair reweighting is applied to both positive and negative pairs. Both positive-pair and negative-pair weights are clipped to the interval $[0.6, 1.4]$. We also use replicate-count reweighting to reduce the bias introduced by highly replicated perturbations, with the reweighting parameter set to 2.0 and the lower bound set to 1.0.

Feature-space augmentation. We use feature-space augmentation to construct additional positive views, especially for perturbations with limited replicates. The augmented view is generated by adding Gaussian noise with a standard deviation of 0.25 and applying feature dropout with a probability of 0.10. For singleton compounds, we force the construction of an augmented positive view using the same noise and dropout settings. By default, within-compound mixup is disabled. When enabled, mixup uses a Beta distribution with $\alpha = 0.4$ and is applied with a probability of 0.6.

Loss configuration. We set the weights of the technical proxy learning loss, biological contrastive loss, and CLUB loss to 1.0, 1.0, and 0.4, respectively. Within the technical proxy learning loss, we use the level-wise weights specified above for the batch-, plate-, and spatial-level technical objectives.

G.1 Complete Training Procedure

G.2 Computational Cost and Scalability

We compare the computational cost of HICON with representative batch-correction methods using 6,191 training samples and a fixed test set of 4,068 samples. Training and inference times are reported as mean $\pm$ standard deviation across repeated runs. Table 6 summarizes the results.

HICON has moderate training cost and fast inference, while using fewer parameters than cpDistiller. The confidence-aware GMM is updated using EM with analytical parameter updates and therefore does not require an additional gradient-based optimization procedure.

We further evaluate scalability by increasing the number of training samples from 1,990 to 6,191, corresponding to a $3.11\times$ increase in data size. The resulting training-time growth is shown in Table 7.

Algorithm 1 Training Procedure of HICON

Require: Cell Painting profiles $\{x_i\}$, perturbation labels $\{p_i\}$, hierarchical metadata $\{g_i^{(l)}\}_{l=0}^{2}$; biological encoder F_{bio}; technical encoder f_s and level-specific heads $\{f_l\}_{l=0}^{2}$; technical prototypes $\{\mu_k^{(l)}\}$.

1: **for** each training iteration τ **do**
2: Sample a mini-batch $\mathcal{B}$.
3: Compute biological representations

$$z_i^{\text{bio}} = F_{\text{bio}}(x_i).$$

4: Compute hierarchical technical representations

$$z_i^{(l)} = f_l(f_s(x_i)), \qquad l = 0, 1, 2,$$

and aggregate

$$z_i^{\text{tech}} = z_i^{(0)} + \rho_1 z_i^{(1)} + \rho_2 z_i^{(2)}.$$

5: **for** each technical level l **do**
6: Compute feature-inferred posterior assignments $\gamma_i^{(l)}$ using the latent technical mixture model.
7: Estimate metadata consistency

$$s_i^{(l)} = \gamma_{i,g_i^{(l)}}^{(l)},$$

and infer confidence $q_i^{(l)}$ using the two-component GMM.
8: Compute confidence weight

$$w_i^{(l)} = \max\Big(q_i^{(l)}, \eta_l(\tau)\Big).$$

9: Construct adaptive targets

$$t_i^{(l)} = w_i^{(l)}\,\text{onehot}(g_i^{(l)}) + (1 - w_i^{(l)})\,\text{stopgrad}(\gamma_i^{(l)}).$$

10: Update technical prototypes using the confidence-weighted responsibility mean.
11: **end for**
12: Compute the hierarchical technical loss $\mathcal{L}_{\text{tech}}$.
13: Construct positive and negative sets using perturbation labels:

$$\mathcal{P}(i) = \{j \neq i : p_j = p_i\}, \qquad \mathcal{N}(i) = \{k : p_k \neq p_i\}.$$

14: Compute pairwise technical weights

$$\beta_{ik} = r\big(\langle \tilde{z}_i^{\text{tech}}, \tilde{z}_k^{\text{tech}} \rangle\big).$$

15: Compute technically conditioned contrastive loss $\mathcal{L}_{\text{bio}}$ using β_{ik}.
16: Compute biological–technical disentanglement loss $\mathcal{L}_{\text{CLUB}}$.
17: Combine objectives:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{tech}}\mathcal{L}_{\text{tech}} + \lambda_{\text{bio}}\mathcal{L}_{\text{bio}} + \lambda_{\text{CLUB}}\mathcal{L}_{\text{CLUB}}.$$

18: Update trainable model parameters by minimizing $\mathcal{L}_{\text{total}}$.
19: **end for**

Ensure: Corrected biological representation Z^{bio}.

HICON's training time increases by $2.83\times$ as the dataset size increases by $3.11\times$, indicating favorable empirical scalability in this setting.

Table 7: Training-time growth when increasing the training set from 1,990 to 6,191 samples.

Method	Time growth
ComBat	3.00×
Scanorama	3.97×
Harmony	16.21×
cpDistiller	4.73×
scVI	3.10×
HICON	**2.83**×

H TECHNICAL HIERARCHY

H.1 POSTERIOR ASSIGNMENT UNDER THE SPHERICAL GMM

At each hierarchical level ℓ, the level-wise technical embedding is modeled by a spherical Gaussian mixture:

$$p(z^{(\ell)}) = \sum_{k=1}^{K_\ell} \pi_k^{(\ell)} \mathcal{N}\left(z^{(\ell)} \mid \mu_k^{(\ell)}, \sigma_\ell^2 I\right), \tag{6}$$

where K_ℓ is the number of latent technical components, $\pi_k^{(\ell)}$ is the mixture prior, $\mu_k^{(\ell)}$ is the prototype of component k, and σ_ℓ^2 is the shared isotropic variance.

Let $\hat{z}_i^{(\ell)} = z_i^{(\ell)} / \|z_i^{(\ell)}\|_2$ denote the normalized embedding of sample i. Under a uniform prior $\pi_k^{(\ell)} = 1/K_\ell$, the posterior responsibility of component k for sample i is

$$\gamma_{ik}^{(\ell)} = p(k \mid \hat{z}_i^{(\ell)}) = \frac{\exp\left(-\frac{1}{2\sigma_\ell^2}\left\|\hat{z}_i^{(\ell)} - \mu_k^{(\ell)}\right\|_2^2\right)}{\sum_{j=1}^{K_\ell} \exp\left(-\frac{1}{2\sigma_\ell^2}\left\|\hat{z}_i^{(\ell)} - \mu_j^{(\ell)}\right\|_2^2\right)}. \tag{7}$$

When both $\hat{z}_i^{(\ell)}$ and $\mu_k^{(\ell)}$ are ℓ_2-normalized, we have

$$\left\|\hat{z}_i^{(\ell)} - \mu_k^{(\ell)}\right\|_2^2 = \|\hat{z}_i^{(\ell)}\|_2^2 + \|\mu_k^{(\ell)}\|_2^2 - 2\hat{z}_i^{(\ell)\top}\mu_k^{(\ell)} = 2 - 2\hat{z}_i^{(\ell)\top}\mu_k^{(\ell)}. \tag{8}$$

Substituting this expression into the posterior gives

$$\gamma_{ik}^{(\ell)} = \frac{\exp\left(-\frac{1}{2\sigma_\ell^2}\left(2 - 2\hat{z}_i^{(\ell)\top}\mu_k^{(\ell)}\right)\right)}{\sum_{j=1}^{K_\ell} \exp\left(-\frac{1}{2\sigma_\ell^2}\left(2 - 2\hat{z}_i^{(\ell)\top}\mu_j^{(\ell)}\right)\right)} \tag{9}$$

$$= \frac{\exp\left(\frac{\hat{z}_i^{(\ell)\top}\mu_k^{(\ell)}}{\sigma_\ell^2}\right)}{\sum_{j=1}^{K_\ell} \exp\left(\frac{\hat{z}_i^{(\ell)\top}\mu_j^{(\ell)}}{\sigma_\ell^2}\right)} = \mathrm{softmax}_k\left(\frac{\hat{z}_i^{(\ell)\top}\mu_k^{(\ell)}}{\tau_\ell}\right), \tag{10}$$

where $\tau_\ell = \sigma_\ell^2$. Therefore, under normalized embeddings and prototypes, the posterior responsibility is equivalent to a similarity-based softmax over latent technical prototypes.

H.2 CONFIDENCE ESTIMATION FROM METADATA CONSISTENCY

The metadata-consistency score is defined as

$$s_i^{(\ell)} = \gamma_{i,g_i^{(\ell)}}^{(\ell)} = p\left(k = g_i^{(\ell)} \mid \hat{z}_i^{(\ell)}\right), \tag{11}$$

where $g_i^{(\ell)}$ denotes the metadata group index of sample i at level ℓ. This score measures the posterior mass assigned to the metadata-indexed component. A high value indicates that the feature-inferred

assignment agrees with the metadata anchor, whereas a low value indicates a mismatch between the observed metadata and the inferred latent technical structure.

This design is motivated by the assumption that metadata is informative but incomplete. Let G denote the observed metadata and T denote the latent technical state. We assume that G contains useful information about T, but does not fully determine it:

$$I(T; G) > 0, \qquad H(T \mid G) > 0.$$

Moreover, the observed profile X may still contain technical information not captured by metadata:

$$I(T; X \mid G) > 0.$$

Therefore, metadata can provide weak technical supervision, while feature-inferred assignments are needed to capture residual latent technical variation.

To estimate the confidence of metadata supervision, we fit a two-component Gaussian mixture model to the metadata-consistency scores:

$$p\Big(s_i^{(\ell)}\Big) = \sum_{r\in\{0,1\}} p(r)\, p\Big(s_i^{(\ell)} \mid r\Big), \tag{12}$$

where r is a latent binary variable indicating whether the sample is metadata-consistent. Specifically, $r = 1$ corresponds to the high-consistency component, while $r = 0$ corresponds to the low-consistency component. Concretely, we model

$$p\Big(s_i^{(\ell)}\Big) = \alpha_\ell \mathcal{N}\left(s_i^{(\ell)} \mid \mu_{\text{con}}^{(\ell)}, \sigma_{\text{con}}^{2(\ell)}\right) + (1-\alpha_\ell)\mathcal{N}\left(s_i^{(\ell)} \mid \mu_{\text{inc}}^{(\ell)}, \sigma_{\text{inc}}^{2(\ell)}\right), \tag{13}$$

where the two components correspond to metadata-consistent and metadata-inconsistent samples, respectively.

The posterior probability of belonging to the metadata-consistent component is used as the confidence estimate:

$$q_i^{(\ell)} = p(r = 1 \mid s_i^{(\ell)}). \tag{14}$$

The final confidence weight is defined as

$$w_i^{(\ell)} = \max\left(q_i^{(\ell)}, \eta_\ell(\tau)\right), \tag{15}$$

where $\eta_\ell(\tau)$ is a training-dependent lower bound and τ denotes the training step. This lower bound prevents overly small confidence weights during early training, when the component means and posterior assignments are still unstable.

The adaptive target used in the main text can be interpreted as a confidence-weighted interpolation between metadata supervision and feature-inferred posterior assignment:

$$t_i^{(\ell)} = w_i^{(\ell)} \text{onehot}\left(g_i^{(\ell)}\right) + \left(1 - w_i^{(\ell)}\right) \text{stopgrad}\left(\gamma_i^{(\ell)}\right). \tag{16}$$

High-confidence samples are guided more strongly by metadata labels, while low-confidence samples rely more on feature-inferred posterior assignments.

H.3 Prototype Update as Online Re-estimation

We clarify the relation between our prototype update and mean re-estimation in standard mixture models. For clarity, we first omit the hierarchy index ℓ.

In a Gaussian mixture model, EM alternates between estimating posterior responsibilities and updating component means. In the E-step, the posterior responsibility of component k for sample i is

$$\gamma_{ik} = p(k \mid \hat{z}_i), \tag{17}$$

which measures the soft membership of sample i to component k. Given these responsibilities, the M-step updates the component mean by a responsibility-weighted empirical average:

$$\mu_k^{\text{EM}} = \frac{\sum_i \gamma_{ik} \hat{z}_i}{\sum_i \gamma_{ik}}. \tag{18}$$

Our prototype update follows the same re-estimation principle, but adapts it to weak metadata supervision and online representation learning. At level ℓ, the prototype is updated as

$$\mu_k^{(\ell)} \leftarrow m_\ell(\tau)\mu_k^{(\ell)} + (1 - m_\ell(\tau)) \frac{\sum_i w_i^{(\ell)} \gamma_{ik}^{(\ell)} \hat{z}_i^{(\ell)}}{\sum_i w_i^{(\ell)} \gamma_{ik}^{(\ell)}}. \tag{19}$$

Compared with the standard EM mean update, this update introduces a confidence weight $w_i^{(\ell)}$ to reduce the influence of samples whose metadata labels are inconsistent with the inferred latent structure. Since the encoder and embeddings are updated during neural network training, we use an online momentum update instead of recomputing a full-dataset M-step at each iteration. The momentum term $m_\ell(\tau)$ stabilizes prototype estimation and reduces oscillation during training.

Thus, the update combines three effects. The posterior responsibility $\gamma_{ik}^{(\ell)}$ provides feature-inferred soft membership to latent technical components. The confidence weight $w_i^{(\ell)}$ suppresses samples with low metadata consistency. The momentum term $m_\ell(\tau)$ smooths online prototype updates. Therefore, our update can be interpreted as an online, confidence-weighted variant of EM-style mean re-estimation.

H.4 Remark on Proxy Quality

The technical branch is designed to learn a representation Z^{tech} that serves as a surrogate for the latent technical variable T. Since Z^{tech} is a continuous representation, we consider the discrete proxy state $\widehat{T}$ induced by Z^{tech}, for example through the posterior assignment over latent technical prototypes.

An interpretable measure of proxy quality is the substitution gap between conditioning on the true technical state T and conditioning on the proxy state $\widehat{T}$:

$$H(T \mid \widehat{T}) + H(\widehat{T} \mid T). \tag{20}$$

The first term measures technical information in T that is missing from the proxy, while the second term measures proxy information that is not explained by the true technical state.

Because T is unobserved, our training objective does not directly optimize this quantity. Instead, the technical branch improves the proxy indirectly by combining metadata guidance with feature-inferred latent assignments. In reliable regions, the proxy remains aligned with observed metadata. In inconsistent regions, it can adapt to the latent technical structure inferred from features. Therefore, the learned technical proxy is expected to provide a more faithful conditioning variable than raw metadata alone.

I Connection to Conditional Mutual Information

I.1 Ideal Objective

Let $x_i \in \mathbb{R}^d$ denote a Cell Painting profile, p_i its perturbation label, and t_i the latent technical state that gives rise to batch-, plate-, and spatial-level technical variation. HICON learns a biological representation

$$z_i^{\text{bio}} = F_{\text{bio}}(x_i)$$

and a technical representation

$$z_i^{\text{tech}} = F_{\text{tech}}(x_i).$$

Ideally, the corrected biological representation should preserve perturbation-relevant information after accounting for latent technical variation. This motivates the conditional mutual information objective

$$\max_{F_{\text{bio}}} I(Z^{\text{bio}}; P \mid T), \tag{21}$$

where P denotes perturbation identity and T denotes the latent technical state.

Since T is unobserved, direct optimization of equation 21 is infeasible. HICON therefore learns a technical proxy Z^{tech}. In practice, the hierarchical technical branch further induces prototype-level technical assignments, denoted as C^{tech}, through the posterior responsibilities of the technical

mixture model. We use C^{tech} as a tractable proxy for the latent technical state T, leading to the proxy objective

$$\max I(Z^{\text{bio}}; P \mid C^{\text{tech}}). \tag{22}$$

This approximation is valid to the extent that C^{tech} captures the technical variation relevant to the relationship between Z^{bio} and P.

I.2 Replicate-Level Conditional Mutual Information

Directly deriving a contrastive objective for $I(Z^{\text{bio}}; P \mid C^{\text{tech}})$ is difficult because P is a discrete perturbation label and the number of perturbations can be large. Instead, HICON uses perturbation replicates as biological supervision. For an anchor sample i, let j be a replicate with the same perturbation label, i.e., $p_j = p_i$. We define

$$U = Z_i^{\text{bio}}, \qquad V = Z_j^{\text{bio}}, \qquad C = C_i^{\text{tech}}.$$

The replicate-level conditional mutual information is

$$I(U; V \mid C) = \mathbb{E}_{p(u,v,c)} \left[\log \frac{p(v \mid u, c)}{p(v \mid c)} \right]. \tag{23}$$

This quantity measures how much information two perturbation replicates share after accounting for the technical condition. If technical variation is properly conditioned out, the remaining shared information between U and V is expected to mainly reflect perturbation-induced biological variation.

I.3 Conditional InfoNCE Lower Bound

We next show that a conditional contrastive objective lower-bounds the replicate-level CMI in equation 23. For each anchor u and condition c, suppose we have one positive sample

$$v_1 \sim p(v \mid u, c)$$

and $K - 1$ negative samples

$$v_2, \ldots, v_K \sim p(v \mid c).$$

Let $f(u, v)$ be a critic function. The conditional InfoNCE loss is

$$\mathcal{L}_{\text{cNCE}} = -\mathbb{E} \left[\log \frac{\exp(f(u, v_1))}{\sum_{k=1}^{K} \exp(f(u, v_k))} \right]. \tag{24}$$

Following the standard InfoNCE argument, the optimal critic satisfies

$$f^*(u, v) = \log \frac{p(v \mid u, c)}{p(v \mid c)} + \text{const}. \tag{25}$$

Therefore, the conditional InfoNCE objective provides the lower bound

$$I(U; V \mid C) \geq \log K - \mathcal{L}_{\text{cNCE}}. \tag{26}$$

Thus, minimizing $\mathcal{L}_{\text{cNCE}}$ maximizes a lower bound of the replicate-level conditional mutual information.

I.4 Technically Conditioned Contrastive Surrogate

The bound in equation 26 assumes that negative samples are drawn from the conditional marginal distribution $p(v \mid c)$. However, exact conditional sampling is not available in minibatch training, especially because the latent technical state is not directly observed. HICON therefore uses the learned technical proxy to approximate conditional sampling by reweighting pairwise comparisons according to inferred technical similarity.

Specifically, for an anchor i and a candidate sample k, HICON defines a technical similarity weight

$$\beta_{ik} = r \left(\left\langle \frac{\text{sg}(z_i^{\text{tech}})}{\|z_i^{\text{tech}}\|_2}, \frac{\text{sg}(z_k^{\text{tech}})}{\|z_k^{\text{tech}}\|_2} \right\rangle \right), \tag{27}$$

where $r(\cdot)$ maps the similarity score to a bounded positive range, and $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation. Intuitively, β_{ik} is larger when samples i and k are inferred to have more comparable technical states.

Let $\mathcal{P}(i)$ denote the set of positive samples sharing the same perturbation as anchor i, and let $\mathcal{A}(i)$ denote all candidate samples in the minibatch. The biological similarity is defined as

$$s_{ik} = \mathrm{sim}\left(z_i^{\mathrm{bio}}, z_k^{\mathrm{bio}}\right).$$

We define the temperature-scaled biological similarity as

$$a_{ik} = \frac{\mathrm{sim}\left(z_i^{\mathrm{bio}}, z_k^{\mathrm{bio}}\right)}{\tau_{\mathrm{bio}}},$$

where τ_{bio} is the contrastive temperature. The technically conditioned contrastive loss is then defined as

$$\mathcal{L}_{\mathrm{bio}} = -\frac{1}{N_b}\sum_{i=1}^{N_b}\log\frac{\sum_{j\in\mathcal{P}(i)}\beta_{ij}\exp(a_{ij})}{\sum_{j\in\mathcal{P}(i)}\beta_{ij}\exp(a_{ij}) + \sum_{k\in\mathcal{N}(i)}\beta_{ik}\exp(a_{ik})}, \tag{28}$$

where $\mathcal{P}(i)$ and $\mathcal{N}(i)$ denote the positive and negative sets defined by perturbation identity.

equation 28 can be viewed as a practical approximation to equation 24. The weights β_{ik} increase the contribution of candidate samples under comparable inferred technical conditions and down-weight comparisons dominated by technical mismatch. Equivalently, the technical weighting can be incorporated into the pairwise contrastive score as

$$\tilde{f}(z_i^{\mathrm{bio}}, z_k^{\mathrm{bio}}, z_i^{\mathrm{tech}}, z_k^{\mathrm{tech}}) = a_{ik} + \log\beta_{ik}. \tag{29}$$

Since InfoNCE-type lower bounds hold for any critic, equation 28 serves as a technically conditioned contrastive surrogate for maximizing $I(U;V \mid C)$. When β_{ik} approximates the relative likelihood that sample k belongs to the same technical condition as anchor i, this weighted objective approximates contrastive learning with negatives drawn from $p(v \mid c_i)$.

Therefore, under the technical-proxy approximation, minimizing $\mathcal{L}_{\mathrm{bio}}$ maximizes a practical lower-bound surrogate of the replicate-level conditional mutual information:

$$I(Z_i^{\mathrm{bio}}; Z_j^{\mathrm{bio}} \mid C_i^{\mathrm{tech}}) \gtrsim \log K - \mathcal{L}_{\mathrm{bio}}, \tag{30}$$

where $\gtrsim$ indicates that the exact conditional sampling distribution is approximated by technical-similarity weighting.

I.5 Relation to Perturbation-Level Conditional Mutual Information

The ideal objective in equation 22 concerns the perturbation label P, whereas equation 30 is defined over replicate representations. These two quantities are connected through the standard replicate assumption: two profiles of the same perturbation share biological information primarily through their perturbation identity. Formally, assuming that replicate representations are conditionally independent given the perturbation and technical condition,

$$Z_i^{\mathrm{bio}} \perp Z_j^{\mathrm{bio}} \mid (P, C^{\mathrm{tech}}), \tag{31}$$

the shared information between replicate representations after conditioning on C^{tech} is mediated by P. By the conditional data processing inequality, we have

$$I(Z_i^{\mathrm{bio}}; Z_j^{\mathrm{bio}} \mid C^{\mathrm{tech}}) \le I(Z_i^{\mathrm{bio}}; P \mid C^{\mathrm{tech}}). \tag{32}$$

Thus, maximizing replicate-level conditional mutual information promotes the desired perturbation-level objective $I(Z^{\mathrm{bio}}; P \mid C^{\mathrm{tech}})$. Since C^{tech} is learned as a proxy for the latent technical state T, this further serves as a practical surrogate for the ideal objective $I(Z^{\mathrm{bio}}; P \mid T)$.

I.6 Dependence Penalty Between Biological and Technical Representations

A variational view of conditional mutual information also motivates reducing the residual dependence between biological and technical representations. For general random variables U, V, C, the conditional mutual information can be written as

$$I(U; V \mid C) = \mathbb{E}_{p(u,v,c)} \left[\log \frac{p(c)p(v, c \mid u)}{p(c \mid u)p(v, c)} \right]. \tag{33}$$

Using an energy-based variational distribution to approximate $p(v, c \mid u)$ gives a lower bound of the form

$$I(U; V \mid C) \geq \mathbb{E}_{p(u,v,c)}[h(u, v, c)] - e^{-1}\mathbb{E}_{p(u)}\mathbb{E}_{p(v,c)} \left[e^{h(u,v,c)} \right] - I(U; C), \tag{34}$$

where h is a critic function. The last term, $-I(U; C)$, shows that maximizing the lower bound encourages the anchor representation U to be less dependent on the conditioning variable C.

In HICON, U corresponds to Z^{bio}, and the technical condition C is represented by the learned technical proxy Z^{tech}. Therefore, we reduce residual technical dependence in the biological branch by minimizing an upper bound on

$$I(Z^{\mathrm{bio}}; Z^{\mathrm{tech}}).$$

Specifically, HICON uses the CLUB objective as a tractable upper bound for this mutual information term:

$$I(Z^{\mathrm{bio}}; Z^{\mathrm{tech}}) \leq \mathcal{L}_{\mathrm{CLUB}}. \tag{35}$$

Minimizing $\mathcal{L}_{\mathrm{CLUB}}$ discourages technical information from leaking into the biological representation.

I.7 Overall Interpretation

Combining the above arguments, HICON can be interpreted as optimizing a practical surrogate for conditional mutual information maximization. The biological contrastive loss $\mathcal{L}_{\mathrm{bio}}$ approximates conditional contrastive learning under comparable inferred technical states. The technical proxy loss $\mathcal{L}_{\mathrm{tech}}$ improves the quality of the learned conditioning variable Z^{tech} or C^{tech}. The CLUB loss $\mathcal{L}_{\mathrm{CLUB}}$ reduces residual dependence between biological and technical representations. The full objective is

$$\mathcal{L}_{\mathrm{total}} = \lambda_{\mathrm{bio}}\mathcal{L}_{\mathrm{bio}} + \lambda_{\mathrm{tech}}\mathcal{L}_{\mathrm{tech}} + \lambda_{\mathrm{CLUB}}\mathcal{L}_{\mathrm{CLUB}}. \tag{36}$$

Rather than exactly optimizing $I(Z^{\mathrm{bio}}; P \mid T)$, this objective provides a tractable technically conditioned contrastive surrogate for preserving perturbation-relevant information while accounting for latent technical variation.

J Compute Resources

All experiments were run on a single NVIDIA RTX 6000 Ada GPU with 48GB memory. Since HICON operates on pre-extracted Cell Painting feature representations, including CellProfiler, DeepProfiler, and OpenPhenom features, one GPU is sufficient for training and evaluation across all benchmark settings. The computational cost is mainly dominated by training the HICON representation-learning model, updating hierarchical technical prototypes, computing technically conditioned contrastive losses, and running downstream evaluations.

Execution time depends on the dataset size and evaluation setting. For smaller benchmarks such as BBBC022 and BBBC036, a single HICON training run typically takes approximately 1–3 hours. For larger JUMP-CP scenarios, a single run typically takes approximately 6–12 hours, depending on the number of samples and perturbation replicates. Metadata-corruption and ablation experiments require multiple runs over different corruption ratios or model variants, but each individual run can be completed on a single 48GB GPU.

K Downstream Analysis Experiments

We provide additional details for the downstream biological utility evaluations. While the main text reports aggregate trends for phenotypic activity against DMSO and phenotypic distinctiveness, here

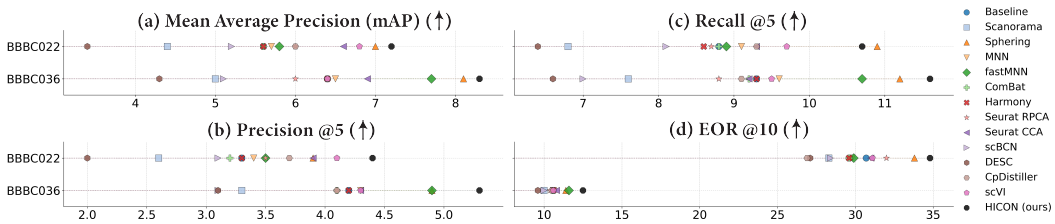

Figure 8: MoA retrieval metrics for DeepProfiler features on BBBC022 and BBBC036 datasets.

we further examine mechanism-of-action (MoA) retrieval. This task evaluates whether corrected Cell Painting representations preserve higher-level biological relationships between compounds that share similar annotated mechanisms, rather than only preserving replicate-level similarity.

We evaluate MoA retrieval on BBBC022 and BBBC036 using DeepProfiler features, where compound-level mechanism annotations are available. For each profile, we use its corrected biological representation as a query and rank all candidate profiles by cosine similarity in the corrected feature space. Profiles associated with the same MoA annotation are treated as relevant matches. We report mean average precision (mAP), Precision@5, Recall@5, and enrichment odds ratio at the 90th-percentile similarity threshold (EOR@90). These metrics evaluate complementary aspects of retrieval quality: mAP measures the overall ranking of relevant MoA matches, Precision@5 and Recall@5 focus on the top retrieved profiles, and EOR@90 measures whether compound pairs sharing the same MoA are enriched among the top 10% most similar pairs relative to the remaining pairs.

As shown in Fig. 8, HICON achieves strong MoA retrieval performance on both BBBC022 and BBBC036. In particular, HICON is consistently competitive across mAP, Precision@5, Recall@5, and EOR@90, indicating that the corrected representations preserve mechanism-level neighborhood structure after batch correction. This is important because MoA retrieval is a stricter downstream test than generic batch-mixing metrics: a representation must remove technical variation while still maintaining biologically meaningful similarities between compounds with related mechanisms.

These results further support the main claim that HICON does not simply mix batches or collapse technical groups. Instead, by conditioning contrastive learning on the learned technical proxy, HICON preserves perturbation-relevant morphology in the biological representation. The MoA retrieval results therefore complement the phenotypic activity and phenotypic distinctiveness evaluations, showing that HICON improves not only benchmark-level correction scores but also downstream biological utility for mechanism-level compound analysis.

L Additional Robustness and Generalization Experiments

L.1 Sensitivity to Perturbation Replicate Availability

HICON uses perturbation replicates to construct biological positive pairs for technically conditioned contrastive learning. We therefore further examine how its performance changes when fewer perturbation replicates are available during training.

We conduct this analysis on JUMP-CP Scenario 4 by progressively limiting the maximum number of training replicates available for each perturbation. The full training set contains an average of 70.2 replicates per perturbation. We compare this full setting with replicate limits of 32, 16, 8, 4, 2, and 1, while keeping the remaining training and evaluation protocol unchanged.

As expected, performance decreases as replicate availability is reduced (Table 8). The degradation is nevertheless gradual rather than abrupt. Reducing the number of replicates from the full setting to 32 produces only a small change in Overall performance ($0.687 \rightarrow 0.682$), whereas further reductions progressively decrease the available replicate-based supervision. Even in the extremely sparse single-replicate setting, performance decreases to 0.548 rather than showing an abrupt collapse.

Table 8: Sensitivity of HICON to perturbation replicate availability on JUMP-CP Scenario 4. The full training set contains an average of 70.2 replicates per perturbation. Overall denotes the aggregate score over the batch-correction and biological-preservation metrics. Higher is better.

#Replicates per perturbation	Overall
Full (mean: 70.2)	**0.687**
32	0.682
16	0.619
8	0.603
4	0.596
2	0.564
1	0.548

Method	Batch correction			Bio metrics				Aggregate score		
	Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
scBCN	0.81	0.33	0.34	0.91	0.12	0.49	0.65	0.49	0.54	0.51
cpDistiller	0.75	0.84	0.21	0.89	0.06	0.33	0.58	0.60	0.47	0.53
scVI	0.81	0.49	0.22	0.91	0.19	0.60	0.71	0.51	0.60	0.56
DESC	0.59	**0.91**	**0.47**	0.91	0.08	0.36	0.61	0.66	0.49	0.57
HICON	**0.92**	0.84	0.26	**0.96**	**0.21**	**0.64**	**0.79**	**0.67**	**0.65**	**0.66**

Table 9: Cross-domain generalization under source and compound shifts across microscopes.

This trend is consistent with the role of perturbation replicates in HICON. For a perturbation with n_p replicates, the number of distinct replicate pairs grows as

$$N_p^+ = \binom{n_p}{2} = \frac{n_p(n_p-1)}{2}.$$

Consequently, reducing replicate availability removes biological pairwise supervision more rapidly than the corresponding reduction in the number of samples. With fewer replicate pairs, it becomes more difficult to identify perturbation-consistent biological variation across different technical conditions.

These results clarify both the robustness and the scope of HICON. The method benefits from richer replicate structure, as expected for a replicate-based contrastive approach, but its performance degrades progressively as replicate availability decreases rather than depending on a sharp replicate threshold. At very low replicate counts, the available data are also intrinsically less informative for distinguishing biological variation from technical variation. Therefore, the observed degradation reflects both reduced replicate-based supervision in HICON and the more general difficulty of batch correction under extremely sparse biological replication.

L.2 Generalization Experiments

We further conduct experiments to evaluate whether the model can generalize to unseen technical conditions. Specifically, we construct a challenging cross-source setting on JUMP-CP, where the model is trained on selected imaging sources and evaluated on held-out sources, with no compound overlap between the training and test sets. Both biological and technical branches are trained exclusively on the training sources; samples and metadata from the held-out sources are not used during training. At test time, technical representations for held-out samples are inferred from their features using the trained encoder and prototypes, without access to target-source metadata. This protocol introduces both source-level technical shifts and compound-level distribution shifts across different microscopes. As shown in Table 9, other deep learning methods show limited performance under the cross-source and cross-compound setting. In contrast, HICON achieves the strongest aggregate scores. These results indicate that HICON generalizes better to unseen imaging sources and compounds under cross-microscope shifts.

Table 10: Ablation performance under different training-data fractions on JUMP-CP Scenario 4. Overall is computed as the unweighted average over all seven metrics, i.e., $(3\,\mathrm{Batch} + 4\,\mathrm{Bio})/7$.

Data fraction	Method	Overall	Batch	Bio
30%	Vanilla InfoNCE	0.44	0.37	0.49
	Replicate contrastive	0.55	0.56	0.55
	Hard conditioning (weight)	0.50	0.44	0.54
	Soft conditioning (w/o CLUB)	0.61	0.64	0.58
	Full HICON	**0.63**	**0.68**	**0.60**
50%	Vanilla InfoNCE	0.53	0.51	0.55
	Replicate contrastive	0.62	0.63	0.61
	Hard conditioning (weight)	0.60	0.58	0.62
	Soft conditioning (w/o CLUB)	0.65	0.65	0.65
	Full HICON	**0.67**	**0.68**	**0.67**
75%	Vanilla InfoNCE	0.54	0.52	0.56
	Replicate contrastive	0.64	0.64	0.64
	Hard conditioning (weight)	0.66	0.64	0.67
	Soft conditioning (w/o CLUB)	0.67	0.66	0.67
	Full HICON	**0.68**	**0.68**	**0.68**

M Ablation Study Details

We provide detailed descriptions of the ablation studies reported in Sec. 4.5. All ablation experiments are conducted on JUMP-CP Scenario 4 using 30% of the training data. This reduced-data setting makes the ablations more challenging, as it contains fewer perturbation replicates, weaker cross-batch alignment signals, and more imbalanced batch distributions. It therefore provides a suitable setting for examining whether each component contributes to robust batch correction and biological preservation under limited supervision.

M.1 Ablation Setting and Data Fraction Analysis

We further quantify the rationale and difficulty of the reduced-data setting used for the component-level ablations. JUMP-CP Scenario 4 provides a manageable setting with abundant perturbation replicates, whereas Scenario 5 is more replicate-sparse but substantially more expensive for extensive component-level ablations. We therefore use a 30% subset of Scenario 4 as an intermediate low-replicate regime that makes differences between model variants more apparent while retaining the same experimental structure.

HICON constructs biological positive pairs from samples sharing the same perturbation. For a perturbation with n_p replicates, the number of positive pairs is

$$N_p^+ = \binom{n_p}{2} = \frac{n_p(n_p-1)}{2}.$$

Therefore, reducing the number of replicates leads to a more-than-linear reduction in available positive pairs and biological supervision.

Table 10 compares the replicate statistics and ablation performance across 30%, 50%, and 75% subsets of the Scenario 4 training data.

The 30% subset contains substantially fewer replicates and positive pairs than the larger subsets, making it a more challenging and discriminative setting for component-level ablations. Consistent with the reduced biological supervision, the ablation variants achieve lower Overall performance at 30% than at 50% or 75%. Importantly, the relative ordering of the variants remains unchanged, with Full HICON consistently achieving the strongest Overall performance across all three data fractions. These results indicate that the reduced-data setting increases the difficulty of the ablation analysis without changing the conclusions regarding the contribution of individual components.

M.2 Hierarchical Technical Modeling Ablation.

This ablation examines whether technical variation should be modeled at multiple experimental scales. Cell Painting data contain not only batch-level effects, but also plate-level and spatial effects induced by row and column positions. To evaluate the role of hierarchical technical modeling, we progressively introduce technical objectives from coarse to fine levels. We first use only the batch-level objective, then add the plate-level objective, and finally include the spatial-level objective. All other components are kept unchanged, so the comparison directly reflects the effect of the technical hierarchy. This experiment tests whether a coarse batch-level proxy is sufficient, or whether finer technical levels are needed to capture within-batch and within-plate variation. The results in Table 1 show that adding plate and spatial levels improves the overall performance, especially for fine-grained technical variation. This finding suggests that technical effects in Cell Painting are hierarchical rather than purely batch-level, and supports the design of hierarchy-aware technical proxy learning.

M.3 Biological Branch Ablation.

This ablation evaluates how biological representation learning should use perturbation labels and technical information. The main question is whether contrastive learning should rely on generic instance-level comparisons, use perturbation-replicate positives, or further condition pairwise comparisons on the learned technical proxy. We compare Vanilla InfoNCE, replicate-based supervised contrastive learning, hard technical conditioning, and soft technical conditioning in Table 2(a). Vanilla InfoNCE learns representations without explicitly using perturbation-replicate positives or technical similarity. Replicate-based supervised contrastive learning uses samples with the same perturbation label as positive pairs, which tests whether perturbation replicates provide useful biological supervision. Hard technical conditioning uses technical similarity as a discrete selection or filtering rule for pairwise comparisons. In contrast, soft technical conditioning uses technical similarity as a continuous weighting signal. All variants use the same input features, data split, and evaluation protocol. When technical conditioning is used, the technical branch is kept unchanged, so that the comparison focuses on how technical information is incorporated into the biological branch. The results show that replicate-based positives substantially improve over Vanilla InfoNCE, confirming the importance of perturbation-level biological anchors. However, replicate positives alone are not sufficient, because contrastive learning can still be affected by technical differences between samples. Hard conditioning provides limited improvement, likely because discrete selection may discard useful biological pairs or introduce unstable decisions. Soft technical conditioning achieves the best overall performance, suggesting that technical similarity is more effective when used as a continuous conditioning signal rather than a strict selection rule.

M.4 Technical Branch Ablation.

This ablation analyzes the components used to learn the technical proxy. The goal is to verify that the technical branch does not simply memorize metadata groups, but instead learns a feature-inferred technical representation that can adapt to noisy or incomplete metadata. We remove different components from the full technical branch, as reported in Table 2(b). The compared variants remove confidence weighting, soft targets, or the prototype update mechanism, and additionally evaluate a fixed-prototype variant. The biological branch and the disentanglement loss are kept unchanged, so the comparison focuses on the design of technical proxy learning.

Removing confidence weighting treats all metadata labels as equally consistent. Removing soft targets makes the technical branch rely only on hard metadata labels without using feature-inferred soft assignments. The *w/o proto. update* variant removes the prototype update mechanism, so the technical prototypes are no longer re-estimated from the current learned representations during training. The *Fixed proto.* variant still uses prototypes for assignment and loss computation, but freezes them after initialization so that they remain unchanged throughout training.

These experiments examine whether technical prototypes should be adaptively refined from the learned feature space or treated as fixed technical references. The results show that adaptive prototype learning contributes to the quality of the learned technical proxy. Together with confidence weighting and soft targets, prototype refinement enables the technical branch to use metadata as weak guidance while adapting to sample-level technical heterogeneity.

Table 11: Hyperparameter sensitivity on the 30% subset of JUMP-CP Scenario 4. One hyperparameter is varied at a time while all other settings are fixed. Results are reported as mean ± standard deviation over three independent runs. Batch and Bio denote the averages over three batch-correction and four biological-preservation metrics, respectively, and Overall is the average over all seven metrics.

Component	Hyperparameter	Setting	Batch	Bio	Overall
Loss weights	λ_0	1.0 (0.5×)	0.68 ± 0.01	0.60 ± 0.01	0.63 ± 0.01
		4.0 (2×)	0.66 ± 0.01	0.61 ± 0.01	0.63 ± 0.01
	λ_1	0.5 (0.5×)	0.67 ± 0.01	0.59 ± 0.02	0.62 ± 0.02
		2.0 (2×)	0.64 ± 0.02	0.61 ± 0.01	0.62 ± 0.01
	λ_2	0.1 (0.5×)	0.67 ± 0.01	0.59 ± 0.01	0.62 ± 0.01
		0.4 (2×)	0.68 ± 0.01	0.59 ± 0.00	0.63 ± 0.00
Hierarchical scaling	ρ_1	0.25 (0.5×)	0.68 ± 0.03	0.59 ± 0.03	0.63 ± 0.03
		1.0 (2×)	0.66 ± 0.02	0.58 ± 0.02	0.61 ± 0.02
	ρ_2	0.1 (0.5×)	0.68 ± 0.01	0.59 ± 0.01	0.63 ± 0.01
		0.4 (2×)	0.65 ± 0.07	0.58 ± 0.05	0.61 ± 0.06
Confidence-aware GMM	EM iterations	5	0.66 ± 0.01	0.59 ± 0.02	0.62 ± 0.01
		10	0.68 ± 0.01	0.60 ± 0.02	0.63 ± 0.02
	Momentum	0.80 (Default -0.05)	0.69 ± 0.01	0.59 ± 0.02	0.63 ± 0.01
		0.90 (Default $+0.05$)	0.67 ± 0.01	0.59 ± 0.02	0.62 ± 0.01
Rescaling function	Clipping interval	$[0.2, 1.8]$	0.67 ± 0.01	0.58 ± 0.02	0.62 ± 0.01
		$[0.4, 1.6]$	0.67 ± 0.01	0.60 ± 0.01	0.63 ± 0.00
		$[0.8, 1.2]$	0.68 ± 0.01	0.60 ± 0.02	0.63 ± 0.01
Full configuration	All hyperparameters	Default	0.68 ± 0.01	0.60 ± 0.01	0.63 ± 0.01

Disentanglement Ablation. This ablation evaluates different strategies for reducing residual dependence between the biological representation and the technical proxy. Although technically conditioned contrastive learning reduces technical confounding, technical information may still remain in the biological representation. Therefore, an explicit separation constraint is used to further disentangle biological and technical factors. We compare no disentanglement, GRL, orthogonality regularization, and CLUB in Table 2(c). The variant without disentanglement is trained without an explicit constraint between the biological and technical representations. GRL removes technical information through adversarial training. Orthogonality regularization encourages the biological and technical representations to occupy less overlapping directions in the representation space. CLUB uses a mutual-information upper-bound estimator to suppress broader statistical dependence between biological and technical representations. All other model components are kept unchanged, so the comparison focuses on the effect of the disentanglement strategy. Among the compared strategies, CLUB achieves the best overall performance.

M.5 Hyperparameter Sensitivity

We further evaluate the sensitivity of HICON to its key hyperparameters on the 30% subset of JUMP-CP Scenario 4 used for the component-level ablation studies. Following a one-factor-at-a-time protocol, we vary one hyperparameter while keeping all remaining settings fixed at their default values. We consider the level-wise loss weights $\lambda_0, \lambda_1, \lambda_2$, hierarchical scaling coefficients ρ_1, ρ_2, the confidence-aware GMM configuration and momentum schedule, and the clipping interval used for technical-similarity reweighting. Results are reported as mean ± standard deviation over three independent runs. Here, $0.5\times$ and $2\times$ denote scaling the corresponding default value by factors of 0.5 and 2, respectively, while "Default ± 0.05" denotes shifting the default momentum setting by ± 0.05.

Table 12: Rank-based aggregation on BBBC022 and JUMP-CP Scenario 4. Methods are ranked independently for each individual metric, and the ranks are averaged over the three batch-correction metrics, four biological-preservation metrics, and all seven metrics. Lower is better.

Dataset / setting	Method	Batch mean rank	Bio mean rank	Overall mean rank
BBBC022	HICON	3.75	**2.88**	**3.17**
	fastMNN	**1.50**	5.63	4.25
	Harmony	3.25	5.38	4.67
JUMP-CP Scenario 4	HICON	**1.83**	**1.75**	**1.79**
	Seurat CCA	2.50	4.50	3.64
	Seurat RPCA	4.00	4.63	4.36

N Full Benchmark Results

N.1 Rank-Based Aggregation Analysis

The main benchmark aggregates multiple batch-correction and biological-preservation metrics into summary scores. Since these metrics may have different numerical distributions, we additionally perform a rank-based analysis to examine whether the main conclusions are robust to the choice of score aggregation.

For each benchmark setting, we rank all compared methods independently for each individual metric, where rank 1 indicates the best-performing method. We then compute the mean rank over the three batch-correction metrics, the four biological-preservation metrics, and all seven metrics:

$$R_{\text{batch}} = \frac{1}{3} \sum_{m \in \mathcal{M}_{\text{batch}}} r_m,$$

$$R_{\text{bio}} = \frac{1}{4} \sum_{m \in \mathcal{M}_{\text{bio}}} r_m,$$

and

$$R_{\text{overall}} = \frac{1}{7} \left(\sum_{m \in \mathcal{M}_{\text{batch}}} r_m + \sum_{m \in \mathcal{M}_{\text{bio}}} r_m \right),$$

where r_m denotes the rank of a method under metric m. Lower mean rank indicates better performance.

We report representative results on BBBC022 and JUMP-CP Scenario 4 in Table 12. These two settings cover different datasets and technical conditions and are used to examine whether the conclusions from the score-based benchmark remain consistent under ordinal aggregation.

N.2 Full Results on JUMP CP

We provide the full results for the five JUMP-CP benchmark scenarios in Tables 13–21. These tables include results on both CellProfiler and DeepProfiler features, and report all individual batch-correction metrics, biological preservation metrics, and their aggregated scores. The results complement the summarized comparisons in the main paper and provide a detailed view of method performance across different source and replicate settings.

N.3 Full Results on BBBC

We provide the full results for the BBBC benchmarks in Tables 22–25. These tables include results on BBBC022 and BBBC036, and report all individual batch-correction metrics, biological preservation metrics, aggregated scores, and downstream biological retrieval metrics. The results complement the summarized comparisons in the main paper and provide a detailed view of method performance across different BBBC settings. kBET is not reported for BBBC022 and BBBC036

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Sphering	0.95	0.67	0.42	0.94	0.24	0.62	0.51	0.68	0.58	0.63
	MNN	0.97	0.69	0.45	0.93	0.25	0.66	0.54	0.70	0.59	0.65
	ComBat	0.97	0.67	0.45	0.93	0.23	0.66	0.55	0.70	0.59	0.64
	Baseline	0.97	0.65	0.44	0.93	0.21	0.63	0.52	0.69	0.57	0.63
	Harmony	0.96	0.67	0.45	0.94	0.23	0.68	0.60	0.69	0.61	0.65
	fastMNN	0.98	0.79	0.44	0.92	0.22	0.67	0.63	0.74	0.61	0.67
	Scanorama	0.65	0.88	0.51	0.93	0.20	0.55	0.62	0.68	0.58	0.63
	Seurat RPCA	0.98	0.72	0.49	0.93	0.23	0.67	0.60	0.73	0.61	0.67
	Seurat CCA	0.97	0.72	0.48	0.94	0.25	0.70	0.61	0.72	0.62	0.67
Deep	scBCN	0.76 ± 0.014	0.29 ± 0.019	0.45 ± 0.012	0.90 ± 0.006	0.10 ± 0.013	0.55 ± 0.014	0.66 ± 0.007	0.50 ± 0.002	0.55 ± 0.008	0.53 ± 0.004
	DESC	0.98 ± 0.072	0.73 ± 0.107	0.43 ± 0.013	0.92 ± 0.007	0.23 ± 0.023	0.69 ± 0.054	0.56 ± 0.039	0.71 ± 0.058	0.60 ± 0.031	0.66 ± 0.045
	cpDistiller	0.84 ± 0.003	0.42 ± 0.035	0.44 ± 0.003	0.91 ± 0.004	0.16 ± 0.013	0.55 ± 0.024	0.66 ± 0.001	0.57 ± 0.010	0.57 ± 0.010	0.57 ± 0.010
	scVI	0.96 ± 0.001	0.65 ± 0.028	0.45 ± 0.006	0.93 ± 0.005	0.27 ± 0.016	0.69 ± 0.004	0.57 ± 0.004	0.69 ± 0.010	0.61 ± 0.004	0.65 ± 0.003
Ours	**HICON**	$\mathbf{0.99 \pm 0.001}$	$\mathbf{0.84 \pm 0.028}$	$\mathbf{0.50 \pm 0.013}$	$\mathbf{0.94 \pm 0.003}$	$\mathbf{0.31 \pm 0.024}$	$\mathbf{0.74 \pm 0.008}$	$\mathbf{0.95 \pm 0.003}$	$\mathbf{0.78 \pm 0.013}$	$\mathbf{0.73 \pm 0.007}$	$\mathbf{0.76 \pm 0.008}$

Table 13: Benchmark results on JUMP-CP Scenario 1 using CellProfiler features.

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Sphering	0.85	0.13	0.22	0.93	0.14	0.48	0.39	0.40	0.48	0.44
	MNN	0.89	0.24	0.36	0.92	0.15	0.52	0.42	0.50	0.50	0.50
	ComBat	0.90	0.24	0.42	0.92	0.18	0.54	0.42	0.52	0.52	0.52
	Baseline	0.90	0.23	0.36	0.92	0.13	0.52	0.41	0.50	0.49	0.50
	Harmony	0.90	0.54	0.55	0.91	0.16	0.52	0.46	0.66	0.51	0.59
	fastMNN	0.83	0.41	0.56	0.90	0.16	0.49	0.45	0.60	0.50	0.55
	Scanorama	0.78	0.34	0.66	0.91	0.12	0.43	0.68	0.59	0.54	0.56
	Seurat RPCA	0.89	0.42	0.55	0.91	0.18	0.54	0.47	0.62	0.53	0.57
	Seurat CCA	0.87	0.48	0.61	0.91	0.18	0.53	0.47	0.65	0.52	0.59
Deep	scBCN	0.64 ± 0.020	0.12 ± 0.028	0.54 ± 0.014	0.88 ± 0.007	0.09 ± 0.012	0.38 ± 0.007	0.61 ± 0.002	0.43 ± 0.015	0.49 ± 0.005	0.46 ± 0.005
	DESC	0.83 ± 0.031	0.29 ± 0.223	0.40 ± 0.075	0.91 ± 0.008	0.13 ± 0.013	0.46 ± 0.030	0.41 ± 0.012	0.51 ± 0.076	0.48 ± 0.013	0.49 ± 0.045
	cpDistiller	0.66 ± 0.010	0.09 ± 0.023	0.50 ± 0.025	0.88 ± 0.005	0.06 ± 0.004	0.32 ± 0.021	0.58 ± 0.007	0.42 ± 0.005	0.46 ± 0.005	0.44 ± 0.000
	scVI	0.83 ± 0.005	0.36 ± 0.022	0.52 ± 0.019	0.93 ± 0.007	0.20 ± 0.012	0.55 ± 0.017	0.46 ± 0.003	0.57 ± 0.004	0.54 ± 0.008	0.55 ± 0.003
Ours	**HICON**	$\mathbf{0.96 \pm 0.001}$	$\mathbf{0.72 \pm 0.007}$	$\mathbf{0.57 \pm 0.019}$	$\mathbf{0.97 \pm 0.002}$	$\mathbf{0.40 \pm 0.006}$	$\mathbf{0.75 \pm 0.004}$	$\mathbf{0.95 \pm 0.001}$	$\mathbf{0.75 \pm 0.007}$	$\mathbf{0.77 \pm 0.001}$	$\mathbf{0.76 \pm 0.004}$

Table 14: Benchmark results on JUMP-CP Scenario 2 using CellProfiler features.

because no perturbations satisfy the minimum sample-size requirements of the current scIB implementation. kBET is computed within biological groups and requires sufficient samples spanning multiple batches, whereas the median number of replicates per perturbation is only 4 for BBBC022 and 8 for BBBC036.

N.4 Full Results on CPG0036

We provide the full results for the CPG0036 benchmark in Tables 26–30. These tables include results across five target-based evaluation settings, covering different compound collections and cell lines. They report all individual batch-correction metrics, biological preservation metrics, and their aggregated scores. The results complement the summarized comparisons in the main paper and provide a detailed view of method performance across different CPG0036 settings.

N.5 Full Results on RxRx1

We provide the full results for the RxRx1 benchmark in Tables 31–34. These tables include results on OpenPhenom features across four cell lines: HepG2, HUVEC, RPE, and U2OS. They report all individual batch-correction metrics, biological preservation metrics, and their aggregated scores. The results complement the summarized comparisons in the main paper and provide a detailed view of method performance across different RxRx1 cell-line settings.

The rank-based analysis leads to conclusions consistent with the score-based aggregation. On JUMP-CP Scenario 4, HICON achieves the lowest mean rank for Batch, Bio, and Overall among the reported methods. On BBBC022, fastMNN obtains a lower Batch mean rank, whereas HICON achieves substantially stronger Bio and Overall mean ranks.

This difference illustrates an important property of rank-based aggregation: it preserves the ordering of methods within each metric but discards the magnitude of performance differences. For example, on BBBC022, HICON achieves a Graph Connectivity score of 0.88, compared with 0.74 for the strongest competing baseline, while its batch-LISI score is only slightly below the best baseline. These differences contribute equally to ordinal ranks regardless of their magnitude. Therefore, we

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Sphering	0.54	0.03	0.00	1.00	0.02	0.29	0.03	0.19	0.34	0.27
	Baseline	0.54	0.05	0.03	1.00	0.02	0.29	0.03	0.21	0.34	0.28
	MNN	0.55	0.03	0.04	1.00	0.03	0.30	0.04	0.21	0.34	0.28
	ComBat	0.55	0.03	0.07	1.00	0.03	0.31	0.04	0.22	0.35	0.29
	Harmony	0.55	0.08	0.45	1.00	0.03	0.33	0.05	0.36	0.35	0.36
	Seurat RPCA	0.55	0.16	0.42	1.00	0.06	0.33	0.05	0.38	0.36	0.37
	Scanorama	0.55	0.30	0.41	1.00	0.03	0.31	0.07	0.45	0.35	0.38
	fastMNN	0.54	0.09	0.62	1.00	0.06	0.28	0.12	0.42	0.37	0.39
	Seurat CCA	0.56	0.18	0.59	1.00	0.06	0.31	0.06	0.44	0.36	0.40
Deep	DESC	0.63	0.00	0.07	1.00	0.05	0.39	0.02	0.23	0.37	0.30
	cpDistiller	0.54	0.02	0.36	1.00	0.02	0.28	0.10	0.31	0.35	0.33
	scBCN	0.54	0.05	0.36	1.00	0.01	0.27	0.09	0.32	0.34	0.33
	scVI	0.55	0.07	0.43	1.00	0.03	0.33	0.05	0.35	0.35	0.35
Ours	**HICON**	0.54	0.19	0.62	1.00	0.07	0.35	0.07	**0.45**	**0.37**	**0.41**

Table 15: Benchmark results on JUMP-CP Scenario 3 using CellProfiler features.

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	kBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Sphering	0.88	0.13	0.18	0.93	0.08	0.39	0.40	0.40	0.45	0.42
	MNN	0.91	0.17	0.26	0.92	0.12	0.44	0.76	0.45	0.56	0.50
	ComBat	0.92	0.13	0.26	0.92	0.12	0.45	0.75	0.44	0.56	0.50
	Baseline	0.91	0.22	0.26	0.92	0.11	0.44	0.75	0.46	0.56	0.51
	Harmony	0.92	0.42	0.45	0.92	0.13	0.47	0.78	0.60	0.58	0.59
	fastMNN	0.88	0.54	0.48	0.91	0.12	0.42	0.78	0.63	0.56	0.60
	Scanorama	0.72	0.56	0.52	0.81	0.92	0.09	0.38	0.60	0.55	0.58
	Seurat RPCA	0.93	0.49	0.46	0.92	0.15	0.49	0.78	0.63	0.59	0.61
	Seurat CCA	0.92	0.63	0.52	0.92	0.14	0.48	0.79	0.69	0.58	0.64
Deep	scBCN	0.70 ± 0.014	0.25 ± 0.016	0.45 ± 0.007	0.89 ± 0.004	0.07 ± 0.005	0.35 ± 0.010	0.72 ± 0.007	0.47 ± 0.005	0.51 ± 0.003	0.49 ± 0.001
	DESC	0.26 ± 0.075	0.21 ± 0.050	0.40 ± 0.028	0.67 ± 0.006	0.89 ± 0.012	0.12 ± 0.019	0.43 ± 0.003	0.29 ± 0.022	0.53 ± 0.010	0.41 ± 0.008
	cpDistiller	0.76 ± 0.034	0.18 ± 0.034	0.42 ± 0.014	0.76 ± 0.000	0.89 ± 0.006	0.06 ± 0.024	0.30 ± 0.017	0.45 ± 0.025	0.50 ± 0.010	0.48 ± 0.017
	scVI	0.89 ± 0.013	0.35 ± 0.038	0.45 ± 0.009	0.92 ± 0.003	0.16 ± 0.007	0.47 ± 0.006	0.78 ± 0.002	0.56 ± 0.017	0.58 ± 0.001	0.57 ± 0.009
Ours	**HICON**	$\mathbf{0.95 \pm 0.001}$	$\mathbf{0.64 \pm 0.004}$	0.48 ± 0.017	$\mathbf{0.95 \pm 0.002}$	$\mathbf{0.23 \pm 0.023}$	$\mathbf{0.63 \pm 0.014}$	$\mathbf{0.89 \pm 0.001}$	$\mathbf{0.69 \pm 0.007}$	$\mathbf{0.68 \pm 0.010}$	$\mathbf{0.68 \pm 0.002}$

Table 16: Benchmark results on JUMP-CP Scenario 4 using CellProfiler features.

use the rank-based analysis as a complementary robustness check rather than as a replacement for the score-based aggregation.

Overall, the consistency between score-based and rank-based evaluations indicates that the benchmark conclusions are not driven by a small subset of metrics or by differences in metric scale.

O Limitations and Broader Impact

HICON relies on biological anchors such as perturbation replicates and controls to guide contrastive learning, and therefore benefits from sufficient replicate structure. Our replicate-sensitivity analysis (Appendix L.1) shows that performance degrades gradually as the number of available replicates decreases. Extremely sparse replicate regimes remain challenging because they provide limited information for separating perturbation-relevant biological variation from technical variation. Thus, HICON does not eliminate the fundamental dependence of Cell Painting analysis on adequate biological replication. HICON also assumes that biological and technical variation can be sufficiently separated. This assumption may be challenged when perturbations are strongly correlated with technical factors, while a loose CLUB bound may leave residual technical information in the biological representation.

By improving batch correction while preserving perturbation-relevant biological structure, HICON may support more reproducible and reliable Cell Painting analysis for downstream applications such as phenotypic activity detection, mechanism-of-action retrieval, compound similarity analysis, and drug discovery. More broadly, the method encourages a shift from treating metadata-defined batches as fixed correction targets toward modeling technical variation as latent, hierarchical, and sample-specific. However, corrected representations should still be interpreted carefully, since residual technical artifacts or overcorrection may affect downstream biological conclusions. Therefore, HICON

Table 17: Benchmark results on JUMP-CP Scenario 5 using CellProfiler features.

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Sphering	0.34	0.01	0.00	1.00	0.00	0.20	0.03	0.12	0.31	0.22
	Baseline	0.34	0.03	0.03	1.00	0.01	0.24	0.03	0.13	0.32	0.23
	MNN	0.35	0.02	0.06	1.00	0.01	0.24	0.04	0.14	0.32	0.23
	ComBat	0.35	0.03	0.09	1.00	0.01	0.25	0.04	0.16	0.33	0.25
	Harmony	0.36	0.05	0.40	1.00	0.02	0.27	0.05	0.27	0.34	0.30
	Seurat RPCA	0.36	0.15	0.41	1.00	0.03	0.27	0.06	0.31	0.34	0.33
	fastMNN	0.36	0.10	0.49	1.00	0.06	0.24	0.07	0.32	0.34	0.33
	Seurat CCA	0.37	0.21	0.55	1.00	0.06	0.26	0.07	0.38	0.35	0.36
	Scanorama	0.34	0.36	0.44	1.00	0.02	0.27	0.12	0.38	0.35	0.36
Deep	cpDistiller	0.34	0.03	0.35	1.00	0.01	0.23	0.11	0.24	0.34	0.29
	scBCN	0.34	0.01	0.49	1.00	0.01	0.21	0.11	0.28	0.33	0.31
	scVI	0.35	0.06	0.36	1.00	0.04	0.31	0.06	0.26	0.35	0.31
	DESC	0.33	0.09	0.43	1.00	0.03	0.28	0.07	0.28	0.35	0.32
Ours	**HICON**	0.39	0.16	0.61	1.00	0.05	0.31	0.13	**0.39**	**0.37**	**0.38**

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Scanorama	0.57	0.95	0.49	0.90	0.13	0.44	0.53	0.67	0.50	0.59
	Sphering	0.95	0.67	0.42	0.94	0.24	0.62	0.51	0.68	0.58	0.63
	Baseline	0.97	0.65	0.44	0.93	0.21	0.63	0.75	0.69	0.63	0.66
	ComBat	0.97	0.67	0.45	0.93	0.23	0.66	0.77	0.70	0.65	0.67
	MNN	0.97	0.69	0.45	0.93	0.25	0.66	0.77	0.70	0.65	0.67
	Harmony	0.96	0.67	0.45	0.94	0.23	0.68	0.80	0.69	0.66	0.68
	Seurat CCA	0.97	0.72	0.48	0.94	0.25	0.70	0.81	0.72	0.68	0.70
	fastMNN	0.98	0.79	0.44	0.92	0.21	0.67	0.82	0.74	0.66	0.70
	Seurat RPCA	0.98	0.72	0.49	0.93	0.23	0.67	0.81	0.73	0.66	0.70
Deep	scBCN	0.51 ± 0.004	0.26 ± 0.010	0.48 ± 0.018	0.88 ± 0.003	0.08 ± 0.003	0.36 ± 0.013	0.51 ± 0.003	0.42 ± 0.006	0.46 ± 0.004	0.44 ± 0.005
	cpDistiller	0.62 ± 0.010	0.34 ± 0.010	0.47 ± 0.014	0.89 ± 0.005	0.10 ± 0.011	0.37 ± 0.019	0.52 ± 0.003	0.48 ± 0.004	0.47 ± 0.010	0.48 ± 0.006
	scVI	0.96 ± 0.007	0.62 ± 0.019	0.46 ± 0.013	0.93 ± 0.009	0.25 ± 0.017	0.68 ± 0.015	0.79 ± 0.002	0.68 ± 0.010	0.66 ± 0.009	0.67 ± 0.002
	DESC	0.98 ± 0.010	0.73 ± 0.012	0.49 ± 0.001	0.93 ± 0.004	0.23 ± 0.011	0.71 ± 0.017	0.79 ± 0.005	0.73 ± 0.007	0.67 ± 0.008	0.70 ± 0.007
Ours	**HICON**	0.98 ± 0.003	0.77 ± 0.041	0.46 ± 0.004	0.96 ± 0.001	0.33 ± 0.019	0.71 ± 0.004	0.87 ± 0.002	$\mathbf{0.74 \pm 0.015}$	$\mathbf{0.72 \pm 0.006}$	$\mathbf{0.73 \pm 0.006}$

Table 18: Benchmark results on JUMP-CP Scenario 1 using DeepProfiler features.

should be used together with quality control, biological validation, and transparent reporting of metadata, replicate structure, and evaluation protocols.

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Sphering	0.68	0.12	0.26	0.92	0.06	0.43	0.33	0.35	0.44	0.40
	Scanorama	0.54	0.22	0.56	0.89	0.14	0.43	0.62	0.44	0.52	0.48
	Baseline	0.66	0.46	0.38	0.89	0.09	0.42	0.60	0.50	0.50	0.50
	ComBat	0.67	0.46	0.36	0.89	0.11	0.43	0.61	0.50	0.51	0.51
	MNN	0.69	0.48	0.38	0.89	0.11	0.43	0.61	0.52	0.51	0.52
	Seurat RPCA	0.72	0.61	0.45	0.90	0.10	0.39	0.61	0.59	0.50	0.55
	Seurat CCA	0.71	0.66	0.45	0.90	0.11	0.40	0.61	0.61	0.51	0.56
	fastMNN	0.63	0.77	0.52	0.89	0.14	0.43	0.63	0.64	0.52	0.58
	Harmony	0.69	0.79	0.56	0.88	0.13	0.43	0.63	0.68	0.52	0.60
Deep	cpDistiller	0.53 ± 0.017	0.09 ± 0.013	0.47 ± 0.055	0.87 ± 0.004	0.05 ± 0.007	0.31 ± 0.011	0.52 ± 0.013	0.36 ± 0.021	0.44 ± 0.006	0.40 ± 0.012
	scBCN	0.49 ± 0.016	0.60 ± 0.010	0.49 ± 0.057	0.87 ± 0.003	0.06 ± 0.009	0.41 ± 0.008	0.60 ± 0.004	0.53 ± 0.022	0.49 ± 0.002	0.51 ± 0.012
	scVI	0.63 ± 0.007	0.70 ± 0.007	0.55 ± 0.024	0.89 ± 0.003	0.16 ± 0.007	0.45 ± 0.011	0.63 ± 0.001	0.63 ± 0.006	0.53 ± 0.004	0.58 ± 0.005
Ours	**HICON**	0.93 ± 0.003	0.77 ± 0.007	0.59 ± 0.023	0.95 ± 0.003	0.32 ± 0.002	0.70 ± 0.003	0.83 ± 0.002	**0.76 ± 0.009**	**0.70 ± 0.000**	**0.73 ± 0.005**

Table 19: Benchmark results on JUMP-CP Scenario 2 using DeepProfiler features.

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Sphering	0.54	0.02	0.00	1.00	0.01	0.24	0.02	0.19	0.32	0.26
	Baseline	0.54	0.01	0.06	1.00	0.02	0.26	0.08	0.20	0.34	0.27
	MNN	0.54	0.01	0.09	1.00	0.01	0.24	0.09	0.21	0.34	0.28
	ComBat	0.54	0.00	0.11	1.00	0.01	0.25	0.09	0.22	0.34	0.28
	Harmony	0.55	0.06	0.58	1.00	0.06	0.30	0.11	0.40	0.37	0.39
	fastMNN	0.54	0.09	0.62	1.00	0.06	0.28	0.12	0.42	0.37	0.39
Deep	scBCN	0.54	0.04	0.19	1.00	0.02	0.24	0.08	0.26	0.35	0.31
	scVI	0.54	0.07	0.61	1.00	0.04	0.29	0.12	0.41	0.36	0.38
	cpDistiller	0.54	0.01	0.52	1.00	0.01	0.25	0.10	0.36	0.34	0.35
Ours	**HICON**	0.54	0.25	0.70	1.00	0.08	0.36	0.17	0.50	0.40	0.45

Table 20: Benchmark results on JUMP-CP Scenario 3 using DeepProfiler features.

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Sphering	0.63	0.04	0.18	0.90	0.05	0.38	0.33	0.28	0.42	0.35
	ComBat	0.57	0.04	0.30	0.87	0.07	0.36	0.68	0.30	0.50	0.40
	Baseline	0.59	0.06	0.30	0.89	0.06	0.33	0.68	0.32	0.49	0.40
	MNN	0.59	0.07	0.28	0.88	0.07	0.36	0.69	0.31	0.50	0.41
	Scanorama	0.49	0.73	0.55	0.90	0.08	0.32	0.70	0.59	0.50	0.54
	fastMNN	0.62	0.27	0.48	0.88	0.09	0.33	0.72	0.46	0.51	0.48
	Harmony	0.65	0.25	0.52	0.88	0.10	0.39	0.71	0.47	0.52	0.50
	Seurat RPCA	0.89	0.39	0.24	0.90	0.10	0.43	0.73	0.51	0.54	0.52
	Seurat CCA	0.88	0.43	0.31	0.90	0.10	0.42	0.73	0.54	0.54	0.54
Deep	scBCN	0.39 ± 0.037	0.05 ± 0.004	0.40 ± 0.040	0.87 ± 0.003	0.04 ± 0.002	0.30 ± 0.004	0.66 ± 0.015	0.28 ± 0.027	0.47 ± 0.005	0.37 ± 0.016
	cpDistiller	0.49 ± 0.013	0.07 ± 0.028	0.42 ± 0.018	0.86 ± 0.002	0.03 ± 0.002	0.21 ± 0.005	0.61 ± 0.004	0.33 ± 0.019	0.43 ± 0.002	0.38 ± 0.010
	DESC	0.46 ± 0.075	0.47 ± 0.050	0.40 ± 0.028	0.89 ± 0.006	0.08 ± 0.012	0.30 ± 0.019	0.68 ± 0.003	0.44 ± 0.022	0.49 ± 0.010	0.47 ± 0.008
	scVI	0.61 ± 0.006	0.21 ± 0.005	0.49 ± 0.011	0.88 ± 0.002	0.11 ± 0.008	0.42 ± 0.007	0.71 ± 0.001	0.44 ± 0.004	0.53 ± 0.002	0.48 ± 0.003
Ours	**HICON**	0.88 ± 0.003	0.72 ± 0.011	0.49 ± 0.006	0.94 ± 0.002	0.19 ± 0.011	0.61 ± 0.001	0.83 ± 0.001	**0.70 ± 0.004**	**0.64 ± 0.002**	**0.67 ± 0.002**

Table 21: Benchmark results on JUMP-CP Scenario 4 using DeepProfiler features.

Type	Method	Batch correction		Bio metrics				Aggregate score		
		Graph connectivity	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	fastMNN	0.74	0.41	0.96	0.02	0.45	0.54	0.58	0.49	0.53
	Harmony	0.72	0.40	0.96	0.02	0.46	0.51	0.56	0.49	0.52
	ComBat	0.72	0.39	0.96	0.02	0.44	0.45	0.56	0.47	0.51
	Baseline	0.72	0.40	0.96	0.02	0.46	0.41	0.56	0.46	0.51
	MNN	0.71	0.38	0.96	0.02	0.45	0.42	0.55	0.46	0.50
	Sphering	0.71	0.36	0.96	0.02	0.42	0.20	0.54	0.40	0.47
	Seurat RPCA	0.65	0.30	0.97	0.02	0.41	0.32	0.48	0.43	0.46
	Seurat CCA	0.63	0.32	0.96	0.02	0.41	0.31	0.48	0.43	0.45
	Scanorama	0.51	0.39	0.96	0.01	0.41	0.31	0.45	0.42	0.44
Deep	scVI	0.71 ± 0.006	0.39 ± 0.005	0.96 ± 0.001	0.02 ± 0.001	0.44 ± 0.007	0.51 ± 0.001	0.55 ± 0.004	0.48 ± 0.002	0.52 ± 0.002
	cpDistiller	0.61 ± 0.019	0.39 ± 0.007	0.96 ± 0.000	0.01 ± 0.001	0.37 ± 0.010	0.39 ± 0.004	0.50 ± 0.010	0.43 ± 0.003	0.47 ± 0.005
	DESC	0.55 ± 0.021	0.39 ± 0.011	0.96 ± 0.002	0.02 ± 0.001	0.42 ± 0.006	0.36 ± 0.009	0.47 ± 0.012	0.44 ± 0.003	0.46 ± 0.006
	scBCN	0.57 ± 0.006	0.30 ± 0.011	0.97 ± 0.001	0.02 ± 0.000	0.45 ± 0.004	0.33 ± 0.003	0.44 ± 0.006	0.44 ± 0.001	0.44 ± 0.003
Ours	**HICON**	**0.88 ± 0.003**	**0.39 ± 0.008**	**0.96 ± 0.001**	**0.03 ± 0.005**	**0.49 ± 0.016**	**0.66 ± 0.002**	**0.64 ± 0.004**	**0.54 ± 0.004**	**0.59 ± 0.003**

Table 22: Performance comparison on BBBC022.

Type	Method	Batch correction		Bio metrics				Aggregate score		
		Graph connectivity	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Sphering	0.46	0.15	0.97	0.01	0.26	0.15	0.31	0.35	0.33
	MNN	0.37	0.16	0.97	0.00	0.26	0.28	0.27	0.38	0.32
	ComBat	0.37	0.16	0.97	0.00	0.26	0.28	0.27	0.38	0.32
	Baseline	0.38	0.16	0.97	0.00	0.25	0.28	0.27	0.38	0.32
	Harmony	0.38	0.18	0.97	0.01	0.28	0.32	0.28	0.40	0.34
	fastMNN	0.42	0.18	0.97	0.01	0.28	0.37	0.30	0.41	0.35
	Scanorama	0.25	0.14	0.98	0.00	0.30	0.20	0.20	0.37	0.28
	Seurat RPCA	0.32	0.15	0.97	0.00	0.24	0.30	0.24	0.38	0.31
	Seurat CCA	0.35	0.17	0.96	0.01	0.25	0.35	0.26	0.39	0.33
Deep	scBCN	0.29 ± 0.002	0.17 ± 0.000	0.97 ± 0.001	0.01 ± 0.000	0.28 ± 0.002	0.21 ± 0.001	0.23 ± 0.001	0.37 ± 0.000	0.30 ± 0.001
	DESC	0.24 ± 0.008	0.15 ± 0.002	0.97 ± 0.001	0.01 ± 0.000	0.29 ± 0.003	0.16 ± 0.005	0.20 ± 0.004	0.36 ± 0.002	0.28 ± 0.002
	cpDistiller	0.33 ± 0.003	0.16 ± 0.001	0.97 ± 0.000	0.00 ± 0.001	0.24 ± 0.002	0.25 ± 0.002	0.25 ± 0.002	0.37 ± 0.001	0.31 ± 0.001
	scVI	0.37 ± 0.004	0.18 ± 0.001	0.97 ± 0.000	0.01 ± 0.000	0.29 ± 0.003	0.35 ± 0.004	0.28 ± 0.002	0.41 ± 0.001	0.34 ± 0.001
Ours	**HICON**	$\mathbf{0.59 \pm 0.004}$	$\mathbf{0.18 \pm 0.005}$	$\mathbf{0.98 \pm 0.000}$	$\mathbf{0.02 \pm 0.003}$	$\mathbf{0.40 \pm 0.013}$	$\mathbf{0.50 \pm 0.002}$	$\mathbf{0.39 \pm 0.003}$	$\mathbf{0.48 \pm 0.003}$	$\mathbf{0.43 \pm 0.002}$

Table 23: Performance comparison on BBBC036.

Type	Method	mAP	P@1	P@5	P@10	R@1	R@5	R@10	EOR@90
Non-Deep	Scanorama	4.4	4.1	2.6	2.3	2.0	6.8	12.2	28.3
	Baseline	5.6	6.1	3.3	2.7	3.2	8.8	13.9	30.7
	Harmony	5.6	6.1	3.3	2.6	3.2	8.6	13.6	29.6
	ComBat	5.6	6.3	3.2	2.7	3.3	8.8	13.8	29.6
	MNN	5.7	6.2	3.4	2.8	3.3	9.1	14.1	29.8
	fastMNN	5.8	6.3	3.5	2.8	3.1	8.9	14.3	29.9
	Seurat RPCA	6.0	6.7	3.5	2.8	3.4	8.7	14.2	32.0
	Seurat CCA	6.6	6.9	3.9	3.0	3.0	9.3	14.6	31.0
	Sphering	7.0	**7.0**	3.9	3.2	4.2	10.9	15.8	33.8
Deep	DESC	3.4	2.4	2.0	2.0	1.4	6.4	11.3	27.1
	scBCN	5.2	5.7	3.1	2.4	2.7	8.1	12.8	28.4
	cpDistiller	6.0	6.5	3.7	3.0	3.4	9.3	14.6	26.9
	scVI	6.8	7.2	4.1	3.3	3.2	9.7	15.3	31.1
Ours	**HICON**	**7.2**	**7.0**	**4.4**	**3.4**	**3.6**	**10.7**	**16.2**	**34.8**

Table 24: Mechanism-of-action retrieval performance on BBBC022 using DeepProfiler features.

Type	Method	mAP	P@1	P@5	P@10	R@1	R@5	R@10	EOR@90
Non-Deep	Scanorama	5.0	4.8	3.3	3.0	2.1	7.6	13.0	10.0
	Seurat RPCA	6.0	6.4	4.1	3.6	2.8	8.8	14.8	10.4
	Baseline	6.4	6.7	4.2	3.6	3.2	9.3	15.2	10.6
	ComBat	6.4	6.7	4.1	3.6	3.2	9.2	15.2	10.6
	Harmony	6.4	6.6	4.2	3.6	3.2	9.3	15.2	10.6
	MNN	6.5	6.9	4.3	3.7	3.2	9.6	15.4	10.6
	Seurat CCA	6.9	7.3	4.3	3.7	3.4	9.2	15.5	10.8
	fastMNN	7.7	8.3	4.9	4.1	3.9	10.7	16.8	11.6
	Sphering	8.1	9.0	4.9	4.2	4.0	11.2	16.9	11.4
Deep	DESC	4.3	3.6	3.1	2.8	1.5	6.6	11.3	9.6
	scBCN	5.1	4.4	3.1	2.9	2.1	7.0	12.3	10.0
	scVI	6.4	6.2	4.3	3.6	2.9	9.5	15.3	10.6
	cpDistiller	6.4	6.5	4.1	3.6	3.1	9.1	15.0	10.5
Ours	**HICON**	**8.3**	**9.3**	**5.3**	**4.2**	**4.2**	**11.6**	**17.3**	**12.5**

Table 25: Mechanism-of-action retrieval performance on BBBC036 using DeepProfiler features.

Type	Method	Batch correction		Bio metrics				Aggregate score		
		Graph connectivity	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Harmony	0.71	0.77	0.97	0.01	0.43	0.49	0.74	0.48	0.61
	fastMNN	0.70	0.74	0.97	0.02	0.44	0.53	0.72	0.49	0.61
	ComBat	0.69	0.73	0.97	0.01	0.41	0.46	0.71	0.46	0.59
	Seurat CCA	0.70	0.63	0.98	0.01	0.40	0.55	0.66	0.48	0.57
	MNN	0.67	0.67	0.97	0.01	0.42	0.44	0.67	0.46	0.57
	Seurat RPCA	0.66	0.65	0.98	0.01	0.41	0.51	0.65	0.48	0.57
	Scanorama	0.58	0.70	0.98	0.01	0.44	0.39	0.64	0.46	0.55
	Baseline	0.64	0.59	0.97	0.01	0.36	0.40	0.62	0.44	0.53
	Sphering	0.64	0.48	0.97	0.01	0.37	0.22	0.56	0.39	0.47
Deep	scVI	0.69 ± 0.006	0.73 ± 0.007	0.97 ± 0.001	0.02 ± 0.001	0.47 ± 0.007	0.52 ± 0.004	0.71 ± 0.005	0.49 ± 0.002	0.60 ± 0.003
	cpDistiller	0.62 ± 0.010	0.66 ± 0.010	0.97 ± 0.001	0.01 ± 0.002	0.39 ± 0.011	0.41 ± 0.006	0.64 ± 0.007	0.44 ± 0.003	0.54 ± 0.004
	DESC	0.49 ± 0.014	0.71 ± 0.013	0.97 ± 0.001	0.01 ± 0.002	0.44 ± 0.010	0.32 ± 0.007	0.60 ± 0.010	0.44 ± 0.003	0.52 ± 0.005
	scBCN	0.54 ± 0.012	0.63 ± 0.011	0.97 ± 0.001	0.01 ± 0.002	0.44 ± 0.009	0.34 ± 0.006	0.59 ± 0.008	0.44 ± 0.003	0.51 ± 0.004
Ours	**HICON**	$\mathbf{0.90 \pm 0.004}$	$\mathbf{0.80 \pm 0.006}$	$\mathbf{0.97 \pm 0.001}$	$\mathbf{0.04 \pm 0.004}$	$\mathbf{0.56 \pm 0.009}$	$\mathbf{0.70 \pm 0.003}$	$\mathbf{0.85 \pm 0.004}$	$\mathbf{0.57 \pm 0.003}$	$\mathbf{0.71 \pm 0.002}$

Table 26: Performance comparison on CPG0036 FMP_HepG2 .

Type	Method	Batch correction		Bio metrics				Aggregate score		
		Graph connectivity	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Seurat CCA	0.69	0.74	0.98	0.01	0.43	0.48	0.72	0.48	0.60
	Harmony	0.69	0.75	0.97	0.01	0.42	0.45	0.72	0.46	0.59
	fastMNN	0.68	0.73	0.97	0.01	0.41	0.49	0.71	0.47	0.59
	ComBat	0.68	0.74	0.97	0.01	0.43	0.42	0.71	0.46	0.58
	Seurat RPCA	0.66	0.73	0.98	0.01	0.42	0.45	0.70	0.47	0.58
	MNN	0.67	0.68	0.97	0.01	0.41	0.40	0.68	0.45	0.56
	Scanorama	0.58	0.66	0.98	0.02	0.45	0.38	0.62	0.46	0.54
	Baseline	0.61	0.50	0.97	0.01	0.36	0.35	0.55	0.42	0.49
	Sphering	0.58	0.46	0.97	0.01	0.35	0.18	0.52	0.38	0.45
Deep	scVI	0.64 ± 0.005	0.73 ± 0.006	0.97 ± 0.001	0.01 ± 0.001	0.43 ± 0.006	0.45 ± 0.004	0.68 ± 0.004	0.47 ± 0.002	0.57 ± 0.002
	DESC	0.53 ± 0.013	0.72 ± 0.012	0.97 ± 0.001	0.01 ± 0.002	0.40 ± 0.009	0.35 ± 0.006	0.63 ± 0.009	0.43 ± 0.003	0.53 ± 0.005
	cpDistiller	0.53 ± 0.009	0.56 ± 0.009	0.97 ± 0.001	0.01 ± 0.002	0.35 ± 0.010	0.29 ± 0.005	0.54 ± 0.006	0.40 ± 0.003	0.47 ± 0.003
	scBCN	0.51 ± 0.011	0.43 ± 0.010	0.97 ± 0.001	0.01 ± 0.002	0.44 ± 0.008	0.31 ± 0.005	0.47 ± 0.007	0.43 ± 0.002	0.45 ± 0.004
Ours	**HICON**	$\mathbf{0.95 \pm 0.004}$	$\mathbf{0.79 \pm 0.005}$	$\mathbf{0.97 \pm 0.001}$	$\mathbf{0.05 \pm 0.004}$	$\mathbf{0.61 \pm 0.008}$	$\mathbf{0.69 \pm 0.003}$	$\mathbf{0.87 \pm 0.003}$	$\mathbf{0.58 \pm 0.002}$	$\mathbf{0.73 \pm 0.002}$

Table 27: Performance comparison on CPG0036 FMP_U2OS .

Type	Method	Batch correction		Bio metrics				Aggregate score		
		Graph connectivity	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	fastMNN	0.77	0.79	0.97	0.01	0.43	0.57	0.78	0.50	0.64
	Harmony	0.77	0.80	0.97	0.02	0.43	0.54	0.79	0.49	0.64
	ComBat	0.77	0.79	0.97	0.02	0.43	0.52	0.78	0.48	0.63
	MNN	0.77	0.79	0.97	0.02	0.42	0.52	0.78	0.48	0.63
	Baseline	0.76	0.77	0.97	0.02	0.43	0.51	0.77	0.48	0.62
	Sphering	0.79	0.78	0.97	0.02	0.47	0.34	0.79	0.45	0.62
	Seurat CCA	0.76	0.74	0.98	0.01	0.39	0.56	0.75	0.49	0.62
	Seurat RPCA	0.74	0.72	0.98	0.01	0.39	0.55	0.73	0.48	0.61
	Scanorama	0.65	0.74	0.98	0.02	0.44	0.47	0.69	0.48	0.58
Deep	scVI	0.76 ± 0.005	0.79 ± 0.006	0.97 ± 0.001	0.02 ± 0.001	0.48 ± 0.006	0.56 ± 0.003	0.78 ± 0.004	0.51 ± 0.002	0.64 ± 0.002
	cpDistiller	0.69 ± 0.008	0.80 ± 0.008	0.97 ± 0.001	0.01 ± 0.002	0.42 ± 0.009	0.48 ± 0.005	0.74 ± 0.006	0.47 ± 0.003	0.61 ± 0.003
	scBCN	0.64 ± 0.010	0.75 ± 0.009	0.97 ± 0.001	0.01 ± 0.002	0.45 ± 0.007	0.46 ± 0.005	0.69 ± 0.007	0.47 ± 0.002	0.58 ± 0.003
	DESC	0.56 ± 0.011	0.82 ± 0.010	0.97 ± 0.001	0.01 ± 0.002	0.43 ± 0.008	0.40 ± 0.006	0.69 ± 0.008	0.45 ± 0.002	0.57 ± 0.004
Ours	**HICON**	$\mathbf{0.94 \pm 0.003}$	$\mathbf{0.80 \pm 0.005}$	$\mathbf{0.97 \pm 0.001}$	$\mathbf{0.04 \pm 0.003}$	$\mathbf{0.58 \pm 0.007}$	$\mathbf{0.73 \pm 0.002}$	$\mathbf{0.87 \pm 0.003}$	$\mathbf{0.58 \pm 0.002}$	$\mathbf{0.73 \pm 0.002}$

Table 28: Performance comparison on CPG0036 IMTM_HepG2 .

Type	Method	Batch correction		Bio metrics				Aggregate score		
		Graph connectivity	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Harmony	0.65	0.76	0.97	0.01	0.38	0.39	0.71	0.44	0.57
	fastMNN	0.63	0.76	0.97	0.01	0.39	0.41	0.70	0.45	0.57
	ComBat	0.64	0.73	0.97	0.01	0.39	0.37	0.69	0.44	0.56
	MNN	0.64	0.74	0.97	0.01	0.39	0.36	0.69	0.43	0.56
	Seurat RPCA	0.60	0.70	0.98	0.01	0.38	0.39	0.65	0.44	0.55
	Seurat CCA	0.61	0.67	0.98	0.01	0.35	0.41	0.64	0.44	0.54
	Baseline	0.62	0.68	0.97	0.01	0.36	0.35	0.65	0.42	0.54
	Scanorama	0.52	0.68	0.98	0.01	0.41	0.32	0.60	0.43	0.52
	Sphering	0.62	0.66	0.97	0.01	0.38	0.19	0.64	0.39	0.52
Deep	scVI	0.60 ± 0.007	0.75 ± 0.008	0.97 ± 0.001	0.02 ± 0.001	0.44 ± 0.008	0.39 ± 0.004	0.68 ± 0.005	0.45 ± 0.002	0.57 ± 0.003
	cpDistiller	0.57 ± 0.011	0.74 ± 0.011	0.97 ± 0.001	0.01 ± 0.002	0.37 ± 0.012	0.33 ± 0.007	0.66 ± 0.008	0.42 ± 0.004	0.54 ± 0.004
	DESC	0.46 ± 0.015	0.76 ± 0.014	0.97 ± 0.001	0.01 ± 0.002	0.41 ± 0.011	0.27 ± 0.008	0.61 ± 0.011	0.42 ± 0.003	0.52 ± 0.006
	scBCN	0.49 ± 0.013	0.63 ± 0.012	0.97 ± 0.001	0.01 ± 0.002	0.39 ± 0.010	0.29 ± 0.007	0.56 ± 0.009	0.42 ± 0.003	0.49 ± 0.005
Ours	**HICON**	$\mathbf{0.88 \pm 0.004}$	$\mathbf{0.79 \pm 0.007}$	$\mathbf{0.97 \pm 0.001}$	$\mathbf{0.03 \pm 0.004}$	$\mathbf{0.54 \pm 0.010}$	$\mathbf{0.59 \pm 0.003}$	$\mathbf{0.84 \pm 0.004}$	$\mathbf{0.53 \pm 0.003}$	$\mathbf{0.69 \pm 0.002}$

Table 29: Performance comparison on CPG0036 MEDINA_HepG2 .

Type	Method	Batch correction		Bio metrics				Aggregate score		
		Graph connectivity	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Harmony	0.63	0.77	0.97	0.01	0.41	0.38	0.70	0.44	0.57
	fastMNN	0.62	0.76	0.97	0.01	0.41	0.40	0.69	0.45	0.57
	ComBat	0.62	0.74	0.97	0.01	0.42	0.34	0.68	0.44	0.56
	MNN	0.62	0.69	0.97	0.02	0.43	0.33	0.66	0.44	0.55
	Seurat CCA	0.62	0.67	0.98	0.01	0.38	0.38	0.65	0.44	0.54
	Seurat RPCA	0.59	0.71	0.98	0.01	0.38	0.36	0.65	0.43	0.54
	Baseline	0.61	0.66	0.97	0.01	0.41	0.32	0.63	0.43	0.53
	Scanorama	0.52	0.72	0.98	0.01	0.43	0.33	0.62	0.44	0.53
	Sphering	0.61	0.64	0.97	0.01	0.42	0.18	0.62	0.39	0.51
Deep	scVI	0.58 ± 0.006	0.73 ± 0.007	0.97 ± 0.001	0.02 ± 0.001	0.45 ± 0.007	0.33 ± 0.004	0.65 ± 0.005	0.44 ± 0.002	0.55 ± 0.003
	cpDistiller	0.57 ± 0.010	0.74 ± 0.010	0.97 ± 0.001	0.01 ± 0.002	0.39 ± 0.011	0.30 ± 0.006	0.65 ± 0.007	0.42 ± 0.003	0.54 ± 0.004
	DESC	0.44 ± 0.014	0.70 ± 0.013	0.97 ± 0.001	0.01 ± 0.002	0.41 ± 0.010	0.22 ± 0.007	0.57 ± 0.010	0.40 ± 0.003	0.49 ± 0.005
	scBCN	0.49 ± 0.012	0.52 ± 0.011	0.97 ± 0.001	0.01 ± 0.002	0.42 ± 0.009	0.25 ± 0.006	0.50 ± 0.008	0.41 ± 0.003	0.46 ± 0.004
Ours	**HICON**	**0.92 ± 0.004**	**0.77 ± 0.006**	**0.97 ± 0.001**	**0.04 ± 0.004**	**0.57 ± 0.009**	**0.63 ± 0.003**	**0.85 ± 0.004**	**0.55 ± 0.003**	**0.70 ± 0.002**

Table 30: Performance comparison on CPG0036 USC_HepG2 .

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	fastMNN	0.60	0.14	0.52	0.95	0.02	0.23	0.50	0.42	0.43	0.42
	Harmony	0.56	0.09	0.52	0.95	0.01	0.24	0.46	0.39	0.42	0.40
	Sphering	0.64	0.15	0.46	0.95	0.02	0.25	0.17	0.41	0.35	0.38
	ComBat	0.51	0.07	0.44	0.95	0.01	0.22	0.39	0.34	0.40	0.37
	Baseline	0.52	0.06	0.45	0.95	0.01	0.21	0.39	0.34	0.39	0.37
	MNN	0.51	0.07	0.45	0.95	0.01	0.22	0.39	0.34	0.39	0.37
	Seurat CCA	0.43	0.34	0.31	0.96	0.01	0.17	0.35	0.36	0.37	0.37
	Seurat RPCA	0.43	0.24	0.28	0.96	0.01	0.17	0.34	0.32	0.37	0.35
	Scanorama	0.23	0.14	0.46	0.96	0.01	0.24	0.29	0.28	0.37	0.32
Deep	cpDistiller	0.54 ± 0.002	0.09 ± 0.006	0.49 ± 0.009	0.95 ± 0.001	0.02 ± 0.002	0.25 ± 0.010	0.46 ± 0.006	0.37 ± 0.004	0.42 ± 0.004	0.40 ± 0.002
	scVI	0.41 ± 0.007	0.03 ± 0.009	0.54 ± 0.001	0.95 ± 0.001	0.02 ± 0.001	0.28 ± 0.008	0.45 ± 0.002	0.33 ± 0.003	0.42 ± 0.002	0.38 ± 0.001
	scBCN	0.28 ± 0.007	0.00 ± 0.000	0.44 ± 0.010	0.94 ± 0.000	0.01 ± 0.002	0.28 ± 0.008	0.28 ± 0.009	0.24 ± 0.001	0.38 ± 0.005	0.31 ± 0.002
	DESC	0.21 ± 0.008	0.00 ± 0.003	0.41 ± 0.006	0.95 ± 0.001	0.01 ± 0.003	0.19 ± 0.008	0.25 ± 0.004	0.21 ± 0.003	0.35 ± 0.003	0.28 ± 0.003
Ours	**HICON**	**0.84 ± 0.004**	**0.41 ± 0.007**	**0.54 ± 0.003**	**0.96 ± 0.000**	**0.06 ± 0.005**	**0.46 ± 0.007**	**0.74 ± 0.001**	**0.60 ± 0.001**	**0.56 ± 0.003**	**0.58 ± 0.002**

Table 31: Performance comparison on RxRx1 OpenPhenom HepG2.

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Scanorama	0.21	0.90	0.27	0.96	0.02	0.22	0.47	0.46	0.42	0.44
	Sphering	0.77	0.11	0.32	0.96	0.03	0.29	0.33	0.40	0.40	0.40
	Harmony	0.55	0.06	0.35	0.96	0.02	0.23	0.62	0.32	0.46	0.39
	ComBat	0.56	0.06	0.34	0.96	0.02	0.23	0.60	0.32	0.45	0.39
	Baseline	0.56	0.05	0.34	0.96	0.02	0.21	0.60	0.32	0.45	0.38
	MNN	0.56	0.04	0.34	0.96	0.02	0.24	0.59	0.31	0.45	0.38
	Seurat CCA	0.39	0.36	0.18	0.97	0.01	0.20	0.53	0.31	0.43	0.37
	fastMNN	0.45	0.04	0.35	0.96	0.01	0.16	0.53	0.28	0.42	0.35
	Seurat RPCA	0.43	0.03	0.16	0.97	0.02	0.21	0.53	0.21	0.43	0.32
Deep	cpDistiller	0.61 ± 0.015	0.06 ± 0.004	0.35 ± 0.005	0.96 ± 0.001	0.02 ± 0.002	0.26 ± 0.017	0.63 ± 0.016	0.34 ± 0.002	0.47 ± 0.009	0.40 ± 0.006
	scVI	0.42 ± 0.004	0.02 ± 0.003	0.35 ± 0.003	0.95 ± 0.001	0.02 ± 0.001	0.21 ± 0.013	0.58 ± 0.002	0.26 ± 0.003	0.44 ± 0.004	0.35 ± 0.003
	scBCN	0.27 ± 0.006	0.02 ± 0.000	0.36 ± 0.003	0.95 ± 0.000	0.02 ± 0.000	0.25 ± 0.003	0.49 ± 0.003	0.22 ± 0.002	0.43 ± 0.001	0.32 ± 0.001
	DESC	0.17 ± 0.006	0.00 ± 0.003	0.35 ± 0.003	0.95 ± 0.001	0.02 ± 0.001	0.25 ± 0.013	0.51 ± 0.003	0.17 ± 0.002	0.43 ± 0.004	0.30 ± 0.003
Ours	**HICON**	**0.92 ± 0.002**	**0.21 ± 0.013**	**0.35 ± 0.004**	**0.97 ± 0.000**	**0.11 ± 0.008**	**0.56 ± 0.008**	**0.88 ± 0.001**	**0.49 ± 0.005**	**0.63 ± 0.004**	**0.56 ± 0.001**

Table 32: Performance comparison on RxRx1 OpenPhenom HUVEC.

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Sphering	0.74	0.38	0.52	0.95	0.02	0.32	0.27	0.55	0.39	0.47
	Harmony	0.57	0.17	0.56	0.95	0.02	0.28	0.51	0.43	0.44	0.44
	MNN	0.56	0.16	0.53	0.95	0.02	0.28	0.49	0.42	0.44	0.43
	ComBat	0.57	0.16	0.54	0.95	0.02	0.27	0.49	0.42	0.44	0.43
	Baseline	0.57	0.17	0.54	0.95	0.02	0.27	0.49	0.42	0.43	0.43
	Seurat RPCA	0.41	0.60	0.42	0.94	0.00	0.17	0.40	0.48	0.38	0.43
	fastMNN	0.51	0.11	0.54	0.95	0.01	0.23	0.46	0.39	0.41	0.40
	Seurat CCA	0.38	0.52	0.26	0.94	0.00	0.14	0.39	0.39	0.37	0.38
	Scanorama	0.24	0.25	0.49	0.96	0.01	0.23	0.31	0.33	0.38	0.35
Deep	cpDistiller	0.60 ± 0.003	0.15 ± 0.000	0.54 ± 0.004	0.95 ± 0.000	0.03 ± 0.000	0.32 ± 0.006	0.54 ± 0.004	0.43 ± 0.002	0.46 ± 0.002	0.45 ± 0.002
	scVI	0.42 ± 0.006	0.07 ± 0.003	0.55 ± 0.008	0.95 ± 0.001	0.02 ± 0.001	0.27 ± 0.009	0.47 ± 0.004	0.34 ± 0.004	0.43 ± 0.002	0.38 ± 0.003
	scBCN	0.34 ± 0.006	0.03 ± 0.003	0.57 ± 0.002	0.95 ± 0.001	0.02 ± 0.000	0.32 ± 0.001	0.38 ± 0.000	0.31 ± 0.002	0.42 ± 0.000	0.36 ± 0.001
	DESC	0.25 ± 0.005	0.01 ± 0.002	0.56 ± 0.003	0.95 ± 0.001	0.01 ± 0.000	0.28 ± 0.003	0.36 ± 0.002	0.27 ± 0.002	0.40 ± 0.001	0.34 ± 0.002
Ours	**HICON**	**0.88 ± 0.005**	**0.56 ± 0.040**	**0.53 ± 0.007**	**0.97 ± 0.001**	**0.06 ± 0.006**	**0.50 ± 0.008**	**0.80 ± 0.002**	**0.66 ± 0.015**	**0.58 ± 0.004**	**0.62 ± 0.008**

Table 33: Performance comparison on RxRx1 OpenPhenom RPE.

Type	Method	Batch correction			Bio metrics				Aggregate score		
		Graph connectivity	KBET	LISI batch	LISI label	Leiden ARI	Leiden NMI	mAP (nonrep)	Batch correction	Bio metrics	Overall
Non-deep	Harmony	0.61	0.20	0.72	0.95	0.01	0.29	0.39	0.51	0.41	0.46
	ComBat	0.58	0.20	0.62	0.95	0.01	0.30	0.32	0.47	0.40	0.43
	Baseline	0.58	0.20	0.63	0.95	0.01	0.28	0.32	0.47	0.39	0.43
	MNN	0.56	0.20	0.63	0.95	0.01	0.29	0.31	0.46	0.39	0.43
	fastMNN	0.59	0.00	0.73	0.95	0.01	0.28	0.39	0.44	0.41	0.42
	Sphering	0.60	0.20	0.66	0.95	0.01	0.28	0.12	0.48	0.34	0.41
	Seurat CCA	0.52	0.20	0.41	0.95	0.01	0.26	0.28	0.38	0.37	0.37
	Scanorama	0.34	0.00	0.63	0.96	0.01	0.37	0.23	0.33	0.39	0.36
	Seurat RPCA	0.49	0.00	0.35	0.95	0.01	0.23	0.25	0.28	0.36	0.32
Deep	cpDistiller	0.59 ± 0.009	0.40 ± 0.000	0.67 ± 0.033	0.95 ± 0.000	0.02 ± 0.001	0.34 ± 0.011	0.38 ± 0.002	0.55 ± 0.008	0.42 ± 0.003	0.49 ± 0.006
	DESC	0.44 ± 0.009	0.20 ± 0.000	0.61 ± 0.029	0.95 ± 0.001	0.01 ± 0.001	0.33 ± 0.004	0.29 ± 0.005	0.42 ± 0.008	0.40 ± 0.003	0.41 ± 0.004
	scVI	0.50 ± 0.009	0.00 ± 0.000	0.70 ± 0.014	0.95 ± 0.001	0.02 ± 0.001	0.34 ± 0.004	0.34 ± 0.005	0.40 ± 0.005	0.41 ± 0.001	0.41 ± 0.003
	scBCN	0.37 ± 0.002	0.00 ± 0.000	0.64 ± 0.029	0.94 ± 0.002	0.02 ± 0.001	0.39 ± 0.003	0.22 ± 0.008	0.34 ± 0.009	0.39 ± 0.003	0.36 ± 0.004
Ours	**HICON**	$\mathbf{0.85 \pm 0.006}$	$\mathbf{0.60 \pm 0.115}$	$\mathbf{0.76 \pm 0.009}$	$\mathbf{0.95 \pm 0.001}$	$\mathbf{0.04 \pm 0.004}$	$\mathbf{0.45 \pm 0.009}$	$\mathbf{0.67 \pm 0.004}$	$\mathbf{0.74 \pm 0.043}$	$\mathbf{0.53 \pm 0.004}$	$\mathbf{0.63 \pm 0.023}$

Table 34: Performance comparison on RxRx1 OpenPhenom U2OS.