CROSS-MODAL DRIFTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning a conditional distribution from paired modalities can be challenging when each source observation has only one observed target. For conditional drifting, using that single paired target as the empirical target specifies a point mass rather than the population-conditional. We introduce Cross-Modal Drifting (CMD), which constructs a conditional target proxy from the paired targets of nearby source observations, weighted by their source-space proximity. A kernel-gradient field matches the generator's feature distribution to this proxy, further smoothed using local target-space covariance. The resulting generator produces each sample in a single network evaluation, with no neighborhood retrieval required at inference. Across cross-modal prediction tasks, CMD achieves the lowest mean Energy Score among the evaluated methods, including multi-step flow-matching baselines whose strongest evaluated setting costs fifty times more at inference. A component ablation supports the benefits of the proposed conditional proxy and, in most settings, of local covariance smoothing.

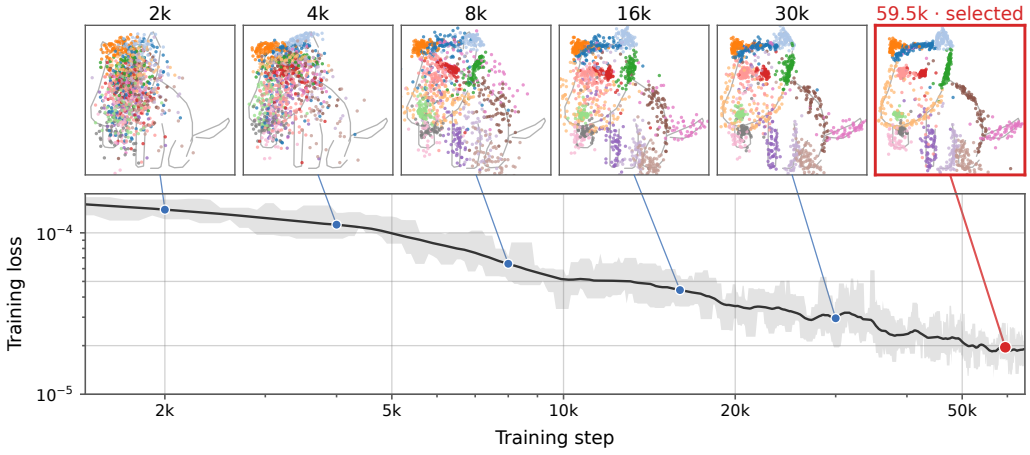

Figure 1: CMD samples across training on the 2-dimensional cat→dog task. A point on cat stroke k is paired with a uniformly drawn point on dog stroke k; the construction is given in Appendix B.8 and Figure 4. Each panel shows the generated points for the 256 test sources (5 draws each) at one checkpoint, coloured by source stroke, over the target drawing in grey. Checkpoints are log-spaced, and the red panel marks the checkpoint selected by early stopping.

1 INTRODUCTION

Many modern generative models learn a transport from a simple reference distribution to a complex data distribution, with optimal transport providing a principled geometry for constructing couplings and probability paths (Peyré & Cuturi, 2019; Lipman et al., 2022; Tong et al., 2023). Diffusion models, score-based models, and continuous normalizing flows implement this transport through time-indexed dynamics, typically requiring repeated neural function evaluations at inference (Ho et al., 2020; Song et al., 2020; Lipman et al., 2022). This sampling cost has motivated methods that

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.

learn to traverse the generative trajectory in a single evaluation. Improved consistency training (iCT) maps points along a probability-flow trajectory to a common endpoint (Song & Dhariwal, 2024), while Shortcut Models bootstrap longer transitions from shorter ones (Frans et al., 2025). MeanFlow and improved MeanFlow (iMeanFlow) instead learn average velocities over time intervals (Geng et al., 2026a;b). These approaches retain time- or noise-indexed trajectories while reducing the number of evaluations needed to traverse them.

Drifting Models instead evolve the generator's pushforward distribution during training through an attractive-repulsive field, leaving a single noise-to-sample evaluation at inference (Deng et al., 2026). Kernel-Gradient Drifting (KGD) generalizes this construction by replacing displacement-based directions with kernel gradients (Esteban-Casadevall et al., 2026). Its field equals the difference between the scores of the kernel-smoothed target and generator distributions, extending the connection to smoothed score matching beyond Gaussian kernels. This provides a principled formulation of the field, but still requires a suitable conditional target distribution. The original conditional drifting construction assumes access to multiple positive samples for each condition, as in class-conditional generation. For instance-level cross-modal data, however, each source observation typically has only one observed target, because the paired measurement is taken once per unit: a CITE-seq cell yields a single RNA and a single protein profile before the assay destroys it (Stoeckius et al., 2017; Luecken et al., 2021), a spatial-transcriptomics spot pairs one histology patch with one expression readout (Jaume et al., 2024), and a caption in an image–text corpus is written for one photograph (Lin et al., 2014). The resulting empirical conditional is a point mass, even when the population conditional contains substantial variability. Although a neural generator can share information across conditions, the empirical target supplied to the field still consists of a single observed partner. This motivates borrowing target information from nearby source observations to construct a richer conditional target distribution.

We introduce *Cross-Modal Drifting* (CMD), which trains a generator to sample one modality conditioned on a paired observation of another, producing each sample with a single network evaluation (one NFE). Using fixed modality-specific representations, CMD assigns distance-based weights to nearby source observations and transfers those weights to their paired targets. The resulting weighted empirical distribution serves as a proxy for the unavailable conditional law. This construction assumes that nearby source representations have similar conditional target distributions. Samples from the proxy and the current generator define a kernel-gradient drift (Esteban-Casadevall et al., 2026). We further smooth the proxy using local target-space covariances inspired by Carré du champ constructions (Jones, 2024; Bamberger et al., 2026). The generator learns from this geometry during training and requires neither neighborhood retrieval nor drift computation at inference. Its training target remains a finite-sample approximation to the unknown population conditional.

Our main contributions are (1) a conditional target proxy for paired data that transfers source-neighborhood weights to paired targets, optionally smoothed by local target covariance; (2) a kernel-gradient drifting objective against this proxy that yields a one-NFE conditional generator; and (3) an evaluation on two synthetic tasks with known conditional laws, and two paired biological benchmarks, with component ablations and a cost analysis. CMD achieves the lowest mean Energy Score on both biological tasks and stays strongest as synthetic dimension grows, trading more expensive training for higher-quality one-evaluation sampling.

2 Background

2.1 Drifting models

Let P_ϵ be a simple prior distribution and let G_θ be a generator inducing the pushforward distribution $Q_\theta = (G_\theta)_\# P_\epsilon$ on the sample space $\mathcal{Z}$. Given a target distribution P, Drifting Models (Deng et al., 2026) evolve Q_θ during training using an attractive-repulsive field. For a generated sample z, the original construction defines

$$V_P^+(z) = \frac{\mathbb{E}_{Z^+\sim P}\left[k(z, Z^+)(Z^+ - z)\right]}{\mathbb{E}_{Z^+\sim P}\left[k(z, Z^+)\right]}, \qquad V_{Q_\theta}^-(z) = \frac{\mathbb{E}_{Z^-\sim Q_\theta}\left[k(z, Z^-)(Z^- - z)\right]}{\mathbb{E}_{Z^-\sim Q_\theta}\left[k(z, Z^-)\right]}. \tag{1}$$

The resulting drift field combines these two terms as $V_{P,Q_\theta}(z) = V_P^+(z) - V_{Q_\theta}^-(z)$, where k is a positive similarity kernel. The positive term attracts generated samples toward the target distribution, whereas subtracting the negative term repels them from regions already occupied by the generator. The practical reference baseline additionally uses the affinity and normalization operations described in Appendices A.1 and A.6 of Deng et al. (2026).

Given $z = G_\theta(\epsilon)$, Drifting Models optimize

$$\mathcal{L}_{\text{drift}}(\theta) = \mathbb{E}_{\epsilon \sim P_\epsilon}\left[\left\|z - \text{sg}[z + \eta V_{P,Q_\theta}(z)]\right\|_2^2\right], \tag{2}$$

where $\eta > 0$ controls the drift magnitude and sg denotes the stop-gradient operator (Van Den Oord et al., 2017), used here to freeze the regression target during each update (Chen & He, 2021; Song & Dhariwal, 2024). Although the numerical value of this objective is $\eta^2 \mathbb{E}_{\epsilon \sim P_\epsilon}[\|V_{P,Q_\theta}(G_\theta(\epsilon))\|_2^2]$, the stop-gradient construction ensures that gradients are propagated through the generated sample rather than through the target field. In the conditional setting, the same procedure is applied to $P(\cdot \mid c)$ and $Q_\theta(\cdot \mid c)$ for each condition c, with positive samples drawn from $P(\cdot \mid c)$. The original class-conditional implementation therefore relies on conditions shared by multiple observations.

2.2 Kernel-gradient drifting

Kernel-Gradient Drifting (KGD), introduced by Esteban-Casadevall et al. (2026), replaces the displacement-weighted field above with a field constructed from kernel gradients. Let $k(z, z')$ be a positive normalized kernel whose gradient $\nabla_z k(z, z')$ is well defined. Smoothing the target and generator distributions with k produces the densities

$$p_k(z) = \mathbb{E}_{Z^+ \sim P}\left[k(z, Z^+)\right], \qquad q_{\theta,k}(z) = \mathbb{E}_{Z^- \sim Q_\theta}\left[k(z, Z^-)\right]. \tag{3}$$

Under standard regularity conditions that allow differentiation and expectation to be interchanged, the kernel-gradient field satisfies

$$V^\nabla_{P,Q_\theta}(z) = \frac{\mathbb{E}_{Z^+ \sim P}\left[\nabla_z k(z, Z^+)\right]}{\mathbb{E}_{Z^+ \sim P}\left[k(z, Z^+)\right]} - \frac{\mathbb{E}_{Z^- \sim Q_\theta}\left[\nabla_z k(z, Z^-)\right]}{\mathbb{E}_{Z^- \sim Q_\theta}\left[k(z, Z^-)\right]} = \nabla_z \log p_k(z) - \nabla_z \log q_{\theta,k}(z).$$

Thus, the attraction and repulsion terms are respectively the score functions of the kernel-smoothed target and generator distributions. Kernel-gradient drifting uses the loss in Equation 2, replacing V_{P,Q_θ} with V^∇_{P,Q_θ}.

Esteban-Casadevall et al. (2026) shows that, if k is characteristic, meaning that the smoothing map $P \mapsto p_k$ is injective, then a field that vanishes everywhere identifies the underlying distributions:

$$V^\nabla_{P,Q_\theta}(z) = 0 \quad \text{for all } z \in \mathcal{Z} \quad \Longrightarrow \quad P = Q_\theta. \tag{4}$$

This guarantee, however, is relative to the target distribution supplied to the field: if that target is itself an approximation, a vanishing field implies equality with the approximation, not with the distribution it stands in for.

3 Cross-modal drifting

Cross-Modal Drifting (CMD) extends (kernel-gradient) drifting to paired cross-modal data in which conditioning observations are typically unique. For each source observation, CMD constructs a local empirical approximation to its conditional target distribution and trains a one-pass conditional generator against this proxy.

3.1 Conditional distributions from paired observations

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be paired observations from source and target modalities. Fixed representation maps ϕ_{s} and ϕ_{t} produce

$$u_i = \phi_{\text{s}}(x_i) \in \mathcal{U}, \qquad z_i = \phi_{\text{t}}(y_i) \in \mathcal{Z}.$$

These maps may be pretrained or fitted on the training data, but remain fixed while the generator is trained. The source representation u_i provides the condition, whereas z_i is the target representation to be generated. Let U and Z denote the corresponding population random variables. For a source query u_q, let $P_q = P(Z \mid U = u_q)$ denote the population conditional distribution of the target representation Z. CMD learns a conditional generator $G_\theta(\epsilon, u_q)$, with $\epsilon \sim P_\epsilon$, whose induced distribution is denoted by $Q_{\theta,q}$. The population objective is to make $Q_{\theta,q}$ approximate P_q.

In paired datasets, however, an observed source u_q is often associated with only one target z_q. Without sharing information across conditions, its empirical conditional distribution is therefore the singleton $\widehat{P}_q^{\,\mathrm{sing}} = \delta_{z_q}$, where δ_{z_q} denotes a point mass at z_q. A single paired target is only one realization from P_q and cannot, by itself, characterize its dispersion or possible multimodal structure. To construct a richer target, we assume that nearby source representations have similar conditional target distributions and use their paired targets to form a local empirical proxy for P_q. To this end, let $\mathcal{B}_q$ denote the eligible training reference bank for query q. CMD replaces the singleton with the weighted empirical proxy

$$\widehat{P}_q = \sum_{i \in \mathcal{B}_q} w_{qi}\, \delta_{z_i}, \qquad w_{qi} \geq 0, \qquad \sum_{i \in \mathcal{B}_q} w_{qi} = 1. \tag{5}$$

For a fixed selected breadth M, the weights depend only on distances between the query's source representation and those of the training observations, as described next.

CMD treats $\widehat{P}_q$ as the target distribution of its conditional drift field. We do not assume that $\widehat{P}_q = P_q$ at finite sample size; the quality of this approximation depends on how well the source representation preserves target-relevant structure, how smoothly $P(Z \mid U)$ varies in that representation, and how densely the reference data cover the relevant source space. Accordingly, the identifiability result in Equation 4 concerns $Q_{\theta,q} = \widehat{P}_q$ at a zero-drift equilibrium, rather than equality with the unknown population conditional P_q.

3.2 Source-neighborhood weighting

For each query u_q, CMD retrieves the nearest eligible source representations from the reference bank $\mathcal{B}_q$. Let $\mathcal{M}$ be the candidate grid for the effective-neighborhood-size hyperparameter M, which we call the proxy breadth, and let $M_{\max} = \max \mathcal{M}$. We use a candidate budget $L = \max\{256, 4M_{\max}\}$ and retain the $L_q = \min\{L, |\mathcal{B}_q|\}$ nearest eligible references under the source-space distance $d_{\mathcal{U}}$. Their indices form the candidate set $\mathcal{C}_q \subseteq \mathcal{B}_q$. The truncation to $\mathcal{C}_q$ is computational; the statistical breadth of the proxy is controlled separately by M. Both candidate retrieval and weighting use only source representations. Once M is fixed, we assign sampling probabilities to the training targets based solely on distances between their paired source representations and the query's source representation.

Within $\mathcal{C}_q$, CMD assigns Gaussian distance weights, equivalently applying a softmax to the negative squared source distances,

$$w_{qi}(\tau_q) = \frac{\exp\left(-\dfrac{d_{\mathcal{U}}(u_q, u_i)^2}{2\tau_q^2}\right)}{\displaystyle\sum_{j \in \mathcal{C}_q} \exp\left(-\frac{d_{\mathcal{U}}(u_q, u_j)^2}{2\tau_q^2}\right)}, \qquad i \in \mathcal{C}_q, \tag{6}$$

and sets $w_{qi} = 0$ for every $i \in \mathcal{B}_q$ such that $i \notin \mathcal{C}_q$. Here, $\tau_q > 0$ is a query-specific source-space scale. Smaller values concentrate the weights on the closest references, whereas larger values distribute mass more evenly across the candidate set.

Rather than selecting τ_q as a separate hyperparameter for every query, we quantify the concentration of w_q using the Kish effective sample size (Akinshin, 2023), which we interpret here as the effective neighborhood size:

$$N_{\mathrm{eff}}(w_q) = \frac{1}{\displaystyle\sum_{i \in \mathcal{C}_q} w_{qi}^2}. \tag{7}$$

For a shared candidate value $M \in \mathcal{M}$, we calibrate τ_q independently for each query so that $N_{\text{eff}}(w_q(\tau_q)) = M$, or the closest feasible value when exact equality cannot be attained. Thus, M has a common interpretation across queries, while τ_q adjusts to the query-specific distribution of source distances. Since each candidate source representation u_i is paired with a target representation z_i, the source-derived weight w_{qi} becomes the weight of z_i in the empirical proxy $\widehat{P}_q$ defined in Equation 5.

We select M before training the generator. We first hold out a validation split from the training data; the remainder is the inner-training set, and the held-out pairs are the inner-validation set. For each candidate value, we construct the proxies $\widehat{P}_q^{(M)}$ using only the inner-training reference bank and evaluate them against the paired inner-validation targets using the energy score in the target evaluation space. We select the value with the lowest mean validation score, choosing the smaller M in the event of an exact tie. After selecting the breadth, we freeze the candidate sets and weights built from inner training. The validation observations remain excluded from generator training and the reference bank.

3.3 Conditional drift construction

For each query u_q, let $s = 1, \ldots, S$ index the positive samples and $r = 1, \ldots, R$ index the generated particles. We draw

$$z_{qs}^{+} \overset{\text{iid}}{\sim} \widehat{P}_q, \qquad \widehat{z}_{qr} = G_\theta(\epsilon_{qr}, u_q), \qquad \epsilon_{qr} \overset{\text{iid}}{\sim} P_\epsilon. \tag{8}$$

Thus, $\{z_{qs}^{+}\}_{s=1}^{S}$ approximates $\widehat{P}_q$, while $\{\widehat{z}_{qr}\}_{r=1}^{R}$ approximates $Q_{\theta,q}$. These two query-specific ensembles provide Monte Carlo approximations of the attraction under $\widehat{P}_q$ and the repulsion under $Q_{\theta,q}$. The resulting field is evaluated at each $\widehat{z}_{qr}$, excluding that particle itself from the repulsion estimate. Because the positive samples are already drawn according to w_{qi}, the weights are not applied again. Sampling instead of summing over the complete candidate set bounds the computational cost and also applies to the CDC-smoothed proxy introduced below.

We use a fixed-bandwidth Matérn-$3/2$ kernel (Williams & Rasmussen, 2006). Let $d(z, z')$ denote the Euclidean distance between two target representations. The kernel is

$$k_h(z, z') = \left(1 + \frac{\sqrt{3}\, d(z, z')}{h}\right) \exp\left(-\frac{\sqrt{3}\, d(z, z')}{h}\right), \tag{9}$$

where h is calibrated once from inner-training target distances (Appendix B.2) and held fixed across all KGD-based variants. This is the conventional unit-amplitude parameterization, for which $k_h(z, z) = 1$. The unit-integral rescaling used in the kernel-smoothed density interpretation is omitted because its multiplicative factor cancels from each normalized kernel-gradient ratio and therefore leaves the drift field unchanged. The rescaling and this cancellation are derived in Appendix A.2.

Let $\widehat{p}_{q,h}$ and $q_{\theta,q,h}$ denote the kernel-smoothed densities of $\widehat{P}_q$ and $Q_{\theta,q}$. The corresponding population field satisfies

$$V_{q,h}^{\nabla}(z) = \nabla_z \log \widehat{p}_{q,h}(z) - \nabla_z \log q_{\theta,q,h}(z) = \nabla_z \log \frac{\widehat{p}_{q,h}(z)}{q_{\theta,q,h}(z)}.$$

For the fixed-bandwidth Matérn kernel, an exact field that vanishes everywhere identifies equality between $Q_{\theta,q}$ and the supplied proxy $\widehat{P}_q$, under the conditions stated in Appendix A.3.

As an optional geometry-aware regularization, CMD replaces the discrete positive-sample distribution $\widehat{P}_q$ with a geometry-aware smoothing inspired by CDC-FM. Using only the training-target representations, we estimate a positive semidefinite local covariance $\Gamma_i^{\star}$ (see Appendix A.4) around each z_i and define

$$\widehat{P}_q^{\text{CDC}} = \sum_{i \in \mathcal{C}_q} w_{qi}\, \mathcal{N}(z_i, \Gamma_i^{\star})\,.$$

This construction preserves the source-derived mixture weights w_{qi}, while replacing each point mass at z_i with a possibly degenerate Gaussian component aligned with the local target-space geometry. Positive samples are obtained by first selecting i with probability w_{qi} and then drawing

$z^+ \sim \mathcal{N}(z_i, \Gamma_i^\star)$. When $\Gamma_i^\star = 0$ for every i, the construction reduces to the original discrete proxy. The covariances describe local target-space geometry and are not assumed to equal the conditional covariance of P_q. Their estimation, maximum supported rank, and scaling are detailed in Appendix A.4. The variable-bandwidth Gaussian kernel of Appendix A.4 is used only to estimate these covariances; the drifting field continues to use the fixed-bandwidth Matérn-$3/2$ kernel.

CDC smoothing changes the positive distribution but not the kernel-gradient operation. For a sampled positive z^+, the field differentiates $k_h(z, z^+)$ with respect to the generated evaluation point z, while treating z^+ and the precomputed covariance $\Gamma_i^\star$ as fixed. Defining the kernel smoother

$$\widehat{p}_{q,h}^{\mathrm{CDC}}(z) = \mathbb{E}_{Z^+ \sim \widehat{P}_q^{\mathrm{CDC}}} \left[k_h(z, Z^+) \right],$$

the population field satisfies

$$V_{q,h}^{\nabla,\mathrm{CDC}}(z) = \nabla_z \log \widehat{p}_{q,h}^{\mathrm{CDC}}(z) - \nabla_z \log q_{\theta,q,h}(z).$$

Appendix A.3 extends KGD identifiability to distributions without densities. It follows that a CDC field vanishing everywhere implies $Q_{\theta,q} = \widehat{P}_q^{\mathrm{CDC}}$.

CMD optimizes the drifting loss from Section 2 using the estimated conditional field. We calibrate the drift multiplier η once per run, using the bandwidth and initial field magnitudes on inner-training observations, and keep it fixed throughout optimization (Appendix B.2). At inference, neither the proxy nor the drift field is required: a conditional sample is generated in one neural function evaluation as $\widehat{z} = G_\theta(\epsilon, u)$, followed, when needed, by the fixed inverse representation map or target decoder.

4 Experiments

Our experiments address how the field construction, the source-neighborhood proxy, and the target covariance affect conditional generation; how CMD compares with existing generative objectives; and what its preparation, training, and inference cost relative to them. Component ablations use the conditional-circle task and BMMC RNA-to-protein generation. The comparison with existing objectives adds sketch-derived conditional generation and HEST-IDC.

4.1 Experimental setup

We evaluate two synthetic tasks with known conditional laws, BMMC CITE-seq RNA-to-protein generation (Luecken et al., 2021), and HEST-IDC histology-to-gene-expression generation (Jaume et al., 2024). Within each dataset, methods share the backbone, conditioning, and validation protocol and differ only in their objectives. Generator hyperparameters are fixed or set heuristically. Each seed or fold trains once, and validation Energy Score (ES) selects the exponential-moving-average (EMA) checkpoint without refitting.

ES is our primary metric: generative models aim to recover conditional distributions, and predictive-mean metrics cannot separate distributions that share a mean but differ in spread. ES is strictly proper on distributions with finite first moment, so its expectation is minimized only by the true conditional (Gneiting & Raftery, 2007), and it is estimable from generated samples and a single observed target per query. On the biological tasks we also report Pearson correlation of the predictive mean. The synthetic tasks provide multiple oracle targets per condition, which we use to estimate the conditional Wasserstein-1 distance (Peyré & Cuturi, 2019) and to compare single-observation with oracle-averaged ES. Appendix B provides the protocols.

4.2 Ablation study

We compare Reference Drifting (A), KGD (B), and KGD augmented with CDC smoothing (C), the source-neighborhood proxy (D), or both (E). Table 1 reports the circle and BMMC ablations. Appendix B.7 provides the circle construction and protocol.

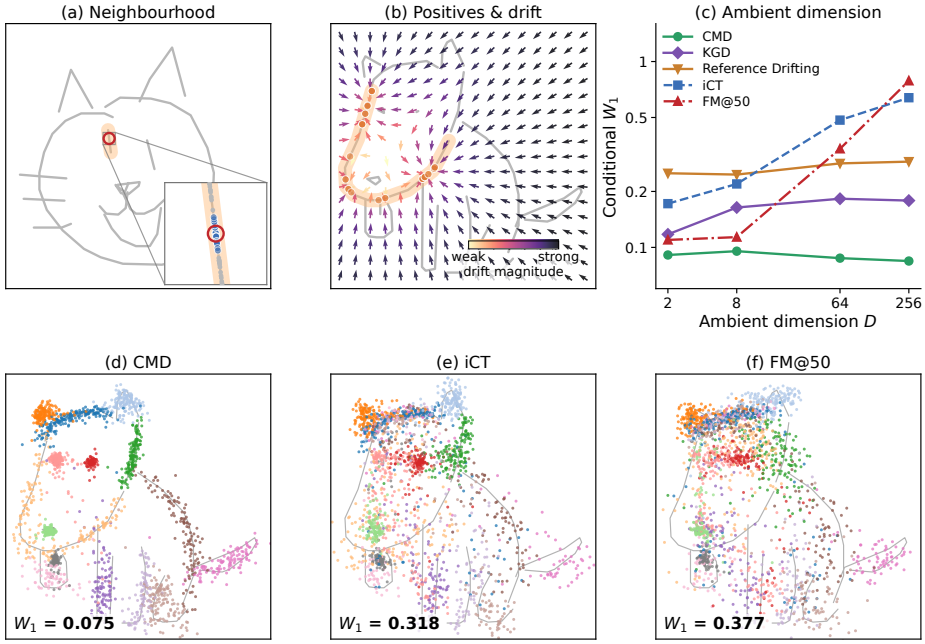

Figure 2: Cat→dog stroke transport on two Quick, Draw! sketches (Google Creative Lab): a point on cat stroke k is paired with a uniform draw along dog stroke k (Appendix B.8). **(a)** A query (ring) on the highlighted stroke and its M=16 retrieved training sources (inset, blue). **(b)** Their targets, the proxy positives (orange), cover the matching dog stroke; arrows show the drift direction they induce, coloured by magnitude. **(c)** Conditional W_1 as the planar task is embedded isometrically in D dimensions (mean of three seeds; solid: drifting methods). **(d)–(f)** Test samples at D=64 (one seed), projected to the plane and coloured by source stroke (Fig. 4). FM@50 uses 50 steps.

	(a) Synthetic circle ($D = 2$)		(b) BMMC CITE-seq	
Method	$W_1 \downarrow$	$\mathrm{ES}_{\mathrm{single}} \downarrow$	ES↓	Pearson↑
A: Reference Drifting	0.217 ± 0.041	0.405 ± 0.011	8.300 ± 0.031	0.871 ± 0.001
B: KGD	0.490 ± 0.001	0.463 ± 0.006	6.855 ± 0.018	0.882 ± 0.000
C: KGD + CDC	0.491 ± 0.001	0.463 ± 0.007	6.789 ± 0.028	$\mathbf{0.882 \pm 0.001}$
D: KGD + proxy	0.187 ± 0.013	$\mathbf{0.393 \pm 0.006}$	6.612 ± 0.009	0.877 ± 0.000
E: KGD + proxy + CDC	$\mathbf{0.186 \pm 0.008}$	0.394 ± 0.006	$\mathbf{6.547 \pm 0.009}$	0.877 ± 0.000

Table 1: Component ablation: (a) synthetic circle with source and target dimension $D = 2$ and 512 training pairs per replicate; (b) BMMC CITE-seq. Test performance is mean ± sample standard deviation across three replicates; bold marks the best unrounded mean within each column, without a significance claim. Synthetic W_1 uses an independent oracle ensemble, while ES uses one observed target per condition. Evaluation and synthetic protocols appear in Appendices B.3 and B.7.

Circle conditional recovery. Each source lies on the unit circle, and its target is obtained by an angular shift sampled uniformly from either $[-1, -0.5]$ or $[0.5, 1]$ radians, choosing the interval with equal probability (Figure 3). Training and validation reveal one target per source, whereas repeated independent draws at held-out sources enable oracle evaluation of the conditional distribution. Appendix B.7 provides the full construction.

Why single-observation ES. In BMMC, each source is paired with only one observed target, so the true conditional distribution cannot be represented empirically at that source. A Wasserstein distance computed against this single observation would measure distance to the point mass $\widehat{P}_q^{\mathrm{sing}} = \delta_{z_q}$ rather than to the unknown conditional law P_q. Even after averaging across test pairs, this objective can be minimized by collapsing the prediction onto a conditional geometric median

and therefore does not identify the full conditional distribution. ES remains estimable from samples of $Q_{\theta,q}$ and the one observed target z_q because it combines sample-to-target distances with a sample-to-sample term that accounts for predictive dispersion. Its expected value is uniquely minimized by the true conditional distribution, making it a strictly proper distributional score (Gneiting & Raftery, 2007).

The circle experiment provides a controlled validation of this evaluation strategy. In panel (a), oracle Wasserstein distance and single-observation ES both place D and E ahead of A–C, differing only in their ordering of D and E. Across the higher-dimensional experiments, Wasserstein distance, single-observation ES, and oracle-averaged ES again identify D and E as the leading variants. Thus, the substantive component conclusion is stable across the three criteria, even though their finite-sample rankings need not coincide exactly.

BMMC. KGD reduces ES by 17.4% relative to Reference Drifting. Adding the source-neighborhood proxy yields a further 3.5-3.6% reduction, with or without CDC, while CDC contributes approximately 1% in either setting. Combining the proxy and CDC produces the lowest ES, 21.1% below Reference Drifting and 4.5% below KGD. While Pearson correlation, the supporting metric that sees the predictive mean but not the conditional spread (Appendix B.3), ranks B and C highest, with E still above Reference Drifting.

4.3 Main results

		BMMC CITE-seq		HEST-IDC	
Method	NFE	ES↓	Pearson↑	ES↓	Pearson↑
FM	50	6.965 ± 0.043	0.882 ± 0.000	6.768 ± 1.427	0.532 ± 0.099
CDC-FM	50	6.938 ± 0.124	0.870 ± 0.010	6.528 ± 1.468	0.512 ± 0.084
MeanFlow	1	7.172 ± 0.049	0.876 ± 0.002	7.120 ± 1.643	$\mathbf{0.546 \pm 0.093}$
iMeanFlow	1	7.128 ± 0.038	0.878 ± 0.001	6.956 ± 1.598	0.544 ± 0.095
Shortcut	1	7.020 ± 0.014	0.880 ± 0.001	6.354 ± 1.304	0.538 ± 0.100
iCT	1	7.017 ± 0.038	0.881 ± 0.002	6.781 ± 1.571	0.537 ± 0.100
Reference Drifting (A)	1	8.300 ± 0.031	0.871 ± 0.001	6.870 ± 1.073	0.508 ± 0.094
KGD (B)	1	6.855 ± 0.018	$\mathbf{0.882 \pm 0.000}$	6.620 ± 1.223	0.506 ± 0.103
CMD (E)	1	$\mathbf{6.547 \pm 0.009}$	0.877 ± 0.000	$\mathbf{6.237 \pm 1.142}$	0.514 ± 0.098

Table 2: Biological benchmarks: mean ± SD over three training seeds (BMMC) or the four official outer folds (HEST-IDC; one run per method and fold). Bold: best unrounded mean per column, without a significance claim. BMMC rows A/B/E reuse Table 1(b); Appendices B.5 and B.6 give full flow-budget sweeps and per-run results. ES is computed in each benchmark's reporting space and is not comparable across datasets.

Following the component study, we compare CMD with MeanFlow, iMeanFlow, Shortcut, iCT, FM and CDC-FM, retaining reference drifting and KGD as controls. FM and CDC-FM are evaluated at 1, 20, and 50 neural function evaluations (NFE), reusing each trained checkpoint across these inference settings. The other methods use one NFE.

Synthetic dimension sweeps. Both synthetic tasks embed a planar process in D-dimensional space through fixed orthonormal maps, so distances and the conditional structure are identical at every D (Appendices B.7 and B.8). On the circle ($D \in \{64, \ldots, 512\}$, 512 training pairs), the proxy variants D and E are the two strongest learned methods under conditional W_1 and both ES variants at every dimension. At D=512, CMD reaches $W_1 = 0.346 \pm 0.051$ against 0.515 ± 0.020 for iCT, the strongest competing objective, while FM at 50 NFE rises from 0.633 at D=64 to 19.625, and CDC-FM reproduces FM wherever it completed ($D \leq 256$; Appendix B.7). This matches the difficulty of predicting high-dimensional noise or velocity with a fixed-width network noted by Li & He (2026), and establishes the robustness of proxy-based drifting under our fixed architecture and budget. On the sketch task (Figure 2, where a cat stroke j maps to a uniform point along dog stroke j, with 15 strokes and 5,000 pairs), CMD has the lowest mean W_1 at every $D \in \{2, 8, 64, 256\}$ with one generator evaluation, staying between 0.084 and 0.095 while FM at 50 NFE degrades from

0.110 to 0.788. Oracle-averaged ES agrees except at $D{=}8$, where FM leads. Figure 1 shows CMD samples converging onto the target strokes over training at $D{=}2$.

BMMC and HEST. Table 2 compares nine methods on BMMC and HEST-IDC. The complete experimental protocols are detailed in Appendices B.5 and B.6. CMD achieves the lowest mean ES on both benchmarks. On BMMC, its ES is 5.6% below CDC-FM and 6.0% below FM at 50 NFE, and is lower than every competing method at every inference budget, for each corresponding seed. On HEST, its ES is 1.8% below Shortcut, 5.8% below KGD, and 4.5% below CDC-FM at 50 NFE. It improves on KGD and Reference Drifting in every HEST fold and on Shortcut in three of four folds, while CDC-FM is best on the remaining fold. Although increasing the flow inference budget improves ES on both datasets, neither flow baseline reaches CMD at 50 NFE. CDC-FM's wider seed spread on BMMC reflects runs that early-stopped at very different training lengths (Appendix B.5). In contrast, the Pearson ordering differs. KGD has the highest mean correlation on BMMC (0.8819 ± 0.0004), whereas MeanFlow leads on HEST (0.5459 ± 0.0929), with CMD obtaining 0.8770 ± 0.0002 and 0.5143 ± 0.0979, respectively. The distributional-score gains therefore do not uniformly translate into better predictive means.

4.4 Computation cost

All costs are reported under one common metric: end-to-end training wall time divided by optimizer updates (Appendix B.4). CMD posts the largest training total on HEST-IDC and, after CDC-FM, on BMMC. On BMMC it takes 28.6 minutes against 3.3–23.3 for every other method but CDC-FM (69.3), because the proxy positives make its update the most expensive of the drift family (8.6 vs. 1.0–7.0 ms). On HEST-IDC it takes 502 minutes against 13–213, but there the update itself is unremarkable (470 ms, between KGD's 467 and CDC-FM's 544) and the total comes from a $\sim 3\times$ longer optimization trajectory. Where CDC's gain is small, KGD-proxy (D) is the cheaper choice: dropping CDC cuts the update from 8.6 to 7.0 ms on BMMC at a cost of 0.065 ES (6.547 vs. 6.612), and from 12.0 to 5.6 ms on the 2-D circle at no cost (W_1 0.187 vs. 0.186). Another option is to activate CDC only in the later stages of training, so that fewer updates carry the covariance-based perturbation. Whether this preserves the benefit of smoothing at a lower total cost remains to be investigated.

Despite this higher training cost, CMD samples with a single generator evaluation and no auxiliary time or step-size input: $50\times$ fewer function evaluations than FM and CDC-FM at their strongest reported setting, with measured sampling time following suit (50 steps cost $46.4\times$ and $49.9\times$ one step of the same network on the circle and sketch). Among one-NFE methods, cost does not separate the panel: the competitors' auxiliary time or step-size inputs leave network size essentially unchanged (Appendix B.4), so the comparison among them rests on sample quality.

5 Conclusion

Cross-Modal Drifting transfers source-neighborhood weights to paired targets and trains a one-evaluation generator through kernel-gradient matching and local covariance smoothing. CMD performs strongly on both circle and sketch-derived conditional generation, including their ambient-dimensional extensions, and achieves the lowest mean Energy Score on BMMC and HEST-IDC. The component studies support constructing a conditional proxy from nearby paired observations, while the additional benefit of covariance smoothing varies across settings. These distributional gains come with higher training cost and do not consistently extend to predictive-mean accuracy. At inference, the trained generator requires only one network evaluation, without neighborhood retrieval or drift computation.

CMD relies on source representations whose neighborhoods preserve conditional similarity and target representations whose geometry provides useful guidance for generation. Understanding when these assumptions hold remains an important limitation and direction for further study. Learning or adapting the representations jointly with the generator, and developing diagnostics of proxy quality before expensive training, are promising next steps.

AI USAGE STATEMENT

In this work, we used generative AI tools to help write and debug code for our methods, synthetic data generation, and model training and evaluation. We reviewed the generated code, revising it where necessary before incorporating it into our work. Our implementation checks included code review, automated regression tests, and numerical consistency checks. We also provide a VS Code debugger profile that accepts a training configuration file as its starting point for step-by-step inspection of the execution flow. Additionally, we used generative AI for literature retrieval and discovery, drafting manuscript sections, improving readability, organizing the paper, and formatting references and LaTeX.

We executed the final experimental runs and handled dataset acquisition and preprocessing. We made the final methodological and reporting decisions and take responsibility for all content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Andrey Akinshin. Weighted quantile estimators. *arXiv preprint arXiv:2304.07265*, 2023.

Jacob Bamberger, Iolo Jones, Dennis Duncan, Michael Bronstein, Pierre Vandergheynst, and Adam Gosztolai. Carré du champ flow matching: better quality-generalisation tradeoff in generative models. In *International Conference on Learning Representations*, volume 2026, pp. 135754–135779, 2026.

Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Andrew H Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature medicine*, 30(3):850–862, 2024.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15745–15753. IEEE, 2021.

Zonghao Chen, Toni Karvonen, Heishiro Kanagawa, François-Xavier Briol, Chris Oates, et al. Stationary mmd points. *arXiv preprint arXiv:2505.20754*, 2025.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

Maria Esteban-Casadevall, Jorge Carrasco-Pollo, Max Welling, Jan-Willem van de Meent, Erik J Bekkers, and Floor Eijkelboom. Kernel-gradient drifting models. *arXiv preprint arXiv:2605.10727*, 2026.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, volume 2025, pp. 34668–34684, 2025.

Alan E Gelfand, Peter Diggle, Peter Guttorp, and Montserrat Fuentes. *Handbook of spatial statistics*. CRC press, 2010.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 38:75460–75482, 2026a.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30467–30476, 2026b.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Google Creative Lab. The Quick, Draw! dataset. `https://github.com/googlecreativelab/quickdraw-dataset`. Accessed September 24, 2026.

Yuhan Hao, Tim Stuart, Madeline H Kowalski, Saket Choudhary, Paul Hoffman, Austin Hartman, Avi Srivastava, Gesmira Molla, Shaista Madad, Carlos Fernandez-Granda, et al. Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature biotechnology*, 42(2): 293–304, 2024.

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

Guillaume Jaume, Paul Doucet, Andrew H Song, Ming Y Lu, Cristina Almagro-Pérez, Sophia J Wagner, Anurag J Vaidya, Richard J Chen, Drew F Williamson, Ahrong Kim, et al. Hest-1k: A dataset for spatial transcriptomics and histology image analysis. *Advances in Neural Information Processing Systems*, 37:53798–53833, 2024.

Iolo Jones. Diffusion geometry. *arXiv preprint arXiv:2405.10858*, 2024.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Malte D Luecken, Daniel Bernard Burkhardt, Robrecht Cannoodt, Christopher Lance, Aditi Agrawal, Hananeh Aliee, Ann T Chen, Louise Deconinck, Angela M Detweiler, Alejandro A Granados, et al. A sandbox for prediction and integration of dna, rna, and proteins in single cells. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (Round 2)*, 2021.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Gabriel Peyré and Marco Cuturi. Computational optimal transport with applications to data sciences. *Foundations and trends® in machine learning*, 11(5-6):355–607, 2019.

Emilio Porcu, Moreno Bevilacqua, Robert Schaback, and Chris J Oates. The matérn model: A journey through statistics, numerical analysis and machine learning. *Statistical Science*, 39(3): 469–492, 2024.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, volume 2024, pp. 15078–15097, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 11:1517–1561, 2010.

Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K Chattopadhyay, Harold Swerdlow, Rahul Satija, and Peter Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nature methods*, 14(9):865–868, 2017.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

A METHODOLOGICAL DETAILS

A.1 SOURCE-NEIGHBORHOOD CALIBRATION

The eligible reference bank $\mathcal{B}_q$ contains inner-training observations only. A training query excludes its own paired row. Validation and test queries retrieve exclusively from the inner-training bank.

During proxy-breadth selection, all candidates in the declared feasible grid $\mathcal{M}$ share the source support budget $L = \max\{256, 4\max(\mathcal{M})\}$, capped by the eligible bank size. For each query, the candidate set contains its L_q nearest eligible source representations.

For each candidate M, we calibrate the query-specific scale τ_q to match the effective neighborhood size $N_{\text{eff}}(w_q(\tau_q))$ to M. This quantity is non-decreasing in τ_q, tending to the number of candidates tied at the minimum distance as $\tau_q \to 0$ and to L_q as $\tau_q \to \infty$. We perform 60 bisection iterations in log-temperature over $[10^{-3}s_q, 10^3 s_q]$, where s_q is the median candidate distance, floored at 10^{-12}. We retain the resulting weights and record the calibration as clipped when the realized effective neighborhood size differs from M by more than 1%.

We select M by evaluating each weighted empirical proxy, before CDC smoothing, against the paired inner-validation target using the energy score in the dataset-specific validation space. Scores are averaged within donors and then equally across donors for BMMC, within slides and then equally across slides for HEST, and across queries otherwise. We choose the candidate with the lowest validation score, breaking exact ties in favor of the smaller value.

After selection, the breadth and inner-training bank remain fixed. BMMC and the synthetic experiments retain the common selection support budget; HEST constructs its training proxies using $L = \max\{256, 4M\}$, capped by the eligible bank size. The resulting training candidate sets and weights remain fixed throughout optimization. The candidate grid, selected breadth, support budget, and proxy-score curve are recorded for each dataset or outer fold. Validation and test targets never enter the reference bank or its fitted geometry.

A.2 KERNEL NORMALIZATION

CMD uses the standard unit-amplitude Matérn-$3/2$ kernel

$$k_h(z, z') = \left(1 + \frac{\sqrt{3}\|z - z'\|_2}{h}\right) \exp\left(-\frac{\sqrt{3}\|z - z'\|_2}{h}\right), \tag{10}$$

where $h > 0$ is the fixed target-space bandwidth. This is the usual covariance-kernel convention, under which $k_h(z, z) = 1$. This convention fixes the kernel's peak value at one but does not impose a unit integral. Consequently, its averages over a probability distribution require normalization to define a proper probability density. As shown below, this normalization leaves CMD's normalized kernel-gradient field unchanged.

For every fixed $h > 0$, the unit-amplitude Matérn-$3/2$ kernel is strictly positive and decreases from one with distance (Gelfand et al., 2010, p. 26), and it decays exponentially with distance (Porcu et al., 2024, Section 5.1), ensuring integrability in every finite dimension. Because k_h depends on (z, z') only through $z - z'$, the substitution $u = z - z'$ gives

$$m_{d_z,h} = \int_{\mathbb{R}^{d_z}} k_h(z, z')\,\mathrm{d}z = \int_{\mathbb{R}^{d_z}} \Big(1 + \frac{\sqrt{3}\|u\|_2}{h}\Big) \exp\Big(-\frac{\sqrt{3}\|u\|_2}{h}\Big)\mathrm{d}u \in (0, \infty), \tag{11}$$

a constant that does not depend on z'. A unit-integral version is therefore

$$\bar{k}_h(z, z') = \frac{k_h(z, z')}{m_{d_z,h}}, \tag{12}$$

so that $z \mapsto \mathbb{E}_{Z' \sim P}[\bar{k}_h(z, Z')]$ integrates to one for every probability distribution P.

This rescaling does not affect CMD's normalized kernel-gradient field. For any probability distribution P,

$$\frac{\mathbb{E}_{Z'\sim P}\left[\nabla_z \bar{k}_h(z,Z')\right]}{\mathbb{E}_{Z'\sim P}\left[\bar{k}_h(z,Z')\right]} = \frac{\mathbb{E}_{Z'\sim P}\left[m_{d_z,h}^{-1}\nabla_z k_h(z,Z')\right]}{\mathbb{E}_{Z'\sim P}\left[m_{d_z,h}^{-1}k_h(z,Z')\right]} = \frac{m_{d_z,h}^{-1}\mathbb{E}_{Z'\sim P}\left[\nabla_z k_h(z,Z')\right]}{m_{d_z,h}^{-1}\mathbb{E}_{Z'\sim P}\left[k_h(z,Z')\right]} = \frac{\mathbb{E}_{Z'\sim P}\left[\nabla_z k_h(z,Z')\right]}{\mathbb{E}_{Z'\sim P}\left[k_h(z,Z')\right]}. \tag{13}$$

The same holds for the target and generator terms, so the resulting drift is identical.

Accordingly, CMD implements the standard unit-amplitude Matérn kernel. Whenever its kernel averages are interpreted as smoothed probability densities, the unit-integral rescaling above is understood implicitly. Its normalization constant need not be evaluated.

A.3 KGD IDENTIFIABILITY BEYOND DISTRIBUTIONS WITH DENSITIES

The score identity and identifiability result in Propositions 3.1 and 3.3 of Esteban-Casadevall et al. (2026) are stated for distributions possessing densities, whereas CMD's targets are a discrete proxy $\widehat{P}_q$ or, with CDC smoothing, a mixture of Gaussians whose covariances may be rank deficient; neither has a density. Both ingredients of the KGD argument nonetheless remain available for such targets under the fixed-bandwidth Matérn-3/2 kernel. First, its kernel smoother $p_h(z) = \mathbb{E}_{Z'\sim P}[k_h(z,Z')]$ is strictly positive (Appendix A.2), and the Matérn kernel satisfies the integrability and differentiability conditions under which the gradient of a kernel mean can be taken inside the expectation (Chen et al., 2025, Assumption 1 and Lemma B.1), so the kernel-gradient field equals the score difference $\nabla_z \log p_h - \nabla_z \log q_h$ for arbitrary Borel probability measures P and Q. Second, the Matérn kernel is characteristic (Sriperumbudur et al., 2010, Section 3.2), so equality of the smoothers implies $P = Q$ (Esteban-Casadevall et al., 2026, Proposition 3.3). Consequently, a CDC field that vanishes everywhere implies $Q_{\theta,q} = \widehat{P}_q^{\mathrm{CDC}}$. We defer a self-contained proof to a longer version of this work.

This is an identifiability statement for the exact distribution-level field. It does not guarantee convergence of generator training or equality for a finite Monte Carlo estimate of the field.

A.4 CDC COVARIANCE ESTIMATION

We estimate local covariance in the drifting-feature space using only inner-training targets, following the variable-bandwidth construction of Bamberger et al. (2026). For each target z_i, the neighborhood $\mathcal{N}_i$ contains z_i and up to $k_{\mathrm{cdc}} - 1$ nearest training targets whose representations differ from z_i. The Gaussian bandwidth at each target is its distance to the k_{bw}-th nearest such neighbor. We normalize the Gaussian weights within each neighborhood to obtain transition probabilities P_{ij}, following Bamberger et al. (2026, Eq. (11)). We compute

$$m_i = \sum_{j\in\mathcal{N}_i} P_{ij} z_j, \qquad \Gamma_i = \sum_{j\in\mathcal{N}_i} P_{ij}(z_j - m_i)(z_j - m_i)^\top.$$

We retain the full supported covariance rank, at most $\min\{d_z, |\mathcal{N}_i| - 1\}$, without additional eigenvalue thresholding. To limit the perturbation magnitude, let r_i denote the distance from z_i to its nearest training target, excluding itself and any identical target feature vectors, and let $b_{0.9}$ be the 90th percentile of $r_i^2/9$ over inner-training. For $\lambda_{\max}(\Gamma_i) > 0$, we use

$$c_i = \min\left\{1,\ \frac{\min\{r_i^2/9, b_{0.9}\}}{\lambda_{\max}(\Gamma_i)}\right\}, \qquad \Gamma_i^\star = \gamma\, c_i\, \Gamma_i, \quad \gamma = 0.5,$$

where $\lambda_{\max}(\Gamma_i)$ denotes the largest eigenvalue of the local covariance matrix Γ_i. This adapts the nearest-neighbor scaling heuristic of Bamberger et al. (2026, Eq. 33). The local variance bound $r_i^2/9$ follows a three-standard-deviation rule: limiting the largest directional standard deviation to $r_i/3$ places three standard deviations within the nearest-neighbor distance. This is a directional heuristic, not a guarantee that the full multidimensional perturbation remains within that radius. The global cap $b_{0.9}$ prevents unusually isolated targets from receiving excessive noise. Because it is a

percentile of squared distances divided by nine, it already has variance units and is applied without further squaring.

The outer minimum with one restricts c_i to downscaling: we reduce Γ_i when its largest eigenvalue exceeds $r_i^2/9$ or $b_{0.9}$. If instead $\lambda_{\max}(\Gamma_i) \le \min\{r_i^2/9, b_{0.9}\}$, we set $c_i = 1$, leaving the covariance unchanged before applying the global factor γ. Thus, the local rescaling limits the Gaussian spread: it can make the Gaussian narrower, but never wider. For illustration, suppose

$$\lambda_{\max}(\Gamma_i) = 0.01, \qquad \min\{r_i^2/9, b_{0.9}\} = 0.04.$$

Without the outer minimum with one, the scaling factor would be $c_i = 0.04/0.01 = 4$. The local rescaling $\Gamma_i \mapsto 4\Gamma_i$ would therefore multiply each variance by four and each standard deviation by two, increasing the Gaussian spread. Our rule instead gives $c_i = \min\{1, 4\} = 1$, leaving this already-small covariance unchanged at the local rescaling step. The global factor γ is applied afterwards.

CDC-FM uses local covariance in a time-dependent flow-matching path, whereas CMD uses $\mathcal{N}(z_i, \Gamma_i^\star)$ as a Gaussian component of its conditional proxy. The source-derived weights w_{qi} determine the component probabilities and do not enter the covariance estimation.

B Experimental details

B.1 Datasets and representations

BMMC. We use the BMMC CITE-seq data and predefined competition split for RNA-to-protein generation Luecken et al. (2021). Inner validation consists of 2,000 randomly drawn cells from the outer-training pool, shared across all methods and training seeds. Feature selection and the transformations used for training and evaluation are fitted on the remaining inner-training cells.

For cell i and gene g, RNA counts c_{ig} are library-size normalized and log transformed before feature selection:

$$\widetilde{c}_{ig} = \log\left(1 + 10^4 \frac{c_{ig}}{\sum_{g'} c_{ig'}}\right).$$

This follows Seurat's LogNormalize convention Hao et al. (2024). The library-size denominator uses the cell's complete input gene panel before masking. We remove zero-variance features and retain the 1,000 genes with the largest inner-training empirical variance.

For cell i and protein p, ADT counts a_{ip} undergo per-cell, Seurat-style CLR normalization before feature masking:

$$g_i = \exp\left(\frac{1}{P}\sum_{p=1}^{P} \log(1 + a_{ip})\right), \qquad \widetilde{a}_{ip} = \log(1 + a_{ip}/g_i),$$

where P is the number of proteins in the full input panel. Zero-count entries contribute zero to the sum but remain included in P. This follows the per-cell CLR convention implemented in Seurat v5.3.0[1], rather than subtraction of the mean log count. Both RNA and ADT normalizations operate on each observation's own counts. Subsequent feature masks are fitted only on inner training; the ADT mask removes zero-variance proteins.

Separate 50-component PCA transforms with whitening are fitted to the RNA and ADT representations using scikit-learn Pedregosa et al. (2011), using inner-training observations only. An additional per-coordinate standardizer is fitted to the target PCA coordinates. All methods use these same source and target representations. Generator outputs are mapped back to the normalized ADT panel by undoing the additional standardization and applying the inverse target PCA transform. Because target PCA reduces dimensionality, decoded predictions lie in the retained PCA affine subspace; the inverse map is a reconstruction rather than an exact recovery of the original protein vector.

For ES, decoded samples and actual held-out normalized ADT observations are mapped by the same per-protein scoring standardizer, fitted on the inner-training normalized ADT panel. This

[1] https://github.com/satijalab/seurat/blob/v5.3.0/R/preprocessing.R

scoring standardizer is separate from the standardizer applied to target PCA coordinates during training. Pearson correlation is computed across proteins for each cell, comparing the predictive mean with the observed normalized ADT vector. Held-out references are not replaced by their PCA reconstructions.

HEST-IDC. We use the official four-fold IDC[2] evaluation protocol Jaume et al. (2024), training each method once per fold with one fixed training seed shared across methods and folds. The held-out slides are TENX99 (fold 0), TENX95 (fold 1), NCBI785 (fold 2), and NCBI783 (fold 3). Within each outer-training fold, a fixed random subset of spots is reserved for validation and shared across methods. Targets are log-transformed expression values, $\log(1 + \text{count})$, for the official 50-gene panel. Source representations are histology-patch embeddings extracted using the pretrained UNI2-h encoder Chen et al. (2024), whose weights remain frozen throughout our experiments.

For each fold, we fit full-rank PCA whitening to the inner-training gene-expression vectors using scikit-learn Pedregosa et al. (2011), retaining all 50 coordinates. We then fit a per-coordinate standardizer to the whitened training coordinates. This follows the same transformation sequence as BMMC, but without target dimensionality reduction. Both transforms remain frozen during training and are shared across methods. We retain the official gene panel without additional gene selection.

Similarly to BMMC, generated samples are decoded by undoing the coordinate standardization and applying the inverse PCA transform. Because all 50 directions are retained, this mapping is invertible up to numerical precision. For validation and test ES, decoded samples and observed log-transformed expression vectors are standardized using a separate per-gene scoring standardizer fitted on inner training. Pearson correlation uses the decoded predictive mean on the official log-transformed expression scale. Both metrics compare against actual held-out observations, and all methods use the same reporting maps.

B.2 Training and model selection

Generator hyperparameters are fixed in advance or calibrated using inner-training data. Each method has one training trajectory per seed or outer fold. BMMC uses seeds 42, 43, and 44. HEST uses seed 42 for each of its four official outer folds. The circle and sketch studies each use three dataset/training replicates, paired across methods and dimensions, as described in Appendices B.7 and B.8.

Validation observations are excluded from optimizer batches, the proxy's reference bank, covariance estimation, and fitted transformations. Validation queries and sampling seeds are fixed across methods and training seeds within each data partition. The checkpoint with the lowest validation ES is retained using exponential-moving-average (EMA) weights, with no subsequent training on validation observations.

For BMMC and HEST, we use Adam with an initial learning rate of 10^{-4}, zero weight decay, gradient-norm clipping at 1, and EMA decay 0.999. The learning rate follows a cosine schedule without restarts toward 10^{-6}, with its horizon equal to the maximum update count. Validation occurs every 2,000 updates. Validation stops after ten consecutive checks without a strictly lower validation ES, and training at the maximum update count or after 72 hours of cumulative wall-clock time, whichever occurs first; on HEST-IDC and in the CDC-FM BMMC reruns, training halts with validation.

BMMC uses an MLP with hidden width 256 and four layers, with a 32-dimensional time embedding where required. The drifting generator takes eight-dimensional Gaussian noise. HEST uses a gene-token DiT with width 256, six blocks, eight attention heads, and MLP ratio 4. The HEST drifting generator takes 50-dimensional Gaussian noise. Within each dataset, backbone widths and depths are held fixed across methods, with input, output, and time-conditioning components adapted to each method.

Proxy and field calibration. Proxy breadth is selected before generator training from $\{4, 8, 16, 32, 64, 128\}$, using the inner-validation procedure in Appendix A.1. The selected breadth

[2] https://hest.readthedocs.io/en/latest/tutorials/4-Running-HEST-Benchmark.html

Setting	BMMC	HEST-IDC	Circle	Sketch
Replicates	3 seeds	4 folds	3 datasets/seeds	3 datasets/seeds
Training pairs	–	–	512	5,000
Query batch size	128	64	32	32
Generated particles R	64	64	16	32
Positive budget S	64	64	16	32
Validation queries	512	All validation spots	512	512
Samples per validation query	64	64	64	64
Samples per test query	64	64	128	128
Validation interval (updates)	2,000	2,000	500	500
Patience (checks)	10	10	10	10
Maximum updates	200,000	200,000	20,000	200,000
Wall-clock limit per run	72 h	72 h	48 h (campaign)	24 h

Table 3: Training and sampling settings. The R and S budgets apply to drifting; realized positive counts depend on the variant. Maximum updates also define the cosine-schedule horizon. The wall-clock limit is an additional stopping criterion; for the circle it bounds the sequential campaign rather than a single run. All methods are evaluated using their best validation-selected EMA checkpoint.

and training support are $(M, L) = (64, 512)$ for BMMC and $(32, 256)$ for every HEST fold. Their inner-validation splits use seed 20240701.

After selecting M, we calibrate a shared bandwidth h by sampling 64 pairs of training targets from each calibration query's proxy, sampling without replacement within each pair but allowing reference entries to recur across pairs. We take the median pairwise target distance within each query, then the median across calibration queries, with a floor of 10^{-6}. Distances are measured in the drifting space before CDC perturbation. The resulting bandwidth h is fixed during training. For each KGD-based run, let v_0 be the median initial field norm on its calibration batch. We set

$$\eta = \frac{0.1h}{v_0}$$

and keep it fixed throughout optimization. This makes the median initial displacement equal to 10% of the kernel bandwidth, providing a conservative update relative to the distance scale over which the kernel varies.

The positive budget S is not necessarily the realized number of positives. For BMMC, A and B use the single observed paired target, and C uses one fresh CDC perturbation of that target per iteration. D and E draw S proxy positives per query, with E applying CDC perturbations. More generally, A–C select at most S eligible observations without replacement, while D/E draw S proxy samples with replacement.

Method-specific settings. The fixed settings are summarized in Table 4. For CDC covariance estimation, as detailed in Appendix A.4, we construct a separate neighborhood in target space, distinct from the source-space neighborhood used to define the CMD proxy. Each covariance neighborhood contains the target itself and its 255 nearest training targets with representations different from the center. The distance to the eighth nearest such target sets the local Gaussian weighting scale. We adopt $k_{\mathrm{cdc}} = 256$ and $k_{\mathrm{bw}} = 8$, values used in the single-cell and image experiments of Bamberger et al. (2026, Appendix G), and hold them fixed across our real-data experiments. C/E retain the full supported covariance rank. CDC-FM retains its own estimator and numerical thresholding, as distinguished in Appendix A.4.

The fixed settings combine published configurations with explicit study-level choices. Reference Drifting uses the published temperature set and field normalization (Deng et al., 2026, Appendix A.6). For MeanFlow and iMeanFlow, p_{interval} is the probability of sampling a nonzero time interval. The values 0.25 and 0.5 follow their respective experimental settings Geng et al. (2026a;b). The iCT noise-sampling parameters and discretization curriculum follow Song & Dhariwal (2024).

Method	Setting
Reference Drifting	Temperatures $\{0.02, 0.05, 0.20\}$; unit multiplier on the sum of normalized fields
KGD variants B–E	Fixed calibrated h and $\eta = 0.1h/v_0$
CDC in C/E and CDC-FM	$\gamma_{\mathrm{cdc}} = 0.5$; maximum supported rank, subject to each estimator's numerical rules
MeanFlow / iMeanFlow	$p_{\mathrm{interval}} = 0.25$ / 0.5
Shortcut	128-level training discretization; half of each batch assigned to the bootstrap objective
iCT	$p_{\mathrm{mean}} = -1.1$, $p_{\mathrm{std}} = 2$, $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$, $\sigma_{\mathrm{data}} = 0.5$, $\rho = 7$, $s_0 = 10$, $s_1 = 1280$

Table 4: Fixed method-specific settings. Shortcut's training discretization is distinct from its inference evaluation count.

Shortcut retains the published 128-step base discretization Frans et al. (2025). Our implementation uses a bootstrap fraction of 0.5, allocating equal numbers of examples to the empirical and self-consistency objectives, rather than the 0.25 fraction used in the paper. The inference evaluation count is specified separately.

For covariance-based methods, retaining the maximum supported rank avoids introducing a separately tuned rank parameter. The common numerical setting $\gamma_{\mathrm{cdc}} = 0.5$ does not imply identical perturbation magnitudes under the different covariance estimators. The KGD bandwidth and drift multiplier follow our training-data calibration rules described above.

Reference Drifting normalizes the field separately at each temperature and sums the normalized fields (Deng et al., 2026, Appendix A.6). Our implementation evaluates affinities in linear-space float64. Under the HEST configuration ($R = 64$, 50 target coordinates), the two smaller temperatures underflow near initialization, so early training is driven by the largest temperature only.

The synthetic experiments use 512 validation pairs and 256 test conditions per replicate, with 512 training pairs for the circle task at ambient dimensions $D \in \{2, 64, 128, 256, 512\}$ and 5,000 for the sketch task at $D \in \{2, 8, 64, 256\}$. The circle MLPs have hidden width 128 and three layers; the sketch MLPs have hidden width 512 and six layers, chosen on a pilot in which the circle-sized network was capacity-limited; both use a 32-dimensional time embedding where required. Generator noise has dimension D for every method, without supplying the models with the underlying two-dimensional structure. They use the same Adam, learning-rate endpoints, weight decay, gradient clipping, and EMA settings as the real-data experiments, with the schedules in Table 3. Validation uses the single saved target at each condition. Oracle samples are reserved for final evaluation as described in Appendices B.7 and B.8.

B.3 Evaluation

Real-data test queries provide one observed target. ES can be evaluated against this observation, and averaging over held-out queries estimates expected conditional predictive performance Gneiting & Raftery (2007). In contrast, Wasserstein-1 distance to the observed point mass reduces to the expected distance from a predictive draw to that observation, rather than distance to the unobserved conditional distribution. Both synthetic tasks supply independent oracle targets at each condition, allowing us to compare single-observation ES with oracle-averaged ES and empirical conditional Wasserstein-1. This assesses whether conclusions drawn from single-observation ES agree with those obtained when the conditional distribution can be sampled directly (Appendices B.7 and B.8).

For query q, draw $K \geq 2$ independent predictive samples $\widehat{y}_{q1}, \ldots, \widehat{y}_{qK}$. Let ψ be the fixed reporting map and y_q the observed held-out target. We estimate ES as

$$\widehat{\mathrm{ES}}_q = \frac{1}{K} \sum_{k=1}^{K} \|\psi(\widehat{y}_{qk}) - \psi(y_q)\|_2 - \frac{1}{2K(K-1)} \sum_{k \neq \ell} \|\psi(\widehat{y}_{qk}) - \psi(\widehat{y}_{q\ell})\|_2. \tag{14}$$

The second sum runs over ordered distinct pairs, yielding an unbiased Monte Carlo estimator. ES is strictly proper for distributions in the reporting space with finite expected Euclidean norm Gneiting & Raftery (2007). The reporting maps and sample budgets are specified in Appendix B.1, Table 3, and the task-specific synthetic appendices.

Aggregation. For BMMC, query-level scores are averaged within each donor, then equally across donors. For HEST, ES is averaged over held-out spots within each fold, and the four folds receive equal weight. Circle scores are averaged over test conditions within each replicate. Sketch test conditions draw their stroke uniformly, rather than in proportion to stroke length as in training, and their per-query scores are likewise averaged within each replicate. All methods share the final reporting map, sample budget, and aggregation rule within each dataset.

We report means and sample standard deviations across three training seeds for BMMC, four outer folds for HEST, and three dataset/training replicates for the synthetic experiments. These standard deviations describe the corresponding sources of variation and are not confidence intervals.

Predictive-mean correlation. Pearson correlation complements ES on the biological tasks by assessing the decoded predictive mean. For BMMC, correlation is computed across proteins within each cell, followed by donor-balanced averaging. For HEST, correlation is computed across held-out spots separately for each gene, then averaged over the 50 genes within each fold, following the benchmark protocol Jaume et al. (2024). These correlations assess point predictions and do not measure the predicted uncertainty.

B.4 Computational cost

Training metric. All training costs reduce to one common metric: mean wall-clock time per optimizer update, that is, end-to-end training wall time divided by the number of updates performed. Everything inside the training loop counts, including data batching, positive retrieval, the loss and its backward pass, EMA updates, periodic validation and checkpointing. The totals in Section 4.4 are this quantity times each method's update count.

Table 5: Mean wall-clock time per optimizer update (ms), averaged over replicates. Circle: 2-D, 512 training pairs, CPU, $\leq$20k updates with patience, 3 replicates. Sketch: 2-D, 5,000 training pairs, CPU, $\leq$200k updates with patience, 3 replicates (CDC-FM: one; Reference Drifting: two). Sketch A averages two replicates, as the third reached the 24-hour wall and has no recorded training time. Sketch CDC-FM reports its single unloaded-node replicate: the other two ran on a heavily loaded node, 7.6$\times$ slower for the identical workload, and are excluded. BMMC (RNA$\rightarrow$ADT): 200k updates, 3 seeds, one A100 (CDC-FM stops on patience after 22k–72k). HEST-IDC: 4 folds, one A100 (E runs 64k updates, the others 22k–30k). Dashes: variant not run in that configuration.

	Circle	Sketch	BMMC	HEST-IDC
Reference Drifting (A)	7.1	141	4.1	471
KGD (B)	4.9	153	1.0	467
KGD-CDC (C)	5.4	–	1.1	–
KGD-proxy (D)	5.6	–	7.0	–
CMD (E)	12.0	162	8.6	470
FM	8.2	213	2.1	209
CDC-FM	–	149	107	544
MeanFlow	4.0	29	6.3	60
iMeanFlow	4.2	29	6.0	65
Shortcut	2.8	29	4.4	39
iCT	2.5	25	3.9	34

Inference metric. Sampling cost is wall-clock time per generated sample, each method using its own sampler at the stated NFE. Circle and sketch times are recorded during the final evaluation (CPU, 3 replicates each). CPU timings vary with node load.

Measured times track the nominal NFE counts: 50 steps cost $46.4\times$ (circle) and $49.9\times$ (sketch) a single step of the same flow model.

Table 6: Sampling cost. Seconds for the full evaluation set of 256 test conditions $\times$ 128 samples (CPU, D=2, 3-replicate mean). Sketch CDC-FM averages two replicates. Dashes: variant not run in that configuration.

	NFE	Circle (s)	Sketch (s)
Reference Drifting (A)	1	0.062	1.500
KGD (B)	1	0.070	1.806
KGD-CDC (C)	1	0.060	–
KGD-proxy (D)	1	0.079	–
CMD (E)	1	0.081	1.615
FM	1	0.063	1.998
FM	20	1.174	39.98
FM	50	2.923	99.72
CDC-FM	1	–	1.301
CDC-FM	20	–	25.92
CDC-FM	50	–	64.86
MeanFlow	1	0.061	1.828
iMeanFlow	1	0.062	1.487
Shortcut	1	0.062	1.752
iCT	1	0.064	1.993

B.5 Supplementary BMMC results

Table 7 reports individual-seed results and the complete FM and CDC-FM inference-budget sweep. All scores use the retained EMA checkpoints and the donor-balanced evaluation defined in Appendix B.3. For each flow model and seed, the same checkpoint is evaluated at 1, 20, and 50 NFE.

CMD achieves the lowest ES for every seed. The proxy lowers ES both with and without CDC in every seed. CDC also lowers ES both with and without the proxy in every seed. Increasing the flow budget from 1 to 20 NFE improves ES by 14.80% for FM and 4.66% for CDC-FM, with much smaller additional gains from 20 to 50 NFE: 0.64% and 0.10% respectively. Neither flow model reaches CMD at any evaluated budget. KGD-CDC has the highest mean Pearson correlation when all ablation variants are included, illustrating the difference between distributional and predictive-mean rankings.

Validation stopped by patience after 22,000–198,000 updates in all 33 runs, except for CDC-FM, optimization then continued to the 200,000-update horizon, which does not change the selected checkpoint. The three seed results share the same data partition, so their standard deviation measures training-run variation rather than variation across test populations.

B.6 Supplementary HEST-IDC results

Table 8 reports results for each held-out slide, with one training run per method and outer fold. CMD has the lowest ES on TENX95, NCBI785, and NCBI783, while CDC-FM at 50 NFE is best on TENX99. CMD improves on KGD and Reference Drifting in every fold. The Pearson winners vary across slides: MeanFlow on TENX99, Shortcut on TENX95, CMD on NCBI785, and iMeanFlow on NCBI783. MeanFlow has the highest Pearson correlation averaged across folds. The reported standard deviations describe variation across these four outer folds, not repeated training seeds within a fold.

Table 9 gives the complete flow inference-budget sweep using the same retained checkpoints. Both flow models improve mean ES as NFE increases, but remain above CMD at 50 NFE. Pearson does not improve monotonically with the inference budget.

All 36 runs stop by patience after 22,000–82,000 updates. CMD runs for 48,000, 82,000, 44,000, and 82,000 updates in folds 0–3, respectively, compared with 22,000–30,000 for the other methods. Every run stops 20,000 updates after its retained checkpoint, consistent with ten unsuccessful validation checks at 2,000-update intervals.

(a) Energy Score ↓

Method	NFE	Seed 42	Seed 43	Seed 44	Mean ± SD
FM	1	8.223	8.289	8.173	8.228 ± 0.058
FM	20	7.010	7.054	6.966	7.010 ± 0.044
FM	50	6.965	7.008	6.922	6.965 ± 0.043
CDC-FM	1	7.074	7.121	7.657	7.284 ± 0.324
CDC-FM	20	7.001	7.019	6.814	6.945 ± 0.113
CDC-FM	50	7.001	7.019	6.796	6.938 ± 0.124
MeanFlow	1	7.220	7.173	7.123	7.172 ± 0.049
iMeanFlow	1	7.162	7.134	7.086	7.128 ± 0.038
Shortcut	1	7.019	7.035	7.007	7.020 ± 0.014
iCT	1	6.980	7.012	7.057	7.017 ± 0.038
Reference Drifting (A)	1	8.336	8.287	8.277	8.300 ± 0.031
KGD (B)	1	6.868	6.862	6.834	6.855 ± 0.018
KGD-CDC (C)	1	6.772	6.821	6.774	6.789 ± 0.028
KGD-proxy (D)	1	6.611	6.620	6.603	6.611 ± 0.009
CMD (E)	1	**6.543**	**6.557**	**6.540**	$\mathbf{6.547 \pm 0.009}$

(b) Pearson ↑

Method	NFE	Seed 42	Seed 43	Seed 44	Mean ± SD
FM	1	0.880	0.879	0.881	0.880 ± 0.001
FM	20	0.882	0.881	0.882	0.881 ± 0.000
FM	50	0.882	0.881	0.882	0.882 ± 0.000
CDC-FM	1	0.859	0.853	0.882	0.864 ± 0.015
CDC-FM	20	0.866	0.863	0.882	0.870 ± 0.010
CDC-FM	50	0.866	0.863	0.882	0.870 ± 0.010
MeanFlow	1	0.874	0.875	0.877	0.876 ± 0.002
iMeanFlow	1	0.877	0.878	0.879	0.878 ± 0.001
Shortcut	1	0.879	0.880	0.881	0.880 ± 0.001
iCT	1	**0.883**	0.881	0.880	0.881 ± 0.002
Reference Drifting (A)	1	0.872	0.870	0.872	0.871 ± 0.001
KGD (B)	1	0.881	0.882	0.882	0.882 ± 0.000
KGD-CDC (C)	1	0.880	**0.883**	**0.883**	$\mathbf{0.882 \pm 0.001}$
KGD-proxy (D)	1	0.877	0.877	0.877	0.877 ± 0.000
CMD (E)	1	0.877	0.877	0.877	0.877 ± 0.000

Table 7: Complete BMMC RNA-to-protein results. Each seed column reports a test summary for the same 15,066 cells, using 64 predictive samples per cell. The final column gives the mean and sample standard deviation across seeds. Bold marks the best mean or individual-seed value within each panel, using values before rounding and without a significance claim. FM and CDC-FM inference budgets reuse the same trained checkpoints.

B.7 Synthetic conditional-circle experiment

Ambient-dimensionality sweep. We embed the same two-dimensional source and target vectors using independently sampled random matrices with orthonormal columns, held fixed throughout training and evaluation: $x_D = A_D x$ and $y_D = B_D y$, with $A_D^\top A_D = B_D^\top B_D = I_2$, for $D \in \{64, 128, 256, 512\}$. The maps preserve Euclidean distances between embedded observations and do not change the intrinsic conditional problem. The generator operates in the ambient coordinates, receives D-dimensional noise, and is not given either embedding matrix. We retain 512 training pairs and the same hidden width of 128 across dimensions. Input/output dimensions, and therefore total parameter counts, change with D. Evaluation uses the full ambient coordinates, so errors outside the embedded target plane are not removed by projection.

The conditional W_1 and both ES summaries are reported in Tables 10–12. D/E are the leading pair at every dimension under all three metrics, although their relative ordering varies. For example, at $D = 128$, D has lower mean W_1 and oracle-averaged ES, while E has slightly lower single-observation ES. The agreement therefore concerns the main method comparisons, not identical rankings for every small difference. This is an empirical comparison of estimators; propriety of ES follows from its theoretical definition, not from agreement on this experiment.

(a) Energy Score ↓

Method	NFE	TENX99	TENX95	NCBI785	NCBI783	Mean ± SD
FM	50	5.416	8.157	7.833	5.666	6.768 ± 1.427
CDC-FM	50	**5.058**	8.136	7.383	5.535	6.528 ± 1.468
MeanFlow	1	5.728	9.251	7.572	5.929	7.120 ± 1.643
iMeanFlow	1	5.672	9.040	7.378	5.734	6.956 ± 1.598
Shortcut	1	5.434	8.045	6.709	5.228	6.354 ± 1.304
iCT	1	5.377	8.339	7.928	5.482	6.781 ± 1.571
Reference Drifting (A)	1	6.000	8.157	7.351	5.970	6.870 ± 1.073
KGD (B)	1	5.656	7.955	7.359	5.511	6.620 ± 1.223
CMD (E)	1	5.480	**7.729**	**6.520**	**5.219**	$\mathbf{6.237 \pm 1.142}$

(b) Pearson ↑

Method	NFE	TENX99	TENX95	NCBI785	NCBI783	Mean ± SD
FM	50	0.414	0.563	0.502	0.647	0.532 ± 0.099
CDC-FM	50	0.423	0.530	0.473	0.619	0.512 ± 0.084
MeanFlow	1	**0.442**	0.572	0.509	0.660	$\mathbf{0.546 \pm 0.093}$
iMeanFlow	1	0.440	0.571	0.504	**0.663**	0.544 ± 0.095
Shortcut	1	0.422	**0.580**	0.498	0.652	0.538 ± 0.100
iCT	1	0.423	0.574	0.496	0.654	0.537 ± 0.100
Reference Drifting (A)	1	0.387	0.548	0.489	0.607	0.508 ± 0.094
KGD (B)	1	0.371	0.537	0.498	0.619	0.506 ± 0.103
CMD (E)	1	0.386	0.533	**0.514**	0.624	0.514 ± 0.098

Table 8: HEST-IDC results by held-out slide. Columns correspond to folds 0–3, respectively. The final column gives the equal-weight mean and sample standard deviation across folds. Every score uses the best validation-selected EMA checkpoint and 64 predictive samples per spot. Bold identifies the best value before rounding within each panel, without a significance claim. FM and CDC-FM use 50 NFE here.

Method	NFE	ES ↓	Mean Pearson ↑
FM	1	7.886 ± 1.679	0.531
FM	20	6.821 ± 1.439	0.532
FM	50	6.768 ± 1.427	0.532
CDC-FM	1	7.062 ± 1.492	0.516
CDC-FM	20	6.548 ± 1.475	0.512
CDC-FM	50	6.528 ± 1.468	0.512

Table 9: HEST-IDC flow inference-budget sweep. ES is reported as mean ± sample standard deviation across four folds. Pearson is the mean across the same folds. All budgets reuse each fold's retained checkpoint.

Independent oracle-versus-oracle sampling gives $W_1 = 0.083 \pm 0.002$ at every dimension. Direct sampling from the frozen proxy gives approximately 0.156 ± 0.013, or 0.154 ± 0.014 after CDC smoothing, also essentially unchanged with dimension. These are finite-sample reference scores, not lower bounds that every estimated model score must exceed. Their stability separates the unchanged proxy approximation from the dimension-dependent difficulty of fitting a generator. The selected breadths are $64, 64, 32$ for replicates 42, 43, and 44, respectively, at each of the four dimensions.

Training limits. All methods complete three replicates at $D \leq 256$. CDC-FM follows the published update, which eigendecomposes a $D \times D$ local covariance for every training example; its per-update cost grows by $2.3\times$ and $4.0\times$ across the two doublings from $D = 64$ to $D = 256$, and its three $D = 512$ runs, at an extrapolated 26–52 hours each, could not complete within the campaign's 48-hour allocation and were stopped, so that entry is marked n.c. Where it completes, CDC-FM reproduces FM to three decimals at every dimension and budget: under the published capping rule, the 512 training targets on the unit circle give a covariance cap of order 10^{-6}, so the perturbation is numerically inert on this task. FM and CDC-FM reach the 20,000-update limit at

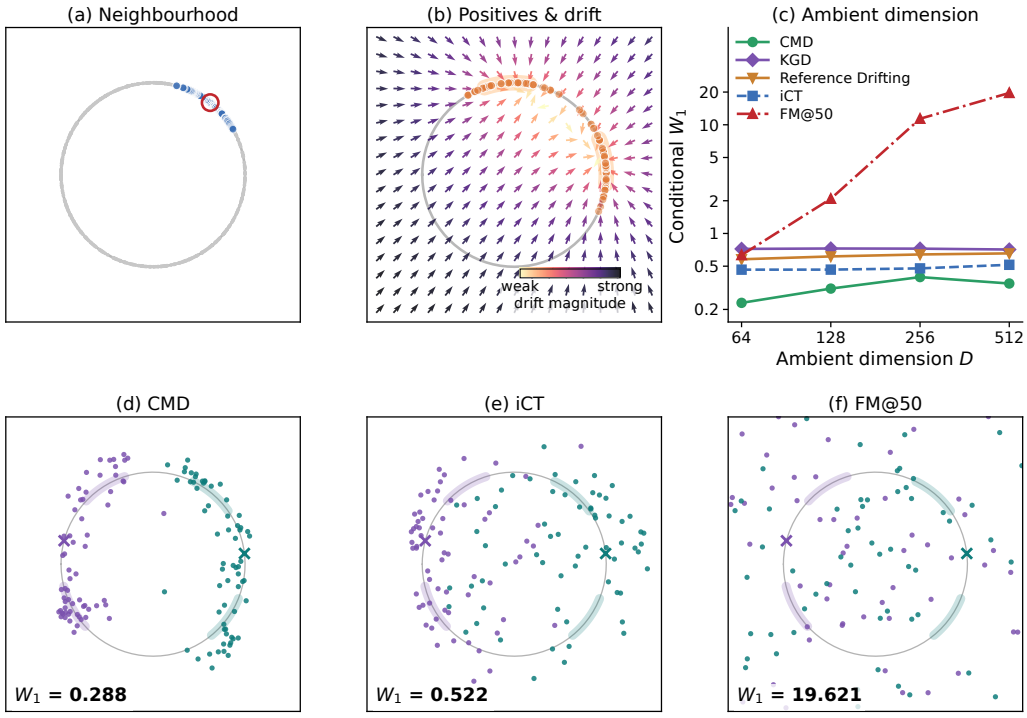

Figure 3: Two-arc circle transport: a point at angle t on the source circle is paired with a point at angle $t \pm u$ on the target circle, $u \sim \mathcal{U}[0.5, 1]$ rad, so every conditional is two arcs. **(a)** A query (ring) and its M=64 retrieved training sources (blue). **(b)** Their targets, the proxy positives (orange), cover the query's two-arc conditional (shaded); arrows show the drift direction they induce, coloured by magnitude. **(c)** Conditional W_1 as the planar task is embedded isometrically in D dimensions (mean of three seeds; solid: drifting methods). **(d)–(f)** Test samples for two conditions ($\times$) at D=512 (one seed), projected to the plane, with the true support shaded. FM@50 uses 50 steps; all others one.

every dimension, as do all D/E runs at $D = 512$. The results therefore describe performance under the stated architecture and training budget, not converged population optima.

B.8 Synthetic cat-to-dog stroke experiment

Construction. We use one fixed cat drawing and one fixed dog drawing from Quick, Draw! Google Creative Lab, each containing 15 strokes. The dataset represents strokes as sequences of two-dimensional pen positions. We treat each stroke as a polyline through its recorded pen positions and draw points uniformly along its ink length, rather than choosing among the recorded positions, which are denser where the pen moved slowly. Stroke j in the source drawing is assigned to stroke j in the target drawing. The cross-drawing correspondence is imposed by our synthetic construction. The same drawing pair is used across replicates.

Let $\gamma_j^{\mathrm{s}}, \gamma_j^{\mathrm{t}} : [0,1] \to \mathbb{R}^2$ parameterize the source and target strokes by normalized arc length, and let ℓ_j^{s} be the length of source stroke j. Training pairs follow

$$\Pr(J = j) = \frac{\ell_j^{\mathrm{s}}}{\sum_{k=1}^{15} \ell_k^{\mathrm{s}}}, \qquad U, V \overset{\mathrm{iid}}{\sim} \mathrm{Uniform}(0,1), \qquad X = \gamma_J^{\mathrm{s}}(U), \quad Y = \gamma_J^{\mathrm{t}}(V),$$

with U and V independent of J. Thus, except at points where source strokes cross, every position on a given source stroke shares the same target distribution: a point spread uniformly along the ink of the corresponding target stroke. Unlike the moving conditional arcs in the circle experiment, this construction has 15 stroke-specific conditional laws. Training reveals one sampled target per source. Independent draws of V provide oracle targets at each held-out condition.

Method	NFE	$D = 64$	$D = 128$	$D = 256$	$D = 512$
Reference Drifting (A)	1	0.579 ± 0.004	0.615 ± 0.007	0.641 ± 0.017	0.656 ± 0.014
KGD (B)	1	0.721 ± 0.002	0.727 ± 0.002	0.725 ± 0.006	0.712 ± 0.009
KGD-CDC (C)	1	0.721 ± 0.003	0.728 ± 0.002	0.726 ± 0.004	0.714 ± 0.014
KGD-proxy (D)	1	0.319 ± 0.197	$\mathbf{0.297 \pm 0.084}$	$\mathbf{0.358 \pm 0.177}$	0.350 ± 0.042
CMD (E)	1	$\mathbf{0.230 \pm 0.022}$	0.311 ± 0.076	0.397 ± 0.149	$\mathbf{0.346 \pm 0.051}$
FM	1	1.493 ± 0.020	3.461 ± 0.022	11.651 ± 0.007	19.832 ± 0.005
FM	20	0.602 ± 0.004	2.048 ± 0.027	11.359 ± 0.005	19.622 ± 0.003
FM	50	0.633 ± 0.005	2.086 ± 0.026	11.364 ± 0.005	19.625 ± 0.003
CDC-FM	1	1.493 ± 0.019	3.461 ± 0.022	11.651 ± 0.007	n.c.
CDC-FM	20	0.602 ± 0.004	2.048 ± 0.027	11.359 ± 0.005	n.c.
CDC-FM	50	0.633 ± 0.005	2.086 ± 0.026	11.364 ± 0.005	n.c.
MeanFlow	1	1.114 ± 0.042	2.888 ± 0.058	11.485 ± 0.005	19.886 ± 0.003
iMeanFlow	1	1.165 ± 0.030	2.911 ± 0.061	11.461 ± 0.005	19.799 ± 0.001
Shortcut	1	1.302 ± 0.033	3.487 ± 0.053	11.498 ± 0.010	19.665 ± 0.009
iCT	1	0.464 ± 0.009	0.465 ± 0.005	0.477 ± 0.002	0.515 ± 0.020

Table 10: Conditional Wasserstein-1 against oracle samples on the conditional-circle task under increasing ambient dimension; lower is better. Mean $\pm$ sample standard deviation across three replicates; n.c.: not completed within the 48-hour allocation. Bold marks the lowest mean per column. All flow budgets reuse the same trained checkpoints.

Method	NFE	$D = 64$	$D = 128$	$D = 256$	$D = 512$
Reference Drifting (A)	1	0.452 ± 0.010	0.453 ± 0.005	0.454 ± 0.008	0.456 ± 0.006
KGD (B)	1	0.469 ± 0.008	0.469 ± 0.009	0.469 ± 0.008	0.468 ± 0.008
KGD-CDC (C)	1	0.468 ± 0.009	0.469 ± 0.008	0.468 ± 0.007	0.469 ± 0.006
KGD-proxy (D)	1	0.414 ± 0.041	0.409 ± 0.016	$\mathbf{0.420 \pm 0.024}$	0.419 ± 0.003
CMD (E)	1	$\mathbf{0.396 \pm 0.006}$	$\mathbf{0.408 \pm 0.011}$	0.426 ± 0.020	$\mathbf{0.414 \pm 0.001}$
FM	1	0.583 ± 0.009	1.100 ± 0.003	3.457 ± 0.002	5.838 ± 0.002
FM	20	0.455 ± 0.006	0.723 ± 0.006	3.370 ± 0.002	5.774 ± 0.001
FM	50	0.456 ± 0.006	0.733 ± 0.006	3.371 ± 0.002	5.775 ± 0.001
CDC-FM	1	0.583 ± 0.009	1.100 ± 0.003	3.457 ± 0.002	n.c.
CDC-FM	20	0.455 ± 0.006	0.723 ± 0.006	3.370 ± 0.002	n.c.
CDC-FM	50	0.456 ± 0.006	0.733 ± 0.006	3.371 ± 0.002	n.c.
MeanFlow	1	0.536 ± 0.021	0.959 ± 0.022	3.410 ± 0.002	5.852 ± 0.001
iMeanFlow	1	0.532 ± 0.016	0.970 ± 0.029	3.403 ± 0.002	5.826 ± 0.001
Shortcut	1	0.549 ± 0.005	1.111 ± 0.014	3.408 ± 0.003	5.797 ± 0.004
iCT	1	0.441 ± 0.004	0.446 ± 0.009	0.444 ± 0.009	0.455 ± 0.004

Table 11: Single-observation Energy Score on the conditional-circle task under increasing ambient dimension; lower is better. Mean $\pm$ sample standard deviation across three replicates; n.c.: not completed within the 48-hour allocation. Bold marks the lowest mean per column. All flow budgets reuse the same trained checkpoints.

Nearby points on different source strokes can have different target conditionals. Consequently, this geometry tests source-neighborhood transfer where spatial proximity need not imply conditional similarity. The results alone do not isolate errors caused by neighborhood mixing from errors introduced while fitting the generator.

Replicates and ambient dimensions. We generate 5,000 training pairs per replicate and use three paired dataset/training seeds, 42, 43, and 44. Test evaluation uses 256 source conditions with uniform stroke sampling, giving each conditional equal probability rather than weighting it by source-stroke length. We evaluate ambient dimensions $D \in \{2, 8, 64, 256\}$. For $D > 2$, independent fixed random matrices with orthonormal columns embed the source and target coordinates, as in Appendix B.7. These embeddings preserve distances between data points and the underlying conditional law, while increasing the ambient dimensions presented to the models.

Training and evaluation. We compare Reference Drifting, KGD, CMD, FM, CDC-FM, MeanFlow, iMeanFlow, Shortcut, and iCT. The sketch campaign uses a 200,000-update ceiling, patience

Method	NFE	$D = 64$	$D = 128$	$D = 256$	$D = 512$
Reference Drifting (A)	1	0.451 ± 0.001	0.452 ± 0.002	0.454 ± 0.001	0.455 ± 0.002
KGD (B)	1	0.468 ± 0.001	0.468 ± 0.002	0.469 ± 0.002	0.468 ± 0.002
KGD-CDC (C)	1	0.468 ± 0.002	0.468 ± 0.003	0.469 ± 0.002	0.469 ± 0.001
KGD-proxy (D)	1	0.411 ± 0.036	$\mathbf{0.406 \pm 0.013}$	$\mathbf{0.418 \pm 0.028}$	0.416 ± 0.004
CMD (E)	1	$\mathbf{0.395 \pm 0.005}$	0.408 ± 0.011	0.423 ± 0.024	$\mathbf{0.415 \pm 0.006}$
FM	1	0.584 ± 0.004	1.099 ± 0.005	3.457 ± 0.003	5.838 ± 0.001
FM	20	0.455 ± 0.001	0.723 ± 0.008	3.370 ± 0.002	5.774 ± 0.001
FM	50	0.456 ± 0.001	0.732 ± 0.008	3.371 ± 0.002	5.775 ± 0.001
CDC-FM	1	0.584 ± 0.004	1.099 ± 0.005	3.457 ± 0.003	n.c.
CDC-FM	20	0.455 ± 0.001	0.723 ± 0.008	3.370 ± 0.002	n.c.
CDC-FM	50	0.456 ± 0.001	0.732 ± 0.008	3.371 ± 0.002	n.c.
MeanFlow	1	0.538 ± 0.012	0.959 ± 0.020	3.410 ± 0.001	5.852 ± 0.001
iMeanFlow	1	0.534 ± 0.009	0.970 ± 0.027	3.403 ± 0.001	5.826 ± 0.001
Shortcut	1	0.549 ± 0.005	1.111 ± 0.013	3.408 ± 0.003	5.797 ± 0.004
iCT	1	0.441 ± 0.000	0.442 ± 0.003	0.444 ± 0.002	0.452 ± 0.001

Table 12: Oracle-averaged Energy Score on the conditional-circle task under increasing ambient dimension; lower is better. Mean $\pm$ sample standard deviation across three replicates; n.c.: not completed within the 48-hour allocation. Bold marks the lowest mean per column. All flow budgets reuse the same trained checkpoints.

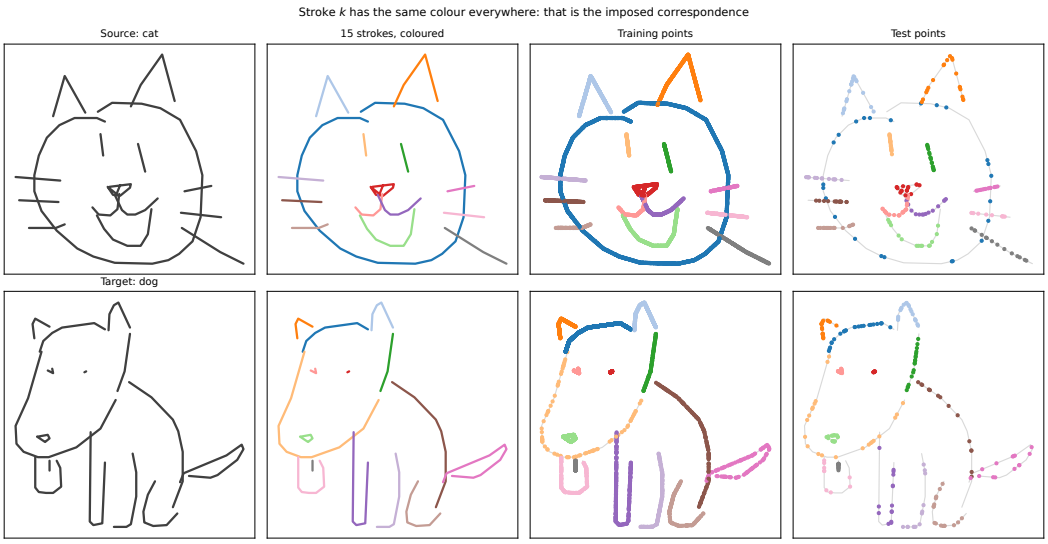

Figure 4: Construction of the cat→dog task: source (top) and target (bottom). Each drawing has 15 strokes, and stroke k has the same colour in both. A training pair takes a point uniformly along the cat's ink, so each stroke in proportion to its length, and pairs it with an independent point drawn uniformly along the dog stroke with the same index. The conditional of any cat point is therefore the whole matching dog stroke. Test sources choose their stroke uniformly (5000 training pairs, 256 test sources). These colours are used for generated points throughout.

of ten validation checks, and a 24-hour wall-clock limit per run; a run that reaches the wall is scored at the checkpoint selected up to that point. The generators contain approximately 3.17 million parameters at $D = 2$ and 3.56 million at $D = 256$, with small differences for method-specific conditioning. Its training settings are the Sketch column of Table 3, a separate configuration from the smaller circle models. We report conditional W_1 against oracle samples and oracle-averaged ES, averaged over test conditions and then summarized across three replicates. FM and CDC-FM are evaluated at 1, 20, and 50 NFE using the same selected checkpoint; all other methods use one NFE.

Conditional recovery. Tables 13 and 14 summarize the dimension-wise results. CMD has the lowest mean conditional W_1 at all four dimensions: 0.091, 0.095, 0.088, and 0.084 at $D =$

Method	NFE	$D = 2$	$D = 8$	$D = 64$	$D = 256$
Reference Drifting (A)	1	0.250 ± 0.026^{w}	0.246 ± 0.062	0.283 ± 0.033	0.337 ± 0.082^{w}
KGD (B)	1	0.118 ± 0.007	0.164 ± 0.004	0.183 ± 0.001	0.179 ± 0.004
CMD (E)	1	0.091 ± 0.012	0.095 ± 0.012	0.088 ± 0.014	0.084 ± 0.007
FM	1	0.178 ± 0.012	0.198 ± 0.014	0.452 ± 0.023	1.985 ± 0.046
FM	20	0.108 ± 0.016	0.113 ± 0.017	0.333 ± 0.033	0.763 ± 0.024
FM	50	0.110 ± 0.017	0.114 ± 0.017	0.339 ± 0.033	0.788 ± 0.019
CDC-FM	1	0.206 ± 0.038^{w}	0.274 ± 0.047^{w}	0.786 ± 0.076^{w}	1.947 ± 0.010^{w}
CDC-FM	20	0.145 ± 0.042^{w}	0.198 ± 0.067^{w}	0.688 ± 0.076^{w}	0.781 ± 0.028^{w}
CDC-FM	50	0.148 ± 0.043^{w}	0.202 ± 0.069^{w}	0.696 ± 0.077^{w}	0.819 ± 0.036^{w}
MeanFlow	1	0.232 ± 0.037	0.318 ± 0.043	0.475 ± 0.105	1.043 ± 0.011
iMeanFlow	1	0.271 ± 0.077	0.476 ± 0.244	0.392 ± 0.023	0.976 ± 0.008
Shortcut	1	0.124 ± 0.017	0.155 ± 0.017	0.500 ± 0.005	0.827 ± 0.021
iCT	1	0.172 ± 0.034	0.219 ± 0.007	0.484 ± 0.272	0.639 ± 0.249

Table 13: Conditional Wasserstein-1 on the cat-to-dog stroke task; lower is better. Mean $\pm$ sample standard deviation across three replicates. ^{w}: includes replicates stopped by the 24-hour wall and scored at their selected checkpoint (all three for CDC-FM at $D = 8$ and $D = 64$, two at $D = 256$, one otherwise). The FM and CDC-FM inference budgets reuse each trained checkpoint.

Method	NFE	$D = 2$	$D = 8$	$D = 64$	$D = 256$
Reference Drifting (A)	1	0.169 ± 0.012^{w}	0.168 ± 0.022	0.184 ± 0.015	0.208 ± 0.035^{w}
KGD (B)	1	0.126 ± 0.004	0.121 ± 0.001	0.115 ± 0.002	0.113 ± 0.003
CMD (E)	1	0.115 ± 0.007	0.120 ± 0.008	0.110 ± 0.009	0.112 ± 0.007
FM	1	0.171 ± 0.010	0.166 ± 0.010	0.292 ± 0.022	0.679 ± 0.009
FM	20	0.117 ± 0.006	0.116 ± 0.005	0.202 ± 0.019	0.466 ± 0.023
FM	50	0.117 ± 0.006	0.115 ± 0.005	0.203 ± 0.019	0.464 ± 0.023
CDC-FM	1	0.197 ± 0.034^{w}	0.227 ± 0.038^{w}	0.477 ± 0.034^{w}	0.673 ± 0.009^{w}
CDC-FM	20	0.131 ± 0.019^{w}	0.147 ± 0.023^{w}	0.432 ± 0.057^{w}	0.464 ± 0.023^{w}
CDC-FM	50	0.131 ± 0.019^{w}	0.147 ± 0.024^{w}	0.430 ± 0.056^{w}	0.464 ± 0.022^{w}
MeanFlow	1	0.187 ± 0.016	0.241 ± 0.033	0.344 ± 0.089	0.504 ± 0.018
iMeanFlow	1	0.214 ± 0.042	0.358 ± 0.169	0.277 ± 0.016	0.504 ± 0.020
Shortcut	1	0.124 ± 0.007	0.132 ± 0.008	0.280 ± 0.002	0.477 ± 0.019
iCT	1	0.141 ± 0.018	0.156 ± 0.002	0.304 ± 0.174	0.428 ± 0.195

Table 14: Oracle-averaged Energy Score on the cat-to-dog stroke task; lower is better. Mean $\pm$ sample standard deviation across three replicates. ^{w}: includes replicates stopped by the 24-hour wall and scored at their selected checkpoint (all three for CDC-FM at $D = 8$ and $D = 64$, two at $D = 256$, one otherwise). The FM and CDC-FM inference budgets reuse each trained checkpoint.

$2, 8, 64, 256$. Its nearest competitor is FM at 20 NFE in low dimension (0.108 at $D = 2$, 0.113 at $D = 8$) and KGD in high dimension (0.183 at $D = 64$, 0.179 at $D = 256$), while Reference Drifting obtains 0.250–0.337. CMD therefore retains its Wasserstein advantage through the full sweep, and its absolute error does not grow with dimension, whereas every non-drifting method degrades (FM at 20 NFE rises from 0.108 to 0.763).

The ES comparison supports the same conclusion with one exception at $D = 8$, where FM at 50 NFE achieves oracle ES 0.115 against 0.120 for CMD despite a larger W_1 (0.114 versus 0.095). At $D = 64$ and $D = 256$, CMD has the lowest oracle ES (0.110, 0.112) with KGD close behind (0.115, 0.113). Agreement between the two distributional criteria is thus substantive rather than an identity of all finite-sample rankings.

Numerical outcomes. CDC-FM follows the published update, which eigendecomposes a $D \times D$ local covariance for every training example. Seven of its twelve runs reached the 24-hour wall and are scored at their selected checkpoint (14.5k–77k updates); the other five stopped on patience. Its scores are therefore a fixed-budget outcome rather than a converged one. Two Reference Drifting runs ($D = 2$, $D = 256$) also reached the wall. One Shortcut run at $D = 8$ and all three at $D = 64$ reached the update ceiling; every remaining run stopped on patience.