CC-TTRL: CONTROL-CONDITIONED TEST-TIME SELECTION OF NORMALIZATION STATISTICS UNDER LABEL SHIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time adaptation methods based on Batch Normalization can correct batch-specific technical variation using unlabeled target samples, but their statistics become biased when the target class distribution shifts. In biomedical imaging, negative controls provide a stable reference for separating technical variation from perturbation-specific signal. CS-ARM-BN exploits this structure by pooling controls and perturbed samples when recomputing BatchNorm statistics, yielding a favorable global bias-variance trade-off. However, the same estimator is applied to every normalization layer and every target batch, despite the fact that class-dependent activation bias varies across network depth and target composition. We introduce CC-TTRL, a control-conditioned test-time adaptation method that selects the source of normalization statistics independently across network stages. For each stage, CC-TTRL chooses among pooled target statistics, control-only statistics, and stored training statistics, producing a discrete family of adaptations that includes CS-ARM-BN as a special case. Candidate configurations are evaluated without target labels using a cross-fitted reward that combines agreement between held-out target controls and source controls with prediction consistency on held-out perturbed samples. A trust-region constraint and improvement veto preserve the CS-ARM-BN solution when evidence for an alternative configuration is insufficient. We search this action space using a lightweight policy-gradient procedure. We evaluate CC-TTRL on mechanism-of-action prediction from high-content cellular microscopy under experimental-batch shift and label shift. Under severe label shift, CC-TTRL improves over CS-ARM-BN and preferentially avoids pooled statistics in deeper network stages, consistent with increased class-dependent activation leakage at greater depth. The proposed control-anchored reward correlates with held-out predictive accuracy and substantially outperforms a feature-space reward. Random search with the same reward and evaluation budget achieves comparable performance to policy-gradient search, indicating that the primary contribution is the layerwise adaptation space and label-free control-guided selection criterion rather than reinforcement learning itself.

1 INTRODUCTION

High-content cellular microscopy provides a scalable way to characterize cellular morphology and perturbation response, supporting applications such as mechanism-of-action prediction, drug–target discovery, and large-scale phenotypic profiling (Chandrasekaran et al., 2024; Kraus et al., 2025; Moshkov et al., 2024; Kraus et al., 2024; Kim et al., 2025; Kenyon-Dean et al., 2024; Agrawal et al., 2026; Lu et al., 2026). Recent progress in self-supervised representation learning, foundation models, contrastive learning, and generative modeling has further increased the biological information that can be extracted from microscopy images (Kraus et al., 2024; Moshkov et al., 2024; Kim et al., 2025; Pham & Plummer, 2024; Palma et al., 2025; Zhang et al., 2025; Navidi et al., 2025). A persistent obstacle, however, is the presence of batch effects: non-biological variation introduced by experimental plates, acquisition conditions, staining protocols, microscopes, and data-generating sites. Such variation can dominate the biological signal of interest and substantially reduce perfor-

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.

mance on unseen experimental batches (Arevalo et al., 2024; Haslum et al., 2024; Yan et al., 2025; Ravi et al., 2026; Tong et al., 2026; Sanchez-Fernandez et al., 2026).

Test-time adaptation (TTA) provides a natural framework for addressing this problem by allowing a trained model to exploit unlabeled samples from the target domain at inference time. Batch Normalization (BN) (Ioffe & Szegedy, 2015) is particularly attractive in this setting because its statistics can be recomputed from target samples without retraining the full network. This principle underlies normalization-based adaptation and has also been incorporated into meta-learning frameworks such as Adaptive Risk Minimization (ARM), in which models are explicitly trained to adapt to unseen domains from unlabeled context (Zhang et al., 2021; Finn et al., 2017). However, target-derived normalization statistics reflect not only technical domain shift but also the composition of the target data. Under label shift, class-dependent activations can contaminate estimates of target means and variances, causing adaptation itself to introduce error. Recent work has consequently studied the sensitivity of normalization-based TTA to label imbalance, temporally correlated classes, unreliable target statistics, and prediction uncertainty (Tomar et al., 2024; Su et al., 2024; Vianna et al., 2024; Lee et al., 2024; Marsden et al., 2024).

Biomedical screening experiments provide additional structure that can be exploited during adaptation. Experimental plates commonly contain negative controls consisting of vehicle-treated (DMSO) samples acquired under the same technical conditions as the perturbed samples but without an active biological perturbation. Such controls provide a reference for technical variation and have been used for normalization and batch correction in image-based profiling (Lin & Lu, 2022; Arevalo et al., 2024). CS-ARM-BN incorporates this structure into meta-learned test-time adaptation by jointly using target controls and perturbed samples to estimate BN statistics (Sanchez-Fernandez et al., 2026). If a target batch contains C controls and L perturbed samples, pooling the two groups increases the effective sample size while reducing the class-dependent bias associated with estimating statistics from perturbed samples alone. This produces a favorable bias–variance trade-off under small target batches and label shift (Sanchez-Fernandez et al., 2026).

Nevertheless, CS-ARM-BN applies the same pooled estimator to every BN layer and every target batch. This assumes that one bias–variance trade-off is appropriate throughout the network. We instead consider three possible sources of normalization statistics: pooled controls and perturbed samples (P), controls only (C), and stored training statistics (S). These alternatives make different compromises. Pooled statistics use the largest target context but remain partially influenced by its class composition; control-only statistics avoid direct contamination from the perturbed-label distribution but are estimated from fewer samples; and stored statistics avoid target-sampling noise but retain the original source–target domain mismatch. We hypothesize that the relative usefulness of these estimators changes with both target-batch composition and network depth, rather than being captured by a single global choice.

We introduce CC-TTRL, a control-conditioned test-time adaptation method that selects the source of normalization statistics independently across the five stages of a ResNet. Each stage chooses from $\{P, C, S\}$, producing $3^5 = 243$ possible configurations. The all-pooled configuration (P, P, P, P, P) is exactly CS-ARM-BN, so the existing adaptation remains explicitly contained in the action space. CC-TTRL does not modify the network weights. Instead, for each unseen target batch, it searches for a stage-wise normalization configuration using only the unlabeled perturbed samples and negative controls available in that batch. Figure 1 summarizes the procedure.

The central challenge is selecting among these configurations without access to target labels. CC-TTRL therefore uses negative controls as a label-free validation signal. Controls are split once per target batch into two fixed halves: one contributes to normalization statistics and the other is reserved for scoring. In contrast, the 36 perturbed samples are divided into two groups of 18 and cross-fit, with the two groups alternating between normalization context and held-out evaluation. Consequently, held-out perturbed samples never contribute to the BN statistics used for their own prediction.

Each candidate is evaluated using two complementary signals. First, a control-anchor score measures how closely predictions induced by the held-out target controls match the reference profile of source controls and how consistently the fixed held-out control set behaves across its two scoring subsets. Second, an agreement score evaluates whether predictions on held-out perturbed samples agree with the majority prediction across candidate configurations. Agreement is not used alone

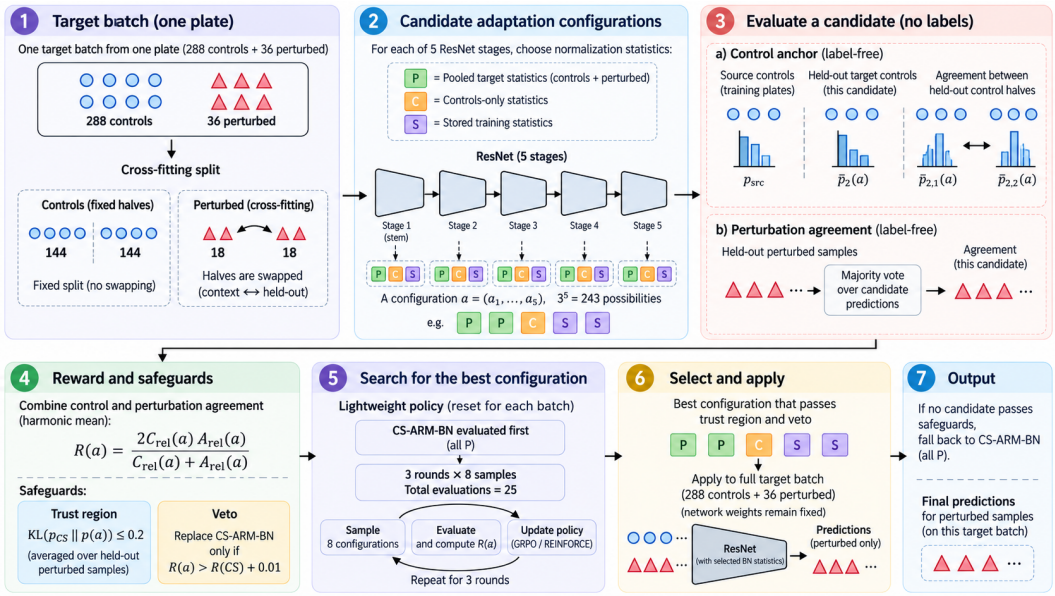


Figure 1: Overview of CC-TTRL. Each target batch contains 36 perturbed samples and 288 negative controls from one plate. The controls are divided once into fixed halves of 144 samples for normalization and held-out scoring, while the perturbed samples are divided 18/18 and cross-fit so that the two halves alternate between normalization context and held-out evaluation. For each of five ResNet stages, a candidate chooses pooled target statistics (P), control-only statistics (C), or stored training statistics (S), yielding $3^5 = 243$ configurations and containing CS-ARM-BN as the all-P configuration. Candidates are evaluated without target labels using control anchoring and prediction agreement on held-out perturbed samples. Their harmonic mean defines the reward. A trust region constrains the average $\mathrm{KL}(p_{\mathrm{CS}} \| p_a)$ over held-out perturbed samples to at most 0.2, and a veto requires a reward improvement greater than 0.01 over CS-ARM-BN. CS-ARM-BN is evaluated first, followed by three search rounds with eight configurations per round, for 25 evaluations in total. The selected configuration is then applied to the full target batch to predict the 36 perturbed samples; controls are used only as normalization context and are not classified.

because a consensus can still be systematically incorrect. CC-TTRL therefore combines the two relative scores using a harmonic mean. Selection is further constrained by a prediction-space trust region,

$$\frac{1}{|X_{\mathrm{held}}|} \sum_{x \in X_{\mathrm{held}}} D_{\mathrm{KL}}\left(p_{\mathrm{CS}}(\cdot \mid x) \,\|\, p_a(\cdot \mid x)\right) \le 0.2,$$

computed over held-out perturbed samples, together with an improvement veto that replaces CS-ARM-BN only when the best admissible candidate exceeds its reward by more than 0.01.

We search the discrete action space with a lightweight group-normalized policy-gradient procedure. CS-ARM-BN is evaluated first as the reference configuration. The policy then samples eight candidate configurations in each of three rounds, resulting in $1 + 3 \times 8 = 25$ evaluations per target batch. The policy maintains one categorical distribution over $\{P, C, S\}$ for each network stage and is reset independently for every batch. However, the policy-gradient search is not the source of the improvement: random search with the same evaluation budget reaches the same accuracy (0.909 versus 0.909, $p = 0.87$). This isolates the primary contribution as the combination of the stage-wise normalization action space and the control-guided, label-free selection criterion rather than the particular search optimizer.

We evaluate CC-TTRL on mechanism-of-action prediction from high-content cellular microscopy using the control-based adaptation setting introduced by CS-ARM-BN (Sanchez-Fernandez et al., 2026). Under severe label shift with $\alpha = 0.01$, CC-TTRL improves our reproduced CS-ARM-BN baseline from 0.886 to 0.909, corresponding to a 2.3 percentage-point improvement, and improves

performance on 165 of 172 target plates. The selected configurations also exhibit a systematic dependence on network depth: pooled statistics remain common in early stages, whereas the deepest stage abandons pooled statistics in 84% of target batches. The proposed reward correlates with held-out predictive accuracy with Spearman $\rho = 0.30$, compared with $\rho = 0.06$ for a feature-space alternative. These results indicate that a fixed normalization strategy can be unnecessarily restrictive under label shift and that negative controls can provide a useful signal for selecting adaptation behavior without target labels.

Our contributions are:

- We formulate test-time normalization as a stage-wise selection problem over pooled, control-only, and stored statistics, with CS-ARM-BN included explicitly in the resulting action space.
- We introduce a cross-fitted, label-free selection criterion combining negative-control anchoring with prediction agreement on perturbed samples, together with conservative trust-region and fallback mechanisms.
- We show that the preferred normalization source varies systematically across network depth and target batches and that conditional selection improves performance under severe label shift without modifying model weights.
- We show through search and reward ablations that the gains arise from the adaptation space and control-guided reward rather than the policy-gradient optimizer itself.

2 Related Work

Test-time and normalization-based adaptation. Batch Normalization (BN) was originally introduced to stabilize deep-network optimization Ioffe & Szegedy (2015), but its dependence on activation statistics also provides a convenient mechanism for domain adaptation. AdaBN replaces source-domain BN statistics with statistics estimated from the target domain **?**, while test-time training and adaptation methods update models using unlabeled target samples during inference **??**. Adaptive Risk Minimization (ARM) further meta-learns models that are explicitly trained to adapt from unlabeled test-time context Zhang et al. (2021). Subsequent work has shown that normalization-based adaptation can become unstable when target batches are small, class-imbalanced, temporally correlated, or affected by label shift **????????**. Recent approaches therefore modify how target statistics are estimated or which features are adapted, including meta-learned BN adaptation **?**, small-batch calibration **?**, robust normalization **?**, channel-selective normalization **?**, and distributional recalibration beyond first- and second-order moments **?**.

Biomedical batch effects and experimental controls. Batch effects are particularly severe in high-content biomedical imaging, where changes in laboratory conditions, plates, reagents, staining, and imaging systems can induce substantial non-biological variation **?**Arevalo et al. (2024). Negative controls are routinely included in such experiments and provide samples acquired under the same technical conditions but without the biological perturbation of interest. CS-ARM-BN exploits this structure by incorporating both negative controls and perturbed samples into BN adaptation and meta-training the model to use this control-conditioned context **?**. This improves robustness to small target batches and label-shifted target distributions, but the adaptation rule itself remains fixed.

Test-time reinforcement learning. Recent work has explored reinforcement learning directly at inference time. TTRL constructs self-supervised rewards from multiple model generations and updates the model on unlabeled test inputs **?**, while TTRV extends test-time RL to vision-language models using group-relative policy optimization and rewards derived from model agreement **?**. Unlike these approaches, our setting provides an external experimental signal: negative controls observed under the same target domain. We therefore use held-out controls to define a label-free reward and learn a test-time policy that decides how the normalization mechanism should adapt to each unseen experimental batch.

3 METHOD

3.1 PROBLEM SETTING

We consider a model $g(\cdot; w)$ trained on multiple experimental source batches and evaluated on an unseen target batch β. Following ARM and CS-ARM-BN (Zhang et al., 2021; Sanchez-Fernandez et al., 2026), adaptation is performed using unlabeled samples from the target batch while the network weights w remain fixed.

Each target batch contains

$$X^\beta = \{x_1, \ldots, x_L\}, \qquad L = 36,$$

perturbed samples whose mechanism-of-action labels must be predicted, together with

$$Z^\beta = \{z_1, \ldots, z_C\}, \qquad C = 288,$$

vehicle-treated DMSO controls from the same plate. The controls capture batch-specific technical variation but are not part of the final mechanism-of-action evaluation.

CS-ARM-BN adapts all BatchNorm layers using statistics estimated from the pooled target context,

$$X^\beta \cup Z^\beta.$$

Pooling reduces estimation variance and dilutes the influence of the perturbed class distribution, but applies the same estimator to every layer and target batch (Sanchez-Fernandez et al., 2026). CC-TTRL instead treats the normalization-statistics source as a batch-specific, stage-wise decision.

3.2 STAGE-WISE NORMALIZATION ACTION SPACE

We group ResNet BatchNorm layers into five stages: the stem and four residual stages. For each stage $s \in \{1, \ldots, 5\}$, CC-TTRL chooses

$$a_s \in \{P, C, S\},$$

where

$$P = \text{pooled target statistics}, \qquad C = \text{controls-only statistics}, \qquad S = \text{stored training statistics}.$$

A complete configuration is

$$a = (a_1, \ldots, a_5) \in \{P, C, S\}^5, \qquad |\mathcal{A}| = 3^5 = 243.$$

The configuration

$$a_0 = (P, P, P, P, P)$$

is exactly CS-ARM-BN; we verified that it produces bit-identical outputs to the original implementation. Hence, the baseline is explicitly contained in the search space and can always be retained. Here, S denotes the stored running BatchNorm statistics learned during training, not target statistics.

For LeViT, whose BatchNorm layers normalize patch tokens, we use the same three statistics sources over five stages: the patch embedding, the three encoder stages, and the classifier head.

Figure 1 summarizes the complete procedure.

3.3 HELD-OUT EVALUATION BY CROSS-FITTING

Candidate configurations must be evaluated without target labels and without scoring perturbed samples using BatchNorm statistics to which those samples contributed. We therefore separate normalization context from held-out evaluation.

The 288 controls are split once per target batch into two fixed halves,

$$Z^\beta = Z_{\text{ctx}} \cup Z_{\text{hold}}, \qquad |Z_{\text{ctx}}| = |Z_{\text{hold}}| = 144.$$

This split remains fixed throughout the search and the control halves are not exchanged.

The 36 perturbed samples are divided into two equal subsets,

$$X^{\beta} = X^{(1)} \cup X^{(2)}, \qquad |X^{(1)}| = |X^{(2)}| = 18.$$

These subsets are cross-fit. In the first fold,

$$X_{\mathrm{ctx}} = X^{(1)}, \qquad X_{\mathrm{hold}} = X^{(2)},$$

and their roles are exchanged in the second fold.

For candidate a, the normalization context at stage s is

$$\mathcal{N}_s(a_s) = \begin{cases} Z_{\mathrm{ctx}} \cup X_{\mathrm{ctx}}, & a_s = P, \\ Z_{\mathrm{ctx}}, & a_s = C, \\ \text{stored training statistics}, & a_s = S. \end{cases}$$

Only held-out perturbations are used to evaluate perturbed-sample behavior, with four image views per perturbation. Cross-fitting therefore ensures that every perturbed sample is scored under statistics estimated without that sample.

3.4 Control-Anchor Score

Negative controls provide a label-free reference for whether an adaptation preserves the expected behavior of vehicle-treated cells. We construct a source reference profile

$$p_{\mathrm{src}}$$

from the softmax outputs of controls in the source batches.

For candidate a, let $\bar{p}_Z(a)$ denote the mean softmax profile of the held-out target controls. Splitting Z_{hold} into two disjoint halves Z_1 and Z_2, we define

$$D(a) = \mathrm{JS}\left(p_{\mathrm{src}}, \bar{p}_Z(a)\right) + \mathrm{JS}\left(\bar{p}_{Z_1}(a), \bar{p}_{Z_2}(a)\right).$$

The first term measures alignment with source controls, while the second measures consistency within the target controls. The score is expressed relative to CS-ARM-BN,

$$C_{\mathrm{rel}}(a) = \frac{D(a_0)}{D(a)}.$$

Candidates whose control predictions are both source-aligned and internally consistent therefore receive larger values.

3.5 Perturbation Agreement

The control anchor alone could favor a configuration that aligns controls while degrading perturbed predictions. We therefore introduce a complementary agreement signal computed only from held-out perturbations.

For each held-out sample x, the evaluated configurations produce predicted labels $\hat{y}_a(x)$. We form a consensus pseudo-label using majority vote,

$$\tilde{y}(x) = \mathrm{mode}_{a \in \mathcal{E}}\, \hat{y}_a(x),$$

where $\mathcal{E}$ denotes the evaluated configurations.

Candidate agreement is

$$A(a) = \frac{1}{|X_{\mathrm{hold}}|} \sum_{x \in X_{\mathrm{hold}}} \mathbb{I}\left[\hat{y}_a(x) = \tilde{y}(x)\right],$$

and is normalized relative to CS-ARM-BN,

$$A_{\mathrm{rel}}(a) = \frac{A(a)}{A(a_0)}.$$

All agreement calculations use held-out perturbations and their four views. Agreement is not used alone because consensus predictions can still be incorrect under distribution shift.

3.6 Decision-Space Reward

The two label-free signals are combined using their harmonic mean,

$$R(a) = \frac{2C_{\mathrm{rel}}(a)A_{\mathrm{rel}}(a)}{C_{\mathrm{rel}}(a) + A_{\mathrm{rel}}(a)}.$$

This penalizes configurations that perform well on only one criterion. Since both terms are defined relative to CS-ARM-BN,

$$R(a_0) \approx 1.$$

The reward operates entirely in prediction space: both the control reference and agreement signal are defined from model outputs rather than feature-space distances.

3.7 Trust Region and Veto

KL divergence is not part of the reward. Instead, it defines a trust region that prevents large deviations from the CS-ARM-BN predictions. For each held-out perturbation x and each of its four views v,

$$T(a) = \frac{1}{4|X_{\mathrm{hold}}|} \sum_{x \in X_{\mathrm{hold}}} \sum_{v=1}^{4} D_{\mathrm{KL}}\left(p_{a_0}(\cdot \mid x, v) \,\|\, p_a(\cdot \mid x, v)\right).$$

Thus, CS-ARM-BN is the reference distribution in the KL divergence. A candidate is eligible only if

$$T(a) \leq 0.2.$$

Let

$$a_{\mathrm{TR}} = \arg \max_{a: T(a) \leq 0.2} R(a)$$

denote the highest-reward candidate satisfying the trust region. CC-TTRL replaces CS-ARM-BN only if

$$R(a_{\mathrm{TR}}) > R(a_0) + 0.01.$$

Otherwise,

$$a^\star = a_0.$$

A candidate must therefore satisfy both the KL constraint and the reward-margin veto before replacing the baseline.

3.8 Policy-Guided Search

CC-TTRL uses a lightweight factorized policy over the 243 stage-wise configurations,

$$\pi_\theta(a) = \prod_{s=1}^{5} \pi_{\theta_s}(a_s),$$

with one categorical distribution over $\{P, C, S\}$ per stage. The policy is reset independently for every target batch and initialized as

$$\pi_{\theta_s}(P) = 0.50, \qquad \pi_{\theta_s}(C) = 0.25, \qquad \pi_{\theta_s}(S) = 0.25.$$

This initialization favors the CS-ARM-BN choice while retaining substantial probability on the alternative statistics sources.

CS-ARM-BN is evaluated first and provides the reference quantities used by the reward, trust region, and veto. The search then runs for three rounds with $K = 8$ sampled configurations per round. Within each round,

$$\hat{A}_i = \frac{R(a_i) - \frac{1}{K}\sum_{j=1}^{K} R(a_j)}{\mathrm{std}_j\left[R(a_j)\right]},$$

and the policy is updated using

$$\theta \leftarrow \theta + \eta \frac{1}{K} \sum_{i=1}^{K} \hat{A}_i \nabla_\theta \log \pi_\theta(a_i).$$

Only the search-policy parameters are updated; the network weights remain fixed. The total search budget is

$$1 + 3 \times 8 = 25$$

evaluations per target batch: one CS-ARM-BN reference and 24 sampled configurations.

After the search, the highest-reward configuration satisfying both safeguards is selected; otherwise CS-ARM-BN is retained. Cross-fitting is used only for configuration selection. For final prediction, normalization statistics are recomputed from the complete target batch,

$$\mathcal{N}_s^{\text{final}}(a_s^\star) = \begin{cases} Z^\beta \cup X^\beta, & a_s^\star = P, \\ Z^\beta, & a_s^\star = C, \\ \text{stored training statistics}, & a_s^\star = S. \end{cases}$$

Thus, P uses all 288 controls and 36 perturbed samples, C uses all 288 controls, and S uses the stored training statistics. The selected configuration is then used to generate the final predictions for the 36 perturbed samples.

4 RESULTS

We compare CC-TTRL with CS-ARM-BN across the evaluation settings of Sanchez-Fernandez et al. (2026). CS-ARM-BN was recomputed on our server on the same target batches as CC-TTRL; Δ is relative to this matched baseline. Published values are shown for reference.

Table 1: MoA accuracy across evaluation settings. "Published" denotes the reported CS-ARM-BN result and "Recomputed" our matched rerun.

Setting	α	Published	Recomputed	CC-TTRL	Δ
New experimental batch					
New batch	–	0.930 ± 0.019	0.928	0.923 ± 0.021	-0.006
Cross-source transfer					
S3→S8	–	0.884 ± 0.005	0.882	0.881 ± 0.008	-0.001
S8→S3	–	0.776 ± 0.016	0.740	0.730 ± 0.005	-0.010
Within-source label shift					
–	4	0.927 ± 0.017	0.923	0.921 ± 0.021	-0.002
–	1	0.924 ± 0.018	0.919	0.921 ± 0.022	$+0.002$
–	0.1	0.906 ± 0.021	0.900	0.912 ± 0.022	$+0.012$
–	0.01	0.894 ± 0.022	0.886	0.906 ± 0.024	$+0.020$
Cross-source label shift					
S3→S8	1	0.878 ± 0.006	0.872	0.876 ± 0.009	$+0.003$
S3→S8	0.01	0.825 ± 0.007	0.824	0.852 ± 0.014	$+0.028$
S8→S3	1	0.788 ± 0.014	0.764	0.762 ± 0.001	-0.001
S8→S3	0.01	0.730 ± 0.015	0.703	0.717 ± 0.005	$+0.014$
LeViT cross-source label shift					
S3→S8	1	0.791 ± 0.025	0.803	0.807 ± 0.014	$+0.004$
S3→S8	0.01	0.707 ± 0.021	0.718	0.750 ± 0.010	$+0.033$
S8→S3	1	0.722 ± 0.015	0.725	0.721 ± 0.015	-0.004
S8→S3	0.01	0.610 ± 0.017	0.635	0.655 ± 0.002	$+0.020$

CC-TTRL does not improve the standard new-batch or cross-source settings, but its advantage emerges as label shift increases. Within-source, it improves the matched baseline by 1.2 points at $\alpha = 0.1$ and 2.0 points at $\alpha = 0.01$. Under cross-source label shift, gains reach 2.8 and 1.4 points, while LeViT shows gains of 3.3 and 2.0 points at $\alpha = 0.01$. The results therefore localize the benefit of CC-TTRL to the severe label-shift regime for which it is designed.

5 Conclusion

CC-TTRL selects pooled, control-only, or stored BatchNorm statistics independently across network stages using a label-free control-guided reward. While it provides no benefit under ordinary batch or source shift, it consistently improves CS-ARM-BN under severe label shift, including across sources and with LeViT, with gains up to 3.3 percentage points. These results support conditional normalization selection when target class composition makes pooled statistics unreliable.

References

Vidit Agrawal, John Peters, Tyler Thompson, Mohammad Sanian, Chau Pham, Nikita Moshkov, Arshad Kazi, Aditya Pillai, Jack Freeman, Byunguk Kang, et al. Chammi-75: Pre-training multi-channel models with heterogeneous microscopy images. In *International Conference on Learning Representations*, volume 2026, pp. 81574–81622, 2026.

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, 2024.

Srinivas Niranj Chandrasekaran, Beth A Cimini, Amy Goodale, Lisa Miller, Maria Kost-Alimova, Nasim Jamali, John G Doench, Briana Fritchman, Adam Skepner, Michelle Melanson, et al. Three million images and morphological profiles of cells treated with matched chemical and genetic perturbations. *Nature Methods*, 21(6):1114–1121, 2024.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Johan Fredin Haslum, Christos Matsoukas, Karl-Johan Leuchowius, and Kevin Smith. Bridging generalization gaps in high content imaging through online self-supervised domain adaptation. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7723–7732. IEEE, 2024.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Kian Kenyon-Dean, Zitong Jerry Wang, John Urbanik, Konstantin Donhauser, Jason Hartford, Saber Saberian, Nil Sahin, Ihab Bendidi, Safiye Celik, Marta Fay, et al. Vitally consistent: Scaling biological representation learning for cell microscopy. *arXiv preprint arXiv:2411.02572*, 2024.

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.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11757–11768. IEEE, 2024.

Oren Kraus, Federico Comitani, John Urbanik, Kian Kenyon-Dean, Lakshmanan Arumugam, Saber Saberian, Cas Wognum, Safiye Celik, and Imran S Haque. Rxrx3-core: benchmarking drug-target interactions in high-content microscopy. *arXiv preprint arXiv:2503.20158*, 2025.

Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. In *International Conference on Learning Representations*, volume 2024, pp. 46977–47002, 2024.

Alexander Lin and Alex Lu. Incorporating knowledge of plates in batch normalization improves generalization of deep learning for microscopy images. In *Machine Learning in Computational Biology*, pp. 74–93. PMLR, 2022.

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.

Robert A Marsden, Mario Döbler, and Bin Yang. Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2543–2553. IEEE, 2024.

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.

Zeinab Navidi, Jun Ma, Esteban Miglietta, Le Liu, Anne Carpenter, Beth Cimini, Benjamin Haibe-Kains, and Bo Wang. Morphodiff: Cellular morphology painting with diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 11613–11637, 2025.

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

Chau Pham and Bryan A Plummer. Enhancing feature diversity boosts channel-adaptive vision transformers. *Advances in Neural Information Processing Systems*, 37:89782–89805, 2024.

Aditya Narayan Ravi, Snehal Vadvalkar, Abhishek Pandey, and Ilan Shomorony. Scalable batch correction for cell painting via batch-dependent kernels and adaptive sampling. *arXiv preprint arXiv:2601.22331*, 2026.

Ana Sanchez-Fernandez, Thomas Pinetz, Werner Zellinger, and Günter Klambauer. Stabilizing in-context multi-source domain adaptation for biomedical images through controls. *arXiv preprint arXiv:2604.20824*, 2026.

Zixian Su, Jingwei Guo, Kai Yao, Xi Yang, Qiufeng Wang, and Kaizhu Huang. Unraveling batch normalization for realistic test-time adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 15136–15144, 2024.

Devavrat Tomar, Guillaume Vray, Jean-Philippe Thiran, and Behzad Bozorgtabar. Un-mixing test-time normalization statistics: Combatting label temporal correlation. In *International Conference on Learning Representations*, volume 2024, pp. 50967–50988, 2024.

Lei Tong, Xujing Yao, Adam Corrigan, Long Chen, Navin Rathna Kumar, Kerry Hallbrook, Jonathan Orme, Yinhai Wang, and Huiyu Zhou. Adversarial batch representation augmentation for batch correction in high-content cellular screening. *Knowledge-Based Systems*, pp. 115829, 2026.

Pedro Vianna, Muawiz Chaudhary, Paria Mehrbod, An Tang, Guy Cloutier, Guy Wolf, Michael Eickenberg, and Eugene Belilovsky. Channel-selective normalization for label-shift robust test-time adaptation. *arXiv preprint arXiv:2402.04958*, 2024.

Chengwei Yan, Yu Zhang, Jiuxin Feng, Heyang Hua, Zhihan Ruan, Zhen Li, Siyu Li, Chaoyang Yan, Pingjing Li, Jian Liu, et al. Triple-effect correction for cell painting data with contrastive and domain-adversarial learning. *Nature Communications*, 16(1):6886, 2025.

Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: Learning to adapt to domain shift. *Advances in neural information processing systems*, 34:23664–23678, 2021.

Yuhui Zhang, Yuchang Su, Chenyu Wang, Tianhong Li, Zoe Wefers, Jeffrey Nirschl, James Burgess, Daisy Ding, Alejandro Lozano, Emma Lundberg, et al. Cellflux: Simulating cellular morphology changes via flow matching. *arXiv preprint arXiv:2502.09775*, 2025.