CONTRASTIVE BECKMANN MAPS: GUIDING ONE-STEP GENERATORS WITHOUT A VELOCITY FIELD

Anonymous authors
Paper under double-blind review

ABSTRACT

Beckmann Transport Models (BTM) recently opened a promising avenue for learning one-step generative maps end-to-end, bypassing the need for distillation, intermediate targets, or external features. However, its quality has so far lagged behind state-of-the-art methods, which largely depend on condition strengthening techniques, such as classifier-free guidance for flows and diffusions: guidance combines velocity or score predictions, and a native map has neither. We address this limitation by introducing Contrastive Beckmann Maps (CBM). We observe that the object defining a BTM, its probability current, is linear in the target distribution, so guidance can be applied to the current itself. CBM subtracts the current toward mismatched image–label pairs, obtained by pairing each label with another image in the batch. Under the assumptions of our analysis, the ideal map avoids the negative region of the contrastive target, while sampling still requires one network call. Contrary to CFG, CBM is a theoretically grounded training technique tailored to the BTM framework, repelling correct samples from negatively drawn samples during the training phase. On ImageNet 256×256, fine-tuning a converged BTM model with CBM for around 15% of an epoch lowers one-step FID from 20.8 to 6.7 and **5.5** with postprocessing. Trained from scratch, CBM reaches the same FID of an optimized BTM with $3.5\times$ fewer epochs. On CIFAR-10, CBM improves one-step FID from 7.5 to 5.3.



Figure 1: **One-step samples from BTM (top) and CBM (bottom)** for five ImageNet classes, with the same noise (curated). BTM samples drift toward neighbouring classes: the arctic fox resembles a weasel, and the lighthouse and volcano resemble a breakwater and a seashore.

1 INTRODUCTION

One-step generators speed up diffusion and flow-matching samplers (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023) with a single neural network call. Most existent one-step samplers are either trained in two stages, where one first trains a multistep

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.

sampler and then distill it into one step, or by self-distillation (Salimans & Ho, 2022; Song et al., 2023; Yin et al., 2024b;a; Salimans et al., 2024; Boffi et al., 2025). The multistep flow and one-step sampler may also be learned jointly with (Geng et al., 2025a;b). Beckmann Transport Models (BTM) (Lee et al., 2026; Tang & Wang, 2026) take a direct route. The endpoint map of a time-independent flow is constant along the flow's probability current. Enforcing the endpoint map being constant along the current gives a loss that trains the map directly from data. There is no teacher, no second prediction head and, unlike drifting models (Deng et al., 2026), no large pretrained feature network.

The simplicity of BTM has cost a great deal of quality. BTM reaches a one-step FID of 17.6 on ImageNet256 after 1280 epochs, while MeanFlow reaches 3.43 (Lee et al., 2026; Geng et al., 2025a). The failure mode of BTM is shown in Figure 1 where BTM mixing up images from different classes. It generates an arctic fox similar to a weasel, and the lighthouse and volcano resemble a breakwater and a seashore. Mixing classes gives images high FID, which is not improved by continuing training as shown by Fig. 3. On CIFAR-10 BTM gets worse with further training at every learning rate. The early plateau of BTM and slow convergence of BTM may due to high variance gradient. Conditional diffusion models share weakness of gradient variance, and supplying an extra signal with guidance has been crucial for the model to be performant. The standard guidance remedy is classifier-free guidance (CFG) (Ho & Salimans, 2022). CFG lowers the FID of SiT (Ma et al., 2024) from 8.3 to 2.06, and is baked into training for one of the state-of-the-art one-step generators Geng et al. (2025b). The BTM paper Lee et al. (2026) leaves extending CFG to BTM as an open problem.

Guidance does not carry over directly to training native one-step generators. CFG combines two predictions of a vector field, the conditional and unconditional velocity or score, and training-time variants build such a combination into a velocity target (Geng et al., 2025a). A BTM has no velocity or score to combine; it has only the map. One can extrapolate between the map's outputs under the correct versus incorrect labels at inference time as in Appendix D.2, at the cost of increasing sampling NFEs. We wish to guide BTM during training so that sampling is still one NFE.

To guide BTM at training time whose velocity is absent, we return to the definition of BTM as the map that is constant along a current $J(x)$. Time-independent velocities do not add when target distributions compose. However, the current toward a mixture of targets is the same mixture of their currents. The current of the time-independent flow is velocity times density. Contrastive Beckmann Maps (CBM) apply guidance to current, by subtracting the current toward a *negative* distribution of mislabelled image from the current towards the target distribution up to a reweighting ω. Under the assumptions of Proposition 5, the ideal CBM-Transport map sends no mass to the negative region of the contrastive target (§3). There are two ways of implementing CBM: CBM-Transport and CBM-Endpoint. CBM-Transport adds one sign-flipped transport term to the BTM loss, and CBM-Endpoint applies the same contrast to the map's outputs. Earlier work contrasts matched and mismatched pairs through a velocity field (**?**) or in pretrained feature space (Deng et al., 2026), but CBM contrasts the current.

Guidance addresses both the plateau and slow training simultaneously. Fine-tuning the converged DiT-XL/2 BTM model with CBM-Transport for 6,000 steps and 15% of an epoch, lowers one-step FID from 20.8 to 6.7 and 5.5 after postprocessing, against 18.3 for continued BTM training (Fig. 3). Classifier predictions become sharper: the median effective number of classes decreases from 4.4 to 2.0 (Fig. 4). Precision rises from 48 to close to the real data. We observe a fidelity–diversity tradeoff similar to CFG. As precision increases with ω, recall falls. Trained from scratch, CBM-Endpoint reaches the FID of an optimized BTM recipe at 50 epochs with about $3.5\times$ fewer epochs and half the compute. A short CBM-Transport dose at the end lowers FID further (Fig. 6). BTM's slow training and its plateau are both repaired by contrastive class distribution signals.

- **Guidance without a velocity field.** We introduce contrastive Beckmann currents and show that the contrastive map is conserved along the currents. CBM generates conditional images in one step; under the assumptions of Proposition 5, the ideal CBM-Transport map avoids the negative region of the contrastive target (§3).
- **Guidance addresses BTM plateau and speeds up training.** CBM lowers the one-step ImageNet FID of the converged BTM checkpoint from 20.8 to 6.7 and 5.5 with postprocessing (Appendix C.3). Pretraining from scratch, CBM reaches a same FID $3.5\times$ faster than an optimized BTM.

- **Learned guidance beats inference-time guidance.** We turn BTM into a multistep sampler with guidance by extrapolates BTM's output away from the output on another label. With six network calls it moves BTM from FID 20.8 to 15.2, higher than the 6.7 FID of CBM with one NFE.

2 Background: Beckmann Transport Models

Let the interpolant be $I_t = \alpha_t I_0 + \beta_t I_1$, $t \in [0,1]$, with noise $I_0 \sim \mu_0 = \mathcal{N}(0, \mathbb{I}_d)$ and data $I_1 \sim \mu_1$, with $\alpha_0 = \beta_1 = 1$ and $\alpha_1 = \beta_0 = 0$. Let ρ_t be the density of I_t and $b_t(x) = \mathbb{E}[\dot{I}_t \mid I_t = x]$ be the flow-matching velocity. BTM (Lee et al., 2026) replaces b_t by a time-independent current which averages over time. The current and occupation measure is defined by

$$J(x) = \int_0^1 \rho_t(x)\, b_t(x)\, dt, \qquad \nu(x) = \int_0^1 \rho_t(x)\, dt. \tag{1}$$

The autonomous velocity $b(x)$ is defined by the ratio $b = J(x)/\nu(x)$, which can be written $b(x)$ as the minimizer of

$$b(x) = \arg\min_{\hat{b}} \mathbb{E}_{t,I_0,I_1} ||\hat{b}(I_t) - \dot{I}_t||^2, \tag{2}$$

from which $b(x)$ can be trained. The occupation measure $\nu(x)$ is the density of I_t for $t \sim \mathcal{U}[0,1]$, and $\nu(x)\, dx$ is the expected time an interpolant spends in dx. Integrating the continuity equation $\partial_t \rho_t + \nabla \cdot (\rho_t b_t) = 0$ over t gives

$$\nabla \cdot J = \mu_0 - \mu_1, \tag{3}$$

the flux constraint of Beckmann's transportation problem (Beckmann, 1952; Santambrogio, 2015). When μ_1 is supported on a manifold $\mathcal{M}$ of codimension at least two, the ODE $\dot{X} = b(X)$ pushes the distribution μ_0 forward to μ_1 as supported on manifold $\mathcal{M}$ (Lee et al., 2026) along the trajectories of the ODE. The map T sending a point along a trajectory to the endpoint is constant along the flow lines of $b(x)$. The flow lines of $b(x)$ are also flow lines of $J(x)$.

$$J(x) \cdot \nabla T(x) = 0 \quad \text{for } x \notin \mathcal{M}, \qquad T(x) = x \quad \text{for } x \in \mathcal{M}, \tag{4}$$

where $J \cdot \nabla T(x)$ is the derivative of T in the direction $J(x)$. BTM learns the conservation of $T(x)$ in the $J(x)$ direction with a stop-gradient transport loss and a boundary loss,

$$\mathcal{L}(\theta) = \underbrace{\mathbb{E}_{t,I_0,I_1} \left\| T_\theta(I_t) - \mathrm{sg}\left(T_\theta(I_t) + \dot{I}_t \cdot \nabla T_\theta(I_t)\right)\right\|^2}_{\mathcal{L}_{\mathrm{tr}}(\theta)} + \lambda_\partial \underbrace{\mathbb{E}_{I_1} \left\| T_\theta(I_1) - I_1 \right\|^2}_{\mathcal{L}_\partial(\theta)}. \tag{5}$$

The gradient of loss (5) vanishes at the ground truth Beckmann Transport Map (Lee et al., 2026, Theorem 3). For class-conditional generation, every quantity above is conditioned on a label c, the map $T_\theta(x, c)$ takes c as input, and $I_1 \sim \mu_1(\cdot \mid c)$. Both terms of (5) have never seen for a given class what does a wrong image look like.

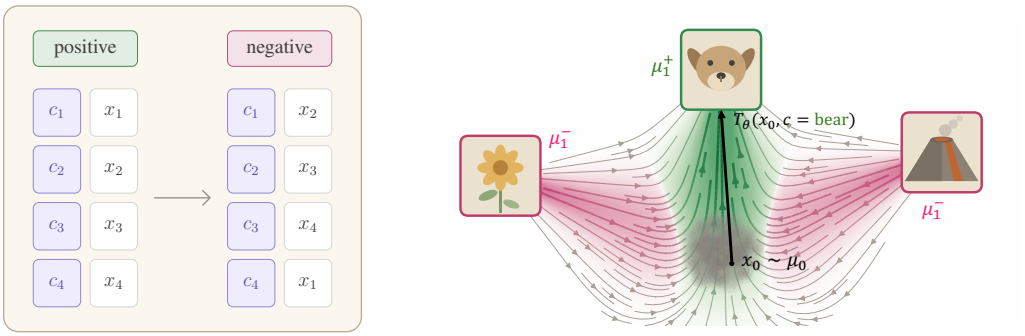
Remark 1. *Currents in equation* (1) *can be added when target distributions compose, but velocities in* (2) *can not. Since I_0 is independent of I_1, $J(x) = \int_0^1 \mathbb{E}[\dot{I}_t\, \delta(x - I_t)]\, dt$ and ν are expectations under $\mu_0 \otimes \mu_1$ and are linear in μ_1: if $\mu_1 = \sum_k a_k \mu_1^{(k)}$, then $J = \sum_k a_k J^{(k)}$ and $\nu = \sum_k a_k \nu^{(k)}$. The velocity $b = J/\nu$ is a ratio, and is not additive across individual velocities $b^k(x)$.*

3 Contrastive Beckmann Maps

3.1 The contrastive current and its target

In the CBM loss, not only do we teach the map $T_\theta(x)$ with the *positive* distribution $\mu_1^+(\cdot \mid c) = \mu_1(\cdot \mid c)$ equal the data of class c, but also what the wrong images for a class look like with the negative distribution denoted as $\mu_1^-(\cdot \mid c)$. It is the distribution of images not in class c. We sample the negative distribution by shuffling the labels of images in a batch (Fig. 2a). Up to the chance that the shuffled labels coincide, the negative distribution is the unconditional data distribution when the number of classes are large. The unconditional distribution is what CFG subtracts in Ho & Salimans (2022). The two target distribution give interpolants

$$I_t^\pm = \alpha_t I_0 + \beta_t I_1^\pm, \qquad I_1^\pm \sim \mu_1^\pm(\cdot \mid c). \tag{6}$$



(a) Positive and negative pairs

(b) contrastive current for the label *bear* (schematic)

Figure 2: **Illustration of negative distribution and contrastive current** (a) Negative samples are obtained by shuffling the labels of images in a batch (b) Currents towards samples of the wrong class are being subtracted, thus the flow in magenta points away from images of wrong class. Current towards the correct image is amplified in green.

The currents $J^{\pm}(x)$, occupation densities $\nu^{\pm}(x)$ and velocities $b^{\pm} = J^{\pm}/\nu^{\pm}$ are then defined by equation (1). The *contrastive Beckmann Current* at guidance strength ω is defined by combining currents $J^{\pm}(x)$ following Remark 1

Definition 2. *Given currents $J^{\pm}(x)$ towards $\mu_1^{\pm}(x \mid c)$ respectively, the contrastive Beckmann Current is*

$$J^{\omega} = (1+\omega)J^{+} - \omega J^{-} = (1+\omega)\,\nu^{+}b^{+} - \omega\,\nu^{-}b^{-}. \tag{7}$$

For $c =$ arctic fox, J^{ω} is the current toward arctic-fox images minus a fraction of the current toward everything else. A proxy for the target distribution that the contrastive Beckmann current transports towards is $\mu_1^{\omega}(x)$ given by the divergence of J^{ω} similar to equation (3).

Proposition 3 (Divergence Condition)**.** *Let the interpolant time t be sampled uniformly, then the contrastive current satisfies $\nabla \cdot J^{\omega} = \mu_0 - \mu_1^{\omega}$ where $\mu_1^{\omega} = (1+\omega)\mu_1^{+} - \omega\mu_1^{-} = \sigma_+^{\omega} - \sigma_-^{\omega}$ is a contrastive measure of unit mass with mutually singular positive and negative parts $\sigma_{\pm}^{\omega}$ which are supported on $\Omega_{\pm}^{\omega}$ respectively.*

The proof follows from divergence equation (3) applied to $J^{\pm}(x)$ respectively (Appendix A). The distribution μ_1^{ω} is the additive counterpart of the distribution $p(x \mid c)^{1+\omega}p(x)^{-\omega}$ that CFG targets heuristically. However μ_1^{ω} is not the exact distribution that CBM samples. In Proposition 5, we show CBM is theoretically robust by avoiding sampling the region on which μ_1^{ω} is negative.

3.2 Learning the contrastive map

Contrastive Beckmann Map $T^{\omega}(x)$ is constant along the flow lines of $J^{\omega}(x)$. We learn T^{ω} with two transport loss terms for the positive and negative distributions. The negative distribution term has its stop-grad target sign flipped. The gradient from the negative distribution term inherits a minus sign from the flip. The boundary loss on positive data only:

$$\begin{aligned}\mathcal{L}^{\omega}_{\text{CBM-T}}(\theta) =& (1+\omega)\underbrace{\mathbb{E}_{t,I_0,I_1^{+}}\left\|T_\theta^{\omega}(I_t^{+}) - \text{sg}\left(T_\theta^{\omega}(I_t^{+}) + \dot{I}_t^{+}\cdot\nabla T_\theta^{\omega}(I_t^{+})\right)\right\|^2}_{\mathcal{L}_{\text{tr}}^{+}} \\ &+ \omega\underbrace{\mathbb{E}_{t,I_0,I_1^{-}}\left\|T_\theta^{\omega}(I_t^{-}) - \text{sg}\left(T_\theta^{\omega}(I_t^{-}) - \dot{I}_t^{-}\cdot\nabla T_\theta^{\omega}(I_t^{-})\right)\right\|^2}_{\mathcal{L}_{\text{tr}}^{-}} + \lambda_\partial\mathcal{L}_\partial(\theta),\end{aligned} \tag{8}$$

T_θ^{ω} requires a label input. For $T_\theta^{\omega}(I_t^{+})$, the input label is the correct label of image I_1^{+}. For $T_\theta^{\omega}(I_t^{-})$, the label is the wrong label of image I_1^{-}. Setting $\omega = 0$ recovers (5). We call training with (8) CBM-Transport (CBM-T). Details of the training CBM-Transport is in Algorithm 1. The map $T^{\omega}(x)$ at which gradient of loss (8) vanishes is characterized by:

Proposition 4 (Conservation along the contrastive current)**.** *Let interpolant time t be sampled uniformly. The population stop-gradient update of* (8) *vanishes at T^ω if and only if $J^\omega(x) \cdot \nabla T^\omega(x) = 0$ almost everywhere and $T^\omega(x) = x$ for μ_1^+-almost every x.*

The next result characterizes the output distribution of the ideal endpoint map: under its assumptions, the map assigns no probability mass to the negative region Ω_-^ω of the contrastive target.

Proposition 5 (Exclusion from the negative region)**.** *Suppose $\mu_1^\pm$ satisfies the geometric and codimension assumptions of Lee et al. (2026, Assumption 1 and Appendix A) on a shared submanifold $\mathcal{M}$. Then the map $T^\omega(x)$ in Proposition 4 pushes μ_0 to $q^\omega = T^\omega_\# \mu_0$ satisfies $q^\omega(A) \le \sigma_+^\omega(A)$ for every measurable A, $q^\omega(\Omega_+^\omega) = 1$ and $q^\omega(\Omega_-^\omega) = 0$.*

Proposition 5 establishes exclusion from the negative region of the contrastive target. We demonstrate Proposition 5 on a toy example in Appendix A. We investigate empirically how guidance strength affects precision and recall in §4.2. The propositions characterise the idealised objective (8). Implementation uses non-uniform time schedulers $t \sim p(t)$ for which the corresponding conservation condition is given in Remark 7. There, the optimal $T^\omega(x)$ is conserved along a reweighted current $J_p^\omega(x)$. As in BTM, our implementation further uses adaptive per-sample weights and an anchor time as in Appendix C.

3.3 CBM-Endpoint

CBM-Transport imposes distribution contrast on the current. We further introduce it on the outputs $T(x)$, by pushing the prediction $T(x \mid c)$ for the true label away from the prediction for a wrong label c^- at the same input x:

$$\mathcal{L}_{\mathrm{SE}}(\theta) = \mathbb{E}_{t, I_0, I_1^+} \left\| T_\theta(I_t^+, c) - \mathrm{sg}\left(2T_\theta(I_t^+, c) - T_\theta(I_t^+, c^-)\right) \right\|^2. \tag{9}$$

The gradient of $\mathcal{L}_{\mathrm{SE}}$ is the negative of the gradient of $\Phi(\theta) = \mathbb{E}\|T_\theta(I_t^+, c) - \mathrm{sg}(T_\theta(I_t^+, c^-))\|^2$. A descent step on $\mathcal{L}_{\mathrm{SE}}$ is an ascent step on Φ, pushing the arctic-fox prediction away from the weasel prediction. The transport and boundary losses keep the push bounded. We apply $\mathcal{L}_{\mathrm{SE}}$ only for $t \le 0.5$, where the input is still mostly noise. The full CBM-Endpoint training loss is

$$\mathcal{L}_{\text{CBM-E}} = \mathcal{L}_{\mathrm{tr}} + \lambda_\partial \mathcal{L}_\partial + \lambda_{\mathrm{SE}} \mathcal{L}_{\mathrm{SE}}. \tag{10}$$

CBM-E is stable empirically over all training epochs (§4.4) and can be used throughout training.

3.4 Training recipe, cost and sampling

The two losses (8) and (10) play different roles (§4.4). CBM-Endpoint is safe to keep on from the first step. CBM-Transport gives the largest gains as a short dose on a trained map but gives some of them back if left on. We therefore train CBM-E from scratch and finish with a short CBM-T dose. To improve an existing BTM checkpoint, CBM-T alone suffices. The negative branch in both CBM-T and CBM-E makes a training step cost about $1.6\times$ that of BTM (Table 6). At matched compute, the CBM method brings down FID much further than native BTM as in Figure 6.

The alternative to learning guidance is to apply it when sampling, by extrapolating away from the output under a wrong label, $X = (1 + s)\, T_\theta(x, c) - s\, T_\theta(x, c^-)$, with two network calls. Adding fresh noise back to X and repeating gives a multistep sampler (Algorithm 3, Appendix B). We use the inference time guidance method as the baseline for the learnt BTM as discussed in section (§4.3).

4 Experiments

We showcase the effect of guidance in Section 4.2 where contrastive guidance with CBM-Transport breaks through the plateau of the converged BTM chekpoint in Lee et al. (2026). Section 4.2 isolates the effect of guidance. Section 4.3 answers whether learning guidance beats applying guidance at sampling time. Section 4.4 shows contrastive guidance can not only break the plateau but also speeds up training from scratch. Section 4.5 places CBM among other one-step generators.

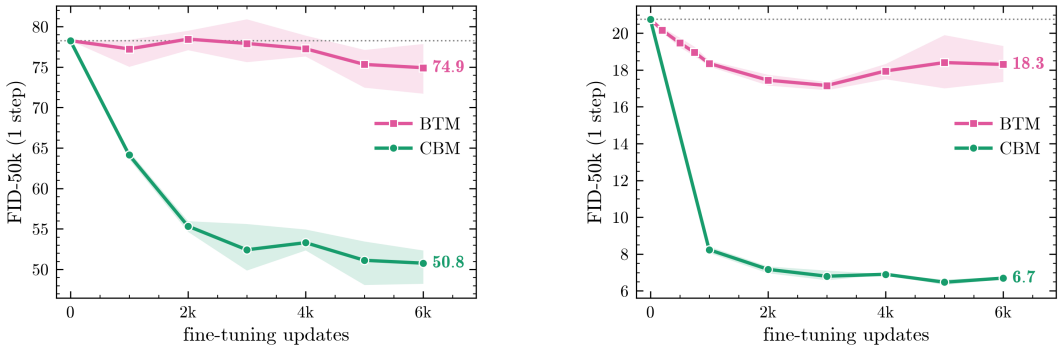
Table 1: **Effect of guidance scale ω.** DiT-XL/2 fine-tuned from the converged BTM checkpoint for 6,000 updates. One-step FID, precision and recall, mean of three seeds and range in brackets. The null control is CBM-Transport with each label rolled together with its image. Fig. 5 is the full sweep.

Arm	FID ↓	Precision ↑	Recall ↑
BTM, trained longer	18.3 [17.4, 19.3]	48.2	67.2
Null control (no mismatched pairs)	18.7 [17.2, 19.4]	45.2	67.9
CBM, $\omega = 0.5$	9.0 [8.7, 9.4]	63.1	55.5
CBM, $\omega = 1$	**6.7** [6.7, 6.8]	70.5	48.1
CBM, $\omega = 1.5$	7.4 [7.1, 7.6]	**71.9**	43.0
CBM, $\omega = 2$	8.4 [8.3, 8.5]	69.5	42.0
Real data (two halves)	–	72.0	72.0

4.1 Setup

Our main experiments are on ImageNet 256×256, and we encode the images in the latent space of SD-VAE (Rombach et al., 2021). To check scalability, we use two model sizes, DiT-B/2 (130M parameters) and DiT-XL/2 (675M) (Peebles & Xie, 2023). We also demonstrate on CIFAR-10, as a secondary dataset. Every model is one-step generators. The standard evaluation is FID (Heusel et al., 2017) using ADM's reference protocol Dhariwal & Nichol (2021), and precision/recall (Kynkäänniemi et al., 2019), and NFE counts network calls. Details are in Appendix C.

4.2 Contrastive guidance breaks BTM through its plateau



(a) DiT-B/2 (b) DiT-XL/2

Figure 3: **CBM keeps improving FID where continued BTM training plateaus.** Both arms start from the same BTM checkpoint and differ only in the loss: (a) an intermediate DiT-B/2 checkpoint, (b) the final DiT-XL/2 checkpoint. Switching to CBM-Transport drops FID within a few thousand updates. Mean of three seeds, bands show the range.

Continuing DiT-XL/2 from a FID 20 checkpoint right before the plateau with the same loss barely helps (Fig. 3b), in line with the FID reported by Lee et al. (2026). If switch the loss to CBM-Transport keeping everything else the same, FID falls steeply in a few thousand updates. Similar drops is seen for an intermediate DiT-B/2 checkpoint (Fig. 3a).

Two controls rule out that CBM helps by training more, or through a larger batch (Table 1). The null control keeps the extra branch with two batches going into two transport losses $\mathcal{L}_{\text{tr}}^{+}$ and $\mathcal{L}_{\text{tr}}^{-}$ but rolls each label together with its image, so no pair is mismatched; it ends at 18.7. Setting $\omega = 0$ is another control removing $\mathcal{L}_{\text{tr}}^{-}$ and ends at 18.3 with modest improvements over the starting checkpoint. Only when the model is trained against images with the wrong label does quality improve substantially. All other parameters are hold equal between CBM and BTM.

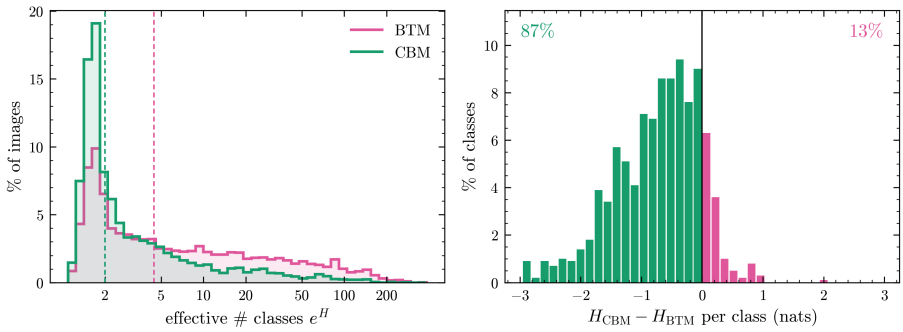


Figure 4: **CBM sharpens classifier predictions.** A pretrained ConvNeXt classifier on one-step samples from BTM and CBM (DiT-XL/2). Left: the effective number of classes e^H of each sample, where H is the entropy of the classifier's predicted distribution; dashed lines mark the medians. Right: the change in mean entropy per prompt class; CBM is sharper for 87% of classes.

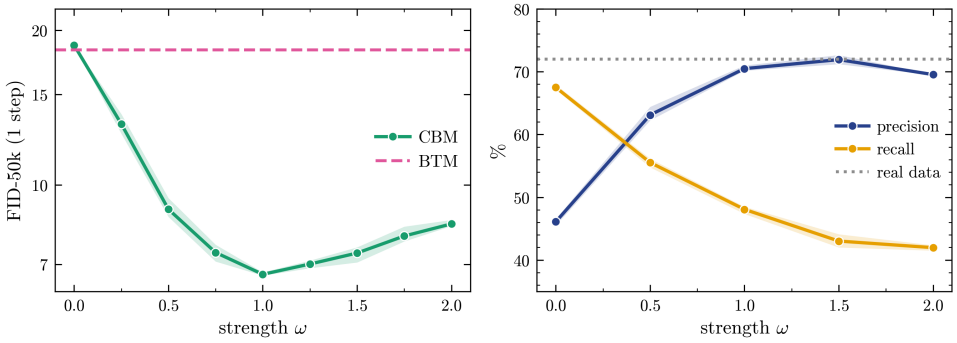


Figure 5: **Effect of guidance** ω DiT-XL/2 fine-tuned from the converged BTM checkpoint. Left: FID vs ω. The dashed line is BTM trained for the same number of updates, and $\omega = 0$ is the CBM recipe without the contrastive term. Right: precision and recall against ω. The dotted line is real data.

We measure class sharpness using a pretrained ConvNeXt classifier Liu et al. (2022). Let H denote the entropy of its predicted class distribution for a sample. Then e^H is the effective number of classes in that distribution, shown in Fig. 4. The median falls from 4.4 for BTM to 2.0 for CBM, the tail of samples spread over tens to hundreds of classes thins out, and mean entropy falls in 87% of the 1000 prompt classes, a sign of class sharpening. Figure 1 visualises class sharpening.

We observe higher precision and lower recall under CBM guidance. As ω grows, precision reaches the real-data level at $\omega = 1.5$ with recall falling. so ω sets the balance between precision and recall as CFG's guidance scale does, and ω requires tuning.

4.3 Learning guidance beats applying it

Table 2 evaluates the CBM checkpoint with one-step FID 6.7 and a postprocessed checkpoint with FID 5.53. A natural question is if inference time guidance elicits similar performance without training. Inference time guidance needs an analogy to integration. For BTM, this may be Algorithm 3: sample noise, and map to an endpoint. Sample a new noise, interpolate with obtained end point at some timestep t_1, map this to an endpoint and repeat to timestep t_2 and so on. This moves BTM only from 20.7 at one call to 15.2 at six (Table 2), while CBM with a single call is far below both (a classifier view is in Appendix D.3). The training and inference time methods are complementary, and enables multistep for BTM as the bottom row of table 2 where inference time guidance brings FID down further after CBM training.

Table 2: **Learned versus applied guidance.** DiT-XL/2, FID against network calls. All three checkpoints are sampled with the guided sampler of Algorithm 3, which extrapolates away from the output under a mismatched label and re-noises between steps. A guided step counts as two calls.

Network calls	1	2	4	6
BTM, guidance applied at sampling time	20.70	18.89	17.09	15.15
CBM, plain checkpoint	6.7	5.69	5.53	5.51
CBM, postprocessed	5.53	5.10	5.01	**4.92**

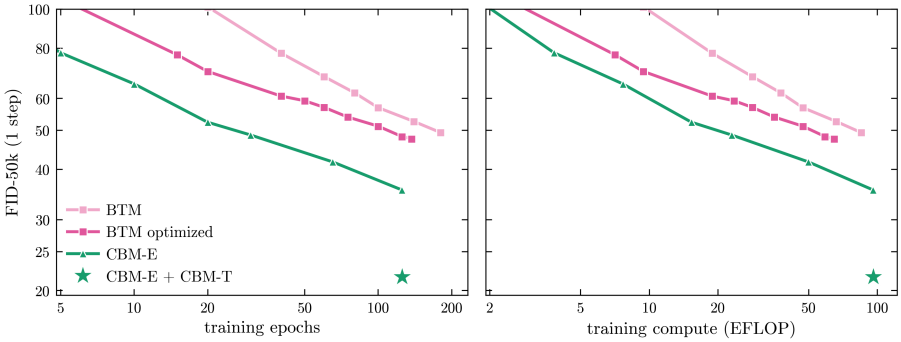


Figure 6: **Training from scratch**, DiT-B/2: one-step FID against epochs (left) and compute (right). BTM with the original recipe and our optimized recipe. CBM has CBM-Endpoint training throughout. Star: a half-epoch CBM-Transport dose at around 150 epochs reaches FID 22.

4.4 Contrastive guidance speeds up training

CBM-Endpoint is the stable pretraining loss and is used from the first step (Fig. 6). It reaches the FID of our optimized BTM recipe at 60 epochs with about $3.5\times$ fewer epochs and about half the compute, even though each CBM step costs $1.6\times$ more, and the final FID of the original recipe (about 49 at 180 epochs) with roughly $6\times$ fewer epochs. A half-epoch CBM-Transport dose at around 150 epochs then adds another large drop (star).

The two losses behave differently when left on (Fig. 9, Table 11). One epoch after branching, while both show initial dropping, CBM-Endpoint keeps decreasing steadily. CBM-Transport climbs up after as empirically observed 9. CBM-Endpoint therefore seems to work best for the pretraining, and CBM-Transport as the finishing finetuning dose as shown by star in in Fig. 6.

4.5 CBM compared to SOTA one-step generators

Table 3 places CBM among one-step generators. CBM is not state of the art. MeanFlow, which learns CFG in training, reaches 3.43. CBM lowers BTM's FID threefold, and is lower than Shortcut's 10.6 FID. CBM does not distil or use additional feature network. On CIFAR-10, BTM peaks at 7.51 and gets worse with further training at every learning rate we tried; 2,500 CBM-Transport updates bring it to 5.32.

5 Related work

One-step generation. Most one-step generators start from a time-dependent flow: they learn its integral operators (Salimans & Ho, 2022; Song et al., 2023; Boffi et al., 2024; 2025), or align a student with a multistep teacher (Yin et al., 2024b;a; Salimans et al., 2024), or straighten its trajectories (Liu et al., 2023), or learn the average velocity alongside the flow (Geng et al., 2025a;b). BTM learns the map alone.

Time-independent flows and native maps. Equilibrium Matching (Wang & Du, 2025) learns a time-independent field as the gradient of an energy, but sampling is still done by multistep integration. BTM (Lee et al., 2026) gives the flow-matching-consistent version. A discrete version (Tang

Table 3: **One-step and Multistep Models on ImageNet256 and CIFAR-10 (C10).** Auxiliary: an additional velocity branch or pretrained feature encoder. Guidance is velocity-based CFG or data-based guidance. It is applied at inference for SiT and incorporated during training for the one-step methods. iMF and Drifting additionally condition on guidance strength. Data-based guidance uses feature-space repulsion in Drifting and contrastive probability currents in CBM.

	Method	NFE	Training type	Auxiliary	Guidance	FID ↓
ImageNet	SiT-XL/2 (Ma et al., 2024)	250	Flow matching	–	–	8.30
	SiT-XL/2 (Ma et al., 2024)	2×250	Flow matching	–	Velocity	2.06
	iCT (Song & Dhariwal, 2023) (number from Lee et al. (2026))	1	Consistency	–	–	30.10
	Shortcut (Frans et al., 2025)	1	Self-distillation	–	Velocity	10.60
	MeanFlow (Geng et al., 2025a)	1	Flow integration	–	Velocity	3.43
	iMF (Geng et al., 2025b)	1	Flow integration	Velocity branch	Velocity	1.72
	Drifting (Deng et al., 2026)	1	Direct map	Feature encoder	Data	1.54
	BTM, reported (Lee et al., 2026)	1	Direct map	–	–	17.58
	CBM	1	Direct map	–	Data	**6.7 / 5.5**[†]
C10	BTM	1	Direct map	–	–	7.51
	CBM (ours)	1	Direct map	–	Data	**5.32**

[†] The plain CBM result is the three-seed mean from Table 1; the postprocessed result uses the checkpoint described in Appendix C.3.

& Wang, 2026) needs several steps to reach competitive perplexity. Drifting models (Deng et al., 2026) also learn a one-step map directly, attracting samples to data and repelling them from other classes in the feature space of a large pretrained encoder, on which their performance depends.

Guidance and contrast. CFG (Ho & Salimans, 2022) and inference-time steering (Holderrieth et al., 2026; Unneberg et al., 2026) guide at sampling time, at extra cost per step. Training-time alternatives fine-tune on a reward (Domingo-Enrich et al., 2025; Potaptchik et al., 2025) or build a guided velocity into the regression target (Geng et al., 2025a). CBM needs neither a reward function nor a velocity. Attraction to the right class and repulsion from other classes, or its own generated samples, appears in drifting models (Deng et al., 2026) and in distillation (Zhang et al., 2026; Feng et al., 2026). Drifting guidance (Deng et al., 2026) uses the same additive target distribution as CBM, while retaining one-step inference through feature-space updates. Signed Rectified Flow (Liao et al., 2026) also studies contrastive targets and exclusion from their negative regions through a time-dependent flow. CBM applies the contrast to current $J^\omega = (1+\omega)J^+ - \omega J^-$ and directly learns an endpoint map through a sign-flipped stop-gradient loss whose ideal population stationary condition is $J^\omega \cdot \nabla T^\omega = 0$, requiring neither a learned velocity field nor an auxiliary feature encoder.

6 Limitations and Conclusion

Limitations. CBM trades diversity for fidelity, as guidance does. Recall falls with ω. CBM-Transport needs dosing, since left on it gives back part of its gain. Negatives require a conditioning signal, we use class labels, and mismatched captions would be the analogue for text conditioning, which we have not tested. Our theory holds at the population level and does not cover the practical modifications of the recipe. Each model size uses one BTM parent, and pretraining curves use one seed in these plots, while observed saw the same tendencies consistently during training. CBM still trails MeanFlow with CFG, and our best number 5.5 uses postprocessing.

Conclusion. BTM is defined by its current, and currents add where velocities do not. Contrastive guidance with Contrastive Beckmann Maps subtracts the current towards the negative distribution from the current towards the correct distribution. CBM tackles the challenge of guiding BTM which has no velocity during training. Guidance is learned in training and free at sampling. CBM needs no teacher, reward function or feature network, changes only which image–label pairs the loss sees, and plugs into any existing BTM checkpoint. Even in this first form, CBM cuts the FID of a converged BTM model threefold in only about 15% of an epoch. CBM suggests that much of the gap to the best one-step models may be closed by a training time guidance signal, and the gap is not intrinsic to training the one-step map end-to-end. BTM is a recent development and we see this work as a stepping stone for principled direct training of one-step generators.

ETHICS STATEMENT

This work studies the optimization of a generative model on ImageNet and CIFAR-10 and introduces no new data collection and no human subjects. It inherits the distributional biases of these datasets and the dual-use profile of class-conditional image generators, and it neither adds to nor mitigates either.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix B specify the losses and the sampler, Appendix C every recipe, starting checkpoint and evaluation protocol, and Appendix A the proofs with their assumptions. Matched fine-tuning arms start from identical parameters with fresh optimizer state and share a generation stream, and setting $\omega = 0$ recovers the base BTM code path. Fine-tuning results report three seeds with their range; pretraining curves use one seed. Source code and model weights will be released after cleaning the codebase.

AI USE STATEMENT

Generative AI tools assisted with implementation of methods in code, supervising experiments designed by humans, and writing a draft of the paper, and to create tables, and writing implementation details. It helped as sparring partner for designing the toy example for demonstrating proposition 5. The idea and methods were developed by humans. The authors are responsible for verifying the final arguments, implementation and reported results.

REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Sina Alemohammad, Zhangyang Wang, and Richard G. Baraniuk. Neon: Negative extrapolation from self-training improves image generation, 2025. URL https://arxiv.org/abs/2510.03597.

Martin Beckmann. A continuous model of transportation. *Econometrica*, 20(4):643–660, 1952. doi: 10.2307/1907646. URL https://www.jstor.org/stable/1907646.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *arXiv preprint arXiv:2406.07507*, 2024. doi: 10.48550/arXiv.2406.07507. URL https://arxiv.org/abs/2406.07507.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1121. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/2ff3605c9e442d55156d99fe4f5f228d-Abstract-Conference.html.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026. doi: 10.48550/arXiv.2602.04770. URL https://arxiv.org/abs/2602.04770.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/852f50969a9e523ec41d26f2f68bd456-Abstract-Conference.html.

Lan Feng, Wuyang Li, Éloi Zablocki, Matthieu Cord, and Alexandre Alahi. Representation distribution matching for one-step visual generation. *arXiv preprint arXiv:2607.02375*, 2026. doi: 10.48550/arXiv.2607.02375. URL https://arxiv.org/abs/2607.02375.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/559a0998fab1d19b80e7e43a5852401c-Abstract-Conference.html. arXiv:2410.12557.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-2534. URL https://papers.neurips.cc/paper_files/paper/2025/hash/6d13e085b79d454da5910e4ca82a3d9d-Abstract-Conference.html. arXiv:2505.13447.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.02012*, 2025b. doi: 10.48550/arXiv.2512.02012. URL https://arxiv.org/abs/2512.02012.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. doi: 10.48550/arXiv.2207.12598. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Peter Holderrieth, Douglas Chen, Luca Eyring, Ishin Shah, Giri Anantharaman, Yutong He, Zeynep Akata, Tommi Jaakkola, Nicholas Matthew Boffi, and Max Simchowitz. Diamond maps: Efficient reward alignment via stochastic flow maps. *arXiv preprint arXiv:2602.05993*, 2026. doi: 10.48550/arXiv.2602.05993. URL https://arxiv.org/abs/2602.05993.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models, 2019. URL https://arxiv.org/abs/1904.06991.

Cheuk-Kit Lee, Florentin Coeurdoux, Yuyuan Chen, Sophia Tang, Peter Potaptchik, Yilun Du, Michael S. Albergo, and Eric Vanden-Eijnden. Beckmann transport models: From autonomous flows to one-step maps. *arXiv preprint arXiv:2608.01692*, 2026. doi: 10.48550/arXiv.2608.01692. URL https://arxiv.org/abs/2608.01692v3. Version 3.

Runlong Liao, Baiyu Su, Lizhang Chen, and Qiang Liu. Signed rectified flow: Negativity-controlled generation. *arXiv preprint arXiv:2607.18516*, 2026. doi: 10.48550/arXiv.2607.18516. URL https://arxiv.org/abs/2607.18516.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11976–11986, 2022. doi: 10.1109/CVPR52688.2022.01167. URL https://openaccess.thecvf.com/content/CVPR2022/html/Liu_A_ConvNet_for_the_2020s_CVPR_2022_paper.html.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024. doi: 10.48550/arXiv.2401.08740. URL https://arxiv.org/abs/2401.08740.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Peter Potaptchik, Cheuk-Kit Lee, and Michael S. Albergo. Tilt matching for scalable sampling and fine-tuning. *arXiv preprint arXiv:2512.21829*, 2025. doi: 10.48550/arXiv.2512.21829. URL https://arxiv.org/abs/2512.21829.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. URL https://arxiv.org/abs/2112.10752.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Multistep distillation of diffusion models via moment matching. *arXiv preprint arXiv:2406.04103*, 2024. doi: 10.48550/arXiv.2406.04103. URL https://arxiv.org/abs/2406.04103.

Filippo Santambrogio. *Optimal Transport for Applied Mathematicians*. Birkhäuser, 2015. doi: 10.1007/978-3-319-20828-2. URL https://link.springer.com/book/10.1007/978-3-319-20828-2.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models, 2023. URL https://arxiv.org/abs/2310.14189.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Sophia Tang and Shiyi Wang. Discrete Beckmann transport models for one-step language modeling and reasoning. *arXiv preprint arXiv:2609.15903*, 2026. doi: 10.48550/arXiv.2609.15903. URL https://arxiv.org/abs/2609.15903.

Ulrik Unneberg, Shiyi Wang, Dinko Franceschi, Boris Kozinsky, Ellen D. Zhong, and Michael S. Albergo. Scalable inference-time steering in molecular design with multimodal meta flow maps. To appear, 2026.

Runqian Wang and Yilun Du. Equilibrium matching: Generative modeling with implicit energy-based models. *arXiv preprint arXiv:2510.02300*, 2025. doi: 10.48550/arXiv.2510.02300. URL https://arxiv.org/abs/2510.02300.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024a. doi: 10.48550/arXiv.2405.14867. URL https://arxiv.org/abs/2405.14867.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024b. doi: 10.1109/CVPR52733.2024.00632. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yin_One-step_Diffusion_with_Distribution_Matching_Distillation_CVPR_2024_paper.html. arXiv:2311.18828.

Yuan Zhang, Chenyi Li, Haodong Yu, Guoqing Ma, Jiajun Zha, Yuanming Yang, Bo Wang, Wei Tang, Wenbo Li, Haoyang Huang, and Nan Duan. Teacher-feature drifting: One-step diffusion distillation with pretrained diffusion representations, 2026. URL https://arxiv.org/abs/2605.07327.

A PROOFS

Throughout, $t \sim \mathcal{U}[0,1]$, $I_0 \sim \mu_0$ is independent of $I_1^\pm \sim \mu_1^\pm(\cdot \mid c)$, the label c is fixed and suppressed, and equations involving the singular measures $\mu_1^\pm$ hold in the sense of distributions.

Remark 6. *Since $\rho_t(x)b_t(x) = \mathbb{E}[\dot{I}_t\,\delta(x - I_t)]$, the current can be written as*

$$J(x) = \int_0^1 \iint (\dot{\alpha}_t x_0 + \dot{\beta}_t x_1)\,\delta(x - \alpha_t x_0 - \beta_t x_1)\,\mu_0(dx_0)\,\mu_1(dx_1)\,dt,$$

which is linear in μ_1; the same holds for ν with $\dot{I}_t$ replaced by 1. If $\mu_1 = \sum_k a_k \mu_1^{(k)}$, then $b = \sum_k a_k \nu^{(k)} b^{(k)} / \sum_k a_k \nu^{(k)}$, which differs from $\sum_k a_k b^{(k)}$ unless the $\nu^{(k)}$ are proportional.

Proposition 3. Each interpolant $I_t^\pm$ has densities $\rho_t^\pm$ obeying $\partial_t \rho_t^\pm + \nabla\cdot(\rho_t^\pm b_t^\pm) = 0$ with $\rho_0^\pm = \mu_0$ and $\rho_1^\pm = \mu_1^\pm$. Integrating over t gives $\nabla \cdot J^\pm = \mu_0 - \mu_1^\pm$, and therefore

$$\nabla \cdot J^\omega = (1+\omega)(\mu_0 - \mu_1^+) - \omega(\mu_0 - \mu_1^-) = \mu_0 - \mu_1^\omega.$$

The total mass of μ_1^ω is $(1+\omega) - \omega = 1$. □

Proposition 4. Perturb T to $T + \varepsilon\eta$ with η smooth and compactly supported. Write $\Delta^\pm = \dot{I}_t^\pm \cdot \nabla T(I_t^\pm)$. The stopped targets do not move, so the per-sample losses become $\|\varepsilon\eta(I_t^+) - \Delta^+\|^2$ on the positive branch, $\|\varepsilon\eta(I_t^-) + \Delta^-\|^2$ on the negative branch and $\|T(I_1^+) + \varepsilon\eta(I_1^+) - I_1^+\|^2$ on the boundary. Their derivatives at $\varepsilon = 0$ give the population stop-gradient update in direction η:

$$G(\eta) = -2(1+\omega)\,\mathbb{E}\langle\Delta^+, \eta(I_t^+)\rangle + 2\omega\,\mathbb{E}\langle\Delta^-, \eta(I_t^-)\rangle + 2\lambda_\partial\,\mathbb{E}\langle T(I_1^+) - I_1^+, \eta(I_1^+)\rangle. \tag{11}$$

Conditioning on $I_t^\pm = x$ and integrating over t, $\mathbb{E}\langle\Delta^\pm, \eta(I_t^\pm)\rangle = \int\langle J^\pm(x)\cdot\nabla T(x), \eta(x)\rangle\,dx$, so

$$G(\eta) = -2\int \big\langle J^\omega(x)\cdot\nabla T(x), \eta(x)\big\rangle\,dx + 2\lambda_\partial \int \big\langle T(x) - x, \eta(x)\big\rangle\,\mu_1^+(dx). \tag{12}$$

The first term is an integral against Lebesgue measure and the second against μ_1^+, which is concentrated on the Lebesgue-null manifold $\mathcal{M}$. Hence $G(\eta) = 0$ for every η if and only if $J^\omega \cdot \nabla T = 0$ almost everywhere and $T(x) = x$ for μ_1^+-almost every x. and $T(x) = T^\omega(x)$ as described in the Proposition. □

Remark 7 (The proof of proposition 4 holds for training loss (8) with time schedulers)**.** *Note that training loss* (8) *by drawing t from a density $p(t)$ instead of uniformly. We may define a $p(t)$-reweighted current $J_p^\omega = \int_0^1 p(t)\rho_t b_t\,dt$. The same argument to the proof above gives conservation of the ground truth $T_p^\omega(x)$ corresponding to training with time scheduling is conserved along the time reweighted current $J_p^\omega(x)$ with $J_p^\omega(x)\cdot\nabla T_p^\omega(x) = 0$.*

Proposition 5. We use Gaussian source $\mu_0 = \mathcal{N}(0, \mathbb{I}_d)$ and straight interpolants $I_t^\pm = (1-t)I_0 + tI_1^\pm$ as stated in the main text. As in BTM (Lee et al., 2026, Appendix A), assume that $\mathcal{M} \subset \mathbb{R}^d$ is a compact embedded C^3 manifold without boundary, of codimension at least two, and that $\mu_1^\pm = \rho_1^\pm\,d\,\mathrm{vol}_\mathcal{M}$ have C^1 probability densities, here allowed to be nonnegative. Fix $\omega \geq 0$.

To establish the result, first observe that Proposition 3 gives

$$\nabla \cdot J^\omega = \mu_0 + \sigma_-^\omega - \sigma_+^\omega.$$

Thus μ_0 and σ_-^ω supply current, while σ_+^ω absorbs it. This suggests that the endpoint mass supplied by μ_0 is at most σ_+^ω.

To make this rigorous, we first establish endpoint convergence using BTM's flow-Jacobian argument. Let X_s be the flow map on $D = \mathbb{R}^d \setminus \mathcal{M}$ of

$$b^\omega = \frac{J^\omega}{\tilde{\nu}}, \qquad \tilde{\nu} = (1+\omega)\nu^+ + \omega\nu^- > 0,$$

and let $\tau(x)$ be its hitting time. Following (Lee et al., 2026) the straight-interpolant geometry gives

$$b^\omega \in C^1(D), \quad \int_D \tilde{\nu}\, dx = 1 + 2\omega, \quad \|J^\omega\|_{L^1} \le (1+\omega)\mathbb{E}|I_1^+ - I_0| + \omega\mathbb{E}|I_1^- - I_0| < \infty.$$

Since $\nabla \cdot (\tilde{\nu} b^\omega) = \rho_0$ on D, we apply BTM's backward-flow identity (Lee et al., 2026, Appendix A, Eq. (27)) . Integrating it along the backward trajectories, discarding the nonnegative terminal term, and changing variables yields

$$\int_D \left[\tau(x) + \int_0^{\tau(x)} |b^\omega(X_s(x))|\, ds \right] d\mu_0(x) \le \int_D (\tilde{\nu} + |J^\omega|)\, dx < \infty. \tag{13}$$

Consequently, μ_0-almost every trajectory has finite hitting time and finite length, so its endpoint $T^\omega(x) = \lim_{s \uparrow \tau(x)} X_s(x)$ exists. This endpoint belongs to $\mathcal{M}$: otherwise local ODE continuation in D would contradict maximality. Hence $q^\omega = T^\omega_\# \mu_0$ is well defined and $q^\omega(\mathcal{M}) = 1$.

It remains to identify where this endpoint mass can be absorbed. Set $r = \mathrm{dist}(\cdot, \mathcal{M})$ and $S_\epsilon = \{r = \epsilon\}$, the boundary of the ϵ-neighborhood of $\mathcal{M}$. Let p be nearest-point projection and $n_\epsilon = \nabla r$ the normal pointing away from $\mathcal{M}$. For $r(x) > \epsilon$, let $H_\epsilon(x)$ be the first hitting point on S_ϵ. For every nonnegative $f \in C(\mathcal{M})$, the weighted flow-tube calculation (Lee et al., 2026, Appendix A, Eqs. (26)–(28)) gives, for almost every sufficiently small ϵ,

$$\begin{aligned} \int_{\{r>\epsilon\}} f(p(H_\epsilon(x)))\, d\mu_0(x) &\le \int_{S_\epsilon} f(p(y))[-J^\omega(y) \cdot n_\epsilon(y)]_+\, d\mathcal{H}^{d-1}(y) \\ &\xrightarrow{\epsilon \downarrow 0} \int_{\mathcal{M}} f\, d\sigma_+^\omega. \end{aligned} \tag{14}$$

The inequality drops the nonnegative incoming flux at the back of each first-entry tube. The limit follows by applying BTM's uniform near-manifold expansion (Lee et al., 2026, Appendix A, Eqs. (39)–(41)) by linearity to $h^\omega = (1+\omega)\rho_1^+ - \omega\rho_1^-$ and taking positive parts: the leading inward normal coefficient is positive and integrates to one over each unit normal sphere, leaving $(h^\omega)_+\, d\,\mathrm{vol}_{\mathcal{M}} = d\sigma_+^\omega$. Vanishing densities are covered by adding and subtracting a strictly positive reference density.

Finally, $p(H_\epsilon(x)) \to T^\omega(x)$ for μ_0-almost every x and $\mu_0(\{r \le \epsilon\}) \to 0$. Dominated convergence in (14) therefore gives $\int f\, dq^\omega \le \int f\, d\sigma_+^\omega$. Thus $q^\omega \le \sigma_+^\omega$. Since σ_+^ω is concentrated on Ω_+^ω and $q^\omega(\mathcal{M}) = 1$, it follows that $q^\omega(\Omega_+^\omega) = 1$ and $q^\omega(\Omega_-^\omega) = 0$.

A.1 A TOY EXAMPLE OF PROPOSITION 5

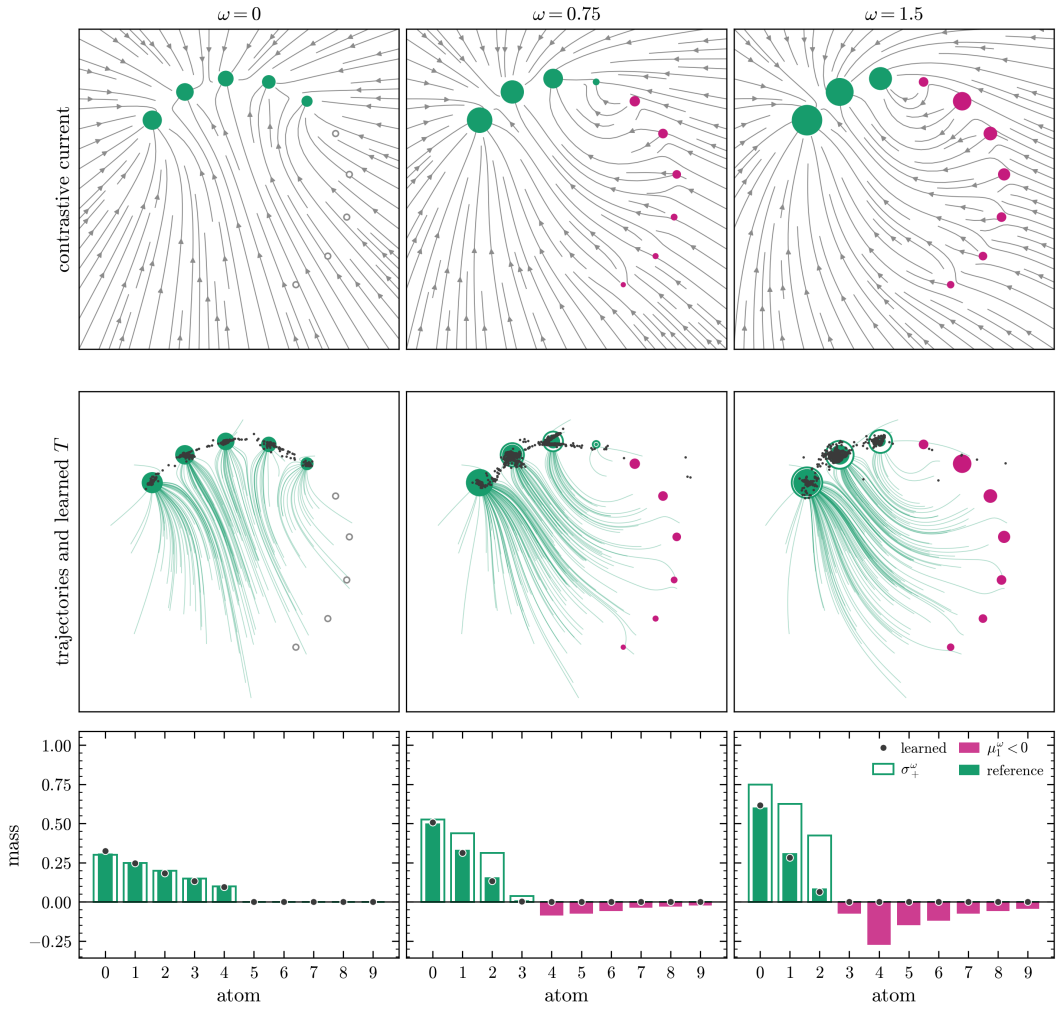


Figure 7: **Proposition 5 on a toy.** Ten atoms in the plane are the data; class c puts mass on atoms 0–4 and the other classes overlap it on atoms 2–4. Columns: guidance strength ω. *Top:* the contrastive target μ_1^ω (green: positive part σ_+^ω; magenta: negative part; area $\propto$ mass) and the flow lines of the contrastive current J^ω. Negative atoms are sources of J^ω. *Middle:* trajectories along J^ω from Gaussian source samples (green) and one-step samples $T_\theta(x_0)$ of a map trained with the CBM-Transport objective on the same sources (black). Outlines mark σ_+^ω, filled discs the mass that source trajectories deliver. *Bottom:* mass per atom. No trajectory ends on a negative atom, and every positive atom receives at most σ_+^ω, as Proposition 5 states; the learned map follows the reference.

Proposition 5 is a statement about the population objective, so we first check it where nothing is learned. We choose a problem in which the contrastive current can be evaluated exactly, integrate its flow lines, and count where the source mass ends. We then train a map with the CBM-Transport loss.

The data are $K = 10$ points $m_0, \dots, m_9 \in \mathbb{R}^2$ on an arc of radius 2.5. A set of points has codimension two in the plane, as the theory requires. The source is $\mu_0 = \mathcal{N}(0, I_2)$ and the interpolant is $I_t = (1-t)I_0 + tI_1$. Class c puts mass $p = (.30, .25, .20, .15, .10, 0, 0, 0, 0, 0)$ on the atoms; the other classes, which the shuffled labels sample, put mass $q = (0, 0, .05, .30, .35, .10, .08, .05, .04, .03)$. The two overlap on atoms 2–4. These are the ambiguous images, and on atoms 3 and 4 the other

classes dominate. The contrastive target of Proposition 3 is

$$\mu_1^\omega = \sum_k w_k \, \delta_{m_k}, \qquad w_k = (1+\omega)p_k - \omega q_k, \tag{15}$$

so atom k lies in Ω_-^ω once $w_k < 0$: atom 4 at $\omega = 0.4$ and atom 3 at $\omega = 1$. We show $\omega \in \{0, 0.75, 1.5\}$, where Ω_-^ω is empty, then atoms 4–9, then atoms 3–9.

Currents are linear in the target, so $J^\omega = \sum_k w_k J_{m_k}$, where the current towards a single atom is an explicit one-dimensional integral in t. We integrate the flow lines of J^ω from 10,000 source samples until they reach an atom. The endpoints sample $q^\omega = T^\omega_\# \mu_0$ without any training, so they test Proposition 5 directly.

We train a small network with the idealised objective, (uniform t, plain L^2 transport terms, no adaptive weights) and evaluate its one-step samples $T_\theta(x_0)$ on the same source points.

Result. Proposition 5 makes three claims, and the reference confirms each of them at every ω (Fig. 7, Table 4):

- $q^\omega(\Omega_-^\omega) = 0$: none of the 10,000 trajectories ends on a negative atom. Atom 4 is sampled by BTM at $\omega = 0$, but receives nothing at $\omega = 0.75$, and atom 3 follows at $\omega = 1.5$. The negative atoms are sources of J^ω: flow lines leave them.
- $q^\omega(A) \le \sigma_+^\omega(A)$: every positive atom receives less than its positive mass. The bound is not attained, because $\sigma_+^\omega(\mathcal{M}) = 1 + \sigma_-^\omega(\mathcal{M}) > 1$: the flow lines that leave the negative atoms also end on the positive ones and take up the rest. At $\omega = 1.5$, for example, atom 1 receives 0.311 of its 0.625.
- $q^\omega(\Omega_+^\omega) = 1$: every trajectory reaches $\mathcal{M}$ and ends on a positive atom.

At $\omega = 0$ the construction reduces to BTM, and the endpoint masses reproduce μ_1^+ within Monte Carlo error. The learned map follows the reference at every ω and places at most 0.2% of its samples on Ω_-^ω. Guidance thus removes the ambiguous images first: the atoms that the other classes dominate. This is the behaviour that Proposition 5 predicts for the between-class images on ImageNet (§4).

A.2 Details of the toy example

Fix $I_1 = m$. The interpolant $I_t = (1-t)I_0 + tm$ has density $\rho_t(x) = \varphi_{(1-t)^2}(x - tm)$, where φ_{s^2} is the density of $\mathcal{N}(0, s^2 I_2)$, and given $I_t = x$ its velocity is $\dot I_t = m - I_0 = (m-x)/(1-t)$. The current and the occupation density are therefore

$$J_m(x) = \int_0^1 \frac{m-x}{1-t}\, \varphi_{(1-t)^2}(x - tm)\, dt, \qquad \nu_m(x) = \int_0^1 \varphi_{(1-t)^2}(x - tm)\, dt. \tag{16}$$

By Remark 1, $J^\pm = \sum_k p_k J_{m_k}$ and $\sum_k q_k J_{m_k}$ respectively (likewise $\nu^\pm$), and $J^\omega = \sum_k w_k J_{m_k}$. The integrand concentrates at $1 - t \approx |x - m|$ near an atom, so we substitute $s = 1 - t$ and use the trapezoidal rule on 400 logarithmically spaced nodes $s \in [10^{-5}, 1]$. As a check of the quadrature and of Proposition 3, the flux of J_m through circles of radius 1, 2 and 4 around the origin matches $\mu_0(\text{disc}) - \delta_m(\text{disc})$ to within 2×10^{-4}.

We integrate $\dot X = J^\omega(X)/\tilde\nu(X)$ with $\tilde\nu = (1+\omega)\nu^+ + \omega\nu^- > 0$, whose flow lines are those of J^ω. We use a second-order Runge–Kutta scheme whose step moves the point by at most $\min(0.05,\, d/4)$, where d is the distance to the nearest atom. A trajectory ends at an atom once it is within 0.005 of it and moving towards it; passing close to an atom that is a source of J^ω does not count. We start from 10,000 samples of μ_0 (fixed seed). All of them end on an atom, so the assumption of Proposition 5 holds here.

$T_\theta : \mathbb{R}^2 \to \mathbb{R}^2$ is a multilayer perceptron with four hidden layers of width 256 and SiLU activations. It is trained with objective (9) in its idealised form: $t \sim \mathcal{U}[0,1]$; plain L^2 transport terms with the stop-gradient targets $\mathrm{sg}(T_\theta(I_t^\pm) \pm \dot I_t^\pm \cdot \nabla T_\theta(I_t^\pm))$, computed with a Jacobian–vector product; boundary loss on μ_1^+ with $\lambda_\partial = 1$; no adaptive weights and no anchor time. We use batches of 2048 pairs per branch and Adam with a cosine learning-rate schedule. As on ImageNet, CBM-Transport fine-tunes a converged BTM map:

Table 4: Endpoint mass per atom of the flow along J^ω (reference, 10,000 source samples) and of the learned one-step map (2,000 of the same samples), with the positive part σ_+^ω of the contrastive target. Atoms 5–9 receive no mass in any row and are omitted. A learned sample counts for an atom within distance 0.25; the remainder lies off $\mathcal{M}$ (last column). Bold: atoms in Ω_-^ω.

ω		atom 0	atom 1	atom 2	atom 3	atom 4	off $\mathcal{M}$
0	$\sigma_+^\omega = \mu_1^+$	.300	.250	.200	.150	.100	
	reference	.308	.251	.196	.144	.102	0
	learned	.324	.247	.182	.134	.096	1.8%
0.75	σ_+^ω	.525	.438	.313	.038	**0**	
	reference	.501	.333	.157	.010	**0**	0
	learned	.506	.314	.134	.003	**0**	4.4%
1.5	σ_+^ω	.750	.625	.425	**0**	**0**	
	reference	.603	.311	.086	**0**	**0**	0
	learned	.617	.282	.065	**.001**	**.001**	3.4%

At $\omega = 0$ one trajectory in 10,000 is counted on atom 6, which carries no mass there: an integration artefact at the 10^{-4} level. The negative parts are $\sigma_-^\omega(\mathcal{M}) = 0.31$ at $\omega = 0.75$ and 0.80 at $\omega = 1.5$.

- $\omega = 0$ (BTM) is trained from scratch for 40k steps at learning rate 10^{-3};
- $\omega = 0.75$ fine-tunes it for 20k steps at 3×10^{-4};
- $\omega = 1.5$ continues from the $\omega = 0.75$ map for 30k steps at 10^{-4}.

One-step samples are taken on the first 2,000 of the reference's source points.

Proposition 4 characterises the fixed point of the stop-gradient update, not whether training reaches it. In this toy it is reached only for moderate negative mass:

- Trained from scratch at $\omega > 0$, the map collapses to a near-constant function. Such a map makes both transport terms vanish.
- At $\omega = 1.5$ straight from BTM, training does not settle; raising ω in two stages does.
- In the earlier variant of the setting (below), a fine-tune from BTM runs away at $\omega = 3$ (the loss reaches 10^{12} within 2k steps) and does not settle at $\omega = 2$. The adaptive per-example weight did not stabilise these runs and degraded the one at $\omega = 1$.

The reference involves no training and confirms Proposition 5 at every ω shown.

B ALGORITHMS

Algorithms 1 and 2 give one training step of CBM-Transport and CBM-Endpoint with a linear interpolant. Negatives are formed by rolling the batch, so each label c_i is paired with the image x_{i+1}.

Algorithm 1 CBM-Transport training step

Draw data $x_1^+[1{:}B]$ with labels $c[1{:}B]$, noise $x_0[1{:}B]$ and times $t[1{:}B] \sim q$
$I_t^+ \leftarrow (1-t)x_0 + t x_1^+, \qquad \dot{I}_t^+ \leftarrow x_1^+ - x_0$
$\Delta^+ \leftarrow D_x T_\theta(I_t^+, c)\, \dot{I}_t^+$ ▷ Jacobian-vector product
$\ell^+ \leftarrow \left\| T_\theta(I_t^+, c) - \mathrm{sg}\left(T_\theta(I_t^+, c) + \Delta^+\right)\right\|^2$
$x_1^-[i] \leftarrow x_1^+[i+1]$ ▷ roll the images; labels stay
$I_t^- \leftarrow (1-t)x_0 + t x_1^-, \qquad \dot{I}_t^- \leftarrow x_1^- - x_0$
$\Delta^- \leftarrow D_x T_\theta(I_t^-, c)\, \dot{I}_t^-$
$\ell^- \leftarrow \left\| T_\theta(I_t^-, c) - \mathrm{sg}\left(T_\theta(I_t^-, c) - \Delta^-\right)\right\|^2$ ▷ flipped target
$g_{\mathrm{tr}} \leftarrow \nabla_\theta \sum_{i=1}^B \left((1+\omega)\gamma_i^+ \ell_i^+ + \omega \gamma_i^- \ell_i^-\right)$ ▷ $\gamma^\pm$: Eq. (17), or 1
$g_\partial \leftarrow \nabla_\theta \sum_{i=1}^B \left\| T_\theta(x_{1,i}^+, c_i) - x_{1,i}^+\right\|^2$
$\theta \leftarrow \mathrm{optimizer}\left(\theta,\ g_{\mathrm{tr}} + \lambda_\partial\, g_\partial\right)$

Algorithm 2 CBM-Endpoint training step

Draw data $x_1^+[1{:}B]$ with labels $c[1{:}B]$, noise $x_0[1{:}B]$ and times $t[1{:}B] \sim q$
$I_t^+ \leftarrow (1-t)x_0 + t x_1^+, \qquad \dot{I}_t^+ \leftarrow x_1^+ - x_0$
$\Delta^+ \leftarrow D_x T_\theta(I_t^+, c)\, \dot{I}_t^+$
$\ell^+ \leftarrow \left\| T_\theta(I_t^+, c) - \mathrm{sg}\left(T_\theta(I_t^+, c) + \Delta^+\right)\right\|^2$
$c^-[i] \leftarrow c[i+1]$ ▷ roll the labels
$\ell_{\mathrm{SE}} \leftarrow \left\| T_\theta(I_t^+, c) - \mathrm{sg}\left(2T_\theta(I_t^+, c) - T_\theta(I_t^+, c^-)\right)\right\|^2 \cdot \mathbf{1}[t \le 0.5]$
$g_{\mathrm{tr}} \leftarrow \nabla_\theta \sum_{i=1}^B \gamma_i^+ \ell_i^+, \qquad g_{\mathrm{SE}} \leftarrow \nabla_\theta \sum_{i=1}^B \ell_{\mathrm{SE},i}$
$g_\partial \leftarrow \nabla_\theta \sum_{i=1}^B \left\| T_\theta(x_{1,i}^+, c_i) - x_{1,i}^+\right\|^2$
$\theta \leftarrow \mathrm{optimizer}\left(\theta,\ g_{\mathrm{tr}} + \lambda_{\mathrm{SE}}\, g_{\mathrm{SE}} + \lambda_\partial\, g_\partial\right)$

Guided multistep sampling. Algorithm 3 spends extra network calls on a BTM or CBM map. Each step after the first adds fresh noise back to the current sample at re-noise time t_r and maps again, optionally extrapolating away from the output under a wrong label. The settings for Table 2 are in Appendix C.6.

Algorithm 3 Guided multistep sampling (k network calls if $s = 0$, $2k$ if $s > 0$)

Require: map $T(\cdot,\cdot)$, label c, steps k, guidance scale $s \ge 0$, re-noise time $t_r \in (0,1)$
Draw $z \sim \mathcal{N}(0, \mathbb{I})$ and set $x \leftarrow z$
for $i = 1, \dots, k$ **do**
  if $i > 1$ **then**
    Draw $z \sim \mathcal{N}(0, \mathbb{I})$ and set $x \leftarrow t_r X + (1 - t_r) z$ ▷ re-noise
  end if
  if $s > 0$ **then**
    Draw $c^- \neq c$ and set $X \leftarrow (1+s)\, T(x, c) - s\, T(x, c^-)$ ▷ two calls
  else
    $X \leftarrow T(x, c)$ ▷ one call
  end if
end for
return X

Adaptive weighting. As in BTM (Lee et al., 2026), the transport losses scale with $\|\Delta^\pm\|^2$, and we reweight each sample by

$$\gamma^\pm = \mathrm{sg}\left(\left(\|\Delta^\pm\|^2 + c\right)^{-p}\right), \tag{17}$$

with $p = 1$ and $c = 0.01$. The weight is detached from the gradient.

C EXPERIMENTAL DETAILS

C.1 COMMON SETUP

Table 5: Setup shared by all ImageNet experiments.

Data	ImageNet-1k training set (1,281,167 images) at 256×256, encoded with the SD-VAE (ft-MSE) into $32 \times 32 \times 4$ latents, scaled by 0.18215. One epoch is one pass over the training set.
Networks	DiT-B/2 (130M parameters) and DiT-XL/2 (675M), class-conditional. B/2 models are trained with class-label dropout 0.1.
B/2 training	Global batch 256 (4 GPUs $\times$ 64), so one epoch is 5,000 steps; bf16 mixed precision; no gradient clipping. Muon (momentum 0.95, Nesterov, 5 Newton–Schulz steps) for matrix parameters and AdamW ($\beta = (0.9, 0.95)$) at one tenth of the Muon learning rate for the rest.
BTM loss (B/2)	Transport and boundary losses (Eq. (5)). The transport term uses the adaptive weight of Eq. (17) and is rescaled to the gradient norm of the boundary term. Training times follow a logit-normal law with standard deviation 1 and mean 0.8 for the first 5 epochs, 0 afterwards.
Anchor time	The map is conditioned on an anchor time γ. The boundary loss holds $T(I_s, \gamma) = I_1$ for $s \geq \gamma$, and transport acts on $[0, \gamma]$. Training draws $\gamma \sim \mathcal{U}[0.6, 1]$. Samples use $\gamma = 0.8$ (ImageNet B/2), 0.9 (ImageNet XL) and 1 (CIFAR-10).
EMA	Decays 0.99, 0.999 and 0.9999 are tracked; B/2 results use 0.999.
Checkpoints	Every 2,500 steps (half an epoch) at B/2.
XL fine-tuning	16 image pairs (32 images) per update, processed in micro-batches of 2; AdamW at a constant learning rate of 3×10^{-6}; adaptive transport and boundary losses ($p = 1$, $c = 0.01$) with boundary balance 0.25; logit-normal training times (mean -1.2, standard deviation 1); EMA 0.99 tracked, raw weights evaluated.

Table 6: Training compute per image and training step (DiT-B/2), measured with PyTorch's FlopCounterMode on one GPU in bf16, forward and backward passes included. The compute axis of Fig. 6 multiplies these by the number of images seen.

Objective	GFLOP per image per step
BTM	367.9
BTM + CBM-Endpoint	597.9
BTM + CBM-Transport	597.7

The native BTM run. The B/2 experiments of Fig. 9 and Appendices D.1 and D.2 branch off one native BTM run at batch 256. Each continuation resumes from its preceding checkpoint with its optimizer state (Table 7). At epoch 35, the run branches into a cosine-cooldown continuation and a continuation without cooldown. The latter maintains the same learning rate and time-sampling distribution throughout the remainder of training; together with the shared initial trajectory, its checkpoints form the "BTM, optimized" line of Fig. 6.

Table 7: Segments of the native BTM run (DiT-B/2, batch 256). The two segments beginning at epoch 35 are separate branches.

Epochs	Learning rate (Muon)	Time law mean	Used as
0–5	10^{-3}	0.8	base at 5 epochs (Fig. 8)
5–15	10^{-3}	0	base at 10 epochs (Figs. 8, 10)
15–35	3×10^{-4}	0	base at 20 and 30 epochs (Figs. 9, 8, 10)
35–40	cosine $3 \times 10^{-4} \to 0$	0	base at 40 epochs (Figs. 9, 8, 10)
35 onward	3×10^{-4}, constant	0	no-cooldown branch (Fig. 6)

C.2 Loss recipes

Table 8: Recipes used when continuing a B/2 checkpoint. Everything not listed is as in Table 5.

Recipe	Strength	Learning rate	Other settings
BTM, continued	–	3×10^{-4}	–
CBM-Endpoint	$\lambda_{\mathrm{SE}} = 1$ (analysis), 0.5 (pretraining)	3×10^{-4} (10^{-3} early in pretraining)	Applied for $t \leq 0.5$; negative label rolled from the batch; per-sample weight $1/(\|\Delta\|^2 + c)$; strength set relative to the boundary-loss gradient.
CBM-Transport (dose)	$\omega = 1$	10^{-4}	Transport restricted to times $t \leq \gamma$; applied for half an epoch unless stated.
Null control	$\omega = 1$	10^{-4}	As CBM-Transport, but each label is rolled together with its image, so no pair is mismatched.

C.3 Fine-tuning (Fig. 3, Tables 1 and 3)

Table 9: Fine-tuning runs of Fig. 3. Within each panel, the two arms start from the same checkpoint, share every setting below and differ only in the loss. The XL arms of Table 1 and Fig. 5 use the XL start, recipe and budget.

Panel	Start	Shared recipe (both arms)	Arms	Budget, seeds
(a) B/2	DiT-B/2 BTM checkpoint after 300k steps at batch 1024 (240 epochs), followed by 2,000 updates of the shared recipe with $\omega = 0$ (FID 78.3)	16 pairs per update; adaptive transport and boundary losses ($p = 1$, $c = 0.01$), boundary balance 0.25; learning rate 2×10^{-6}	BTM: Eq. (5). CBM: Eq. (8), $\omega = 1$	6,000 updates; 3 seeds each
(b) XL	BTM checkpoint, EMA 0.9999 (FID 20.8)	XL fine-tuning recipe of Table 5 (learning rate 3×10^{-6})	BTM: Eq. (5). CBM: Eq. (8), $\omega = 1$	6,000 updates; 3 seeds each

Postprocessing checkpoint (footnote of Table 3). The method is based on (Alemohammad et al., 2025) Starting from the XL checkpoint at FID 20.8 plus 6,000 native BTM updates at learning rate 10^{-6}), we (1) run two independent CBM-Transport fine-tunes, one of 10,000 updates with an earlier version of the recipe followed by 500 updates of the final recipe (FID 6.30), and one of 6,000 updates of the final recipe (FID 6.17); (2) average their weights uniformly (FID 6.08); (3) run 200 plain-L2 self-training updates of the average on 30,000 of its own samples at learning rate 10^{-6}; and (4) extrapolate away from that self-trained copy, $\theta = \theta_{\mathrm{avg}} + 1.35\,(\theta_{\mathrm{avg}} - \theta_{\mathrm{self}})$, which gives FID 5.535 (5.521 on a second sample seed). This checkpoint is the postprocessed CBM row of Table 2. All other results use plain fine-tunes.

CIFAR-10 (Table 3). Class-conditional CIFAR-10 (32×32, pixel space), DiT-B/2 (patch 2), trained with the BTM objective (anchor time $\gamma \sim \mathcal{U}[0.6, 1]$, label dropout 0.1, adaptive transport loss $p = 1$, $c = 0.01$), Muon (momentum 0.95, Nesterov, 5 Newton–Schulz steps, AdamW side rate 10^{-4}, no weight decay), gradient clipping at 5, dropout 0.13 and global batch 512 (1,024 between 100k and 120k steps). Native BTM runs 100k steps at learning rate 10^{-3} with time-law mean -1.2, then mean $+0.8$ with a cosine tail from 115k steps. It peaks at step 125k (FID 7.509; 7.626 on a second draw) and gets worse with further training at every learning rate tried. CBM fine-tunes that peak for 2,500 steps with CBM-Transport at $\omega = 2$, learning rate 3×10^{-5}, the gradient balancer off and boundary weight 0.05, reaching 5.317 (5.358 on a second draw); three seeds of this recipe land in $[5.32, 5.39]$. FID of the EMA-0.9999 weights, one step at anchor $\gamma = 1$.

C.4 PRETRAINING (FIG. 6)

Table 10: Runs of Fig. 6, all DiT-B/2 from scratch. All lines use the same ADM reference statistics, with about 8 log-spaced evaluation points per line; CBM evaluations use 50,000 generated samples.

Line	Run
CBM	BTM with CBM-Endpoint ($\lambda_{\mathrm{SE}} = 0.5$, $t \leq 0.5$, rolled negative) from step 0, batch 256 on 16 GPUs. Time law mean 0.8 until 5 epochs, then 0. Learning rate 10^{-3}, lowered to 3×10^{-4} at 15 epochs, cosine to 0 over epochs 35–40; continued afterwards.
CBM + dose (star)	A half-epoch CBM-Transport dose (Table 8) on the CBM checkpoint at around 150 epochs.
BTM, original	The released BTM recipe at batch 1024 (Lee et al., 2026): two runs evaluated every 20 epochs (20–180), taking the better of their EMA-0.999 and EMA-0.9999 reads at each epoch.
BTM, optimized	The native BTM run of Table 7.

C.5 TWO LOSSES (FIG. 9)

Both panels start from the native BTM run (Table 7): at 20 epochs (FID 75.8) and at 40 epochs, after the learning-rate cooldown (FID 66.2). Each arm continues from that checkpoint with one recipe of Table 8, one seed, and is scored every half epoch. The combined arm spends the same three-epoch budget as CBM-Endpoint for 2.5 epochs, followed by a half-epoch CBM-Transport dose; the points after three epochs continue with CBM-Transport.

Table 11: FID of the arms of Fig. 9 after one, three and six extra epochs. The combined arm spends 2.5 epochs on CBM-Endpoint and then switches to CBM-Transport, which stays on to the end of the run: at $+3$ it has had half an epoch of it (the dose), at $+6$ three and a half. Best per column in bold.

	from 20 epochs (75.8)			from 40 epochs (66.2)		
Arm	$+1$	$+3$	$+6$	$+1$	$+3$	$+6$
BTM, continued	75.4	74.1	72.7	66.7	65.6	64.8
+ CBM-Endpoint	58.7	57.6	**55.8**	53.0	52.7	51.3
+ CBM-Transport	**53.2**	60.5	63.5	**44.0**	51.8	55.2
CBM-Endpoint, then CBM-Transport	–	**43.4**	58.0	–	**37.1**	**51.0**

C.6 GUIDED SAMPLING (TABLE 2)

All three rows of Table 2 use Algorithm 3 with the settings in Table 12. The CBM settings apply to both the plain and postprocessed checkpoints.

Table 12: Sampler settings (re-noise time t_r, guidance scale s, steps k) for Table 2. All three checkpoints use the same settings, chosen for BTM and not re-tuned for CBM. Each step re-noises the previous output to $(1 - t_r)\,\epsilon + t_r\,\hat{x}$ with fresh ϵ and maps it again; with $s > 0$ every step, the first included, is guided with a random other class and costs two network calls. At 6 calls the output is re-noised twice, first to $t_r = 0.25$ and then to 0.6.

Network calls	1	2	4	6
BTM (t_r, s, k)	(–, 0, 1)	(0.35, 0, 2)	(0.35, 0.05, 2)	(0.25→0.6, 0.1, 3)
CBM (t_r, s, k)	(–, 0, 1)	(0.35, 0, 2)	(0.35, 0.05, 2)	(0.25→0.6, 0.1, 3)

D ADDITIONAL RESULTS

D.1 SWITCHING THE LOSSES ON AT DIFFERENT POINTS OF TRAINING

Switching each loss on at five points of the native BTM run (Fig. 8, Table 13) shows the pattern of Fig. 9 throughout training. CBM-Endpoint gains the most early (31 FID at 5 epochs, 13 at 40), a half-epoch CBM-Transport dose gains about 20 at every stage from 10 epochs on, and the combination gains most everywhere (27 to 36). The CBM-Transport null control, with no mismatched pairs, gains nothing at the three points where we ran it. Figure 9 shows the different behaviour of the two losses. Experimentally, we found CBM-E to work most stable during pretraining, and CBM-T as the final finetuning. Applying the latter too long shows some overfitting-like behaviour, which degraded performance.

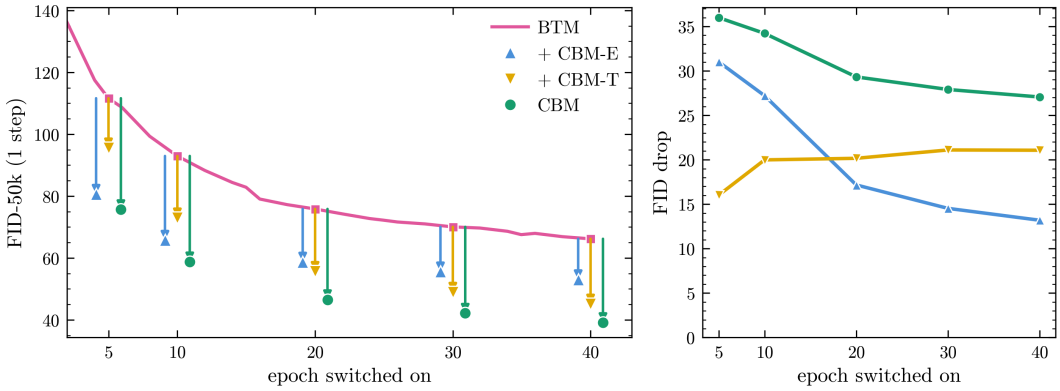


Figure 8: **Each loss switched on at five points of the native BTM run**, DiT-B/2. (a) The BTM training curve and the FID reached after one epoch of CBM-Endpoint (blue), half an epoch of CBM-Transport (orange), or both in that order (green). (b) The FID improvement over the starting checkpoint.

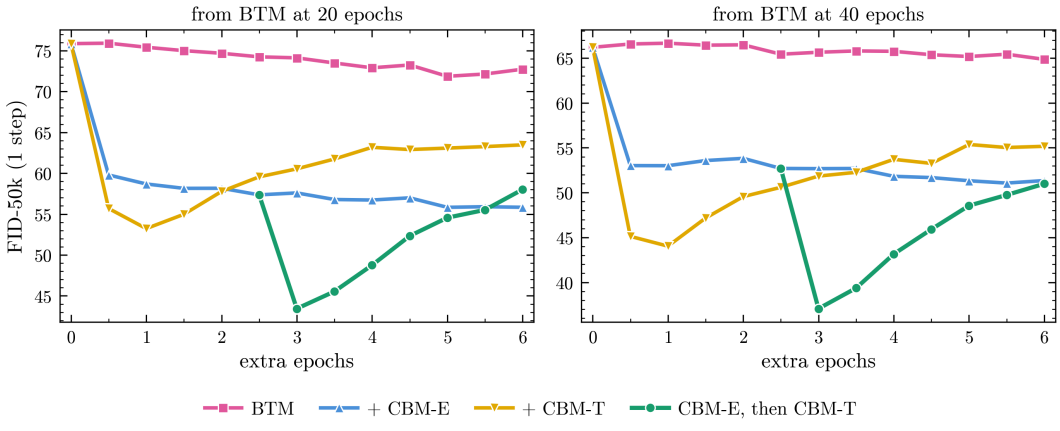


Figure 9: **Two losses, two behaviours**, DiT-B/2. Continuing native BTM checkpoints at 20 and 40 epochs (FID 75.8 and 66.2) with BTM, CBM-Endpoint (+CBM-E) or CBM-Transport (+CBM-T). CBM-E drops FID and holds it; CBM-T drops it further but climbs back if kept on. CBM-E for 2.5 epochs and then a half-epoch CBM-T dose (green) is lowest at 3 epochs, the end of the dose; the green arm then continues with CBM-T and also climbs back.

Table 13: FID change from switching each loss on at five points of the native BTM run. One seed per cell.

Switched on at (FID)	5 ep (111.6)	10 ep (93.0)	20 ep (75.8)	30 ep (70.1)	40 ep (66.2)
+ CBM-E, 1 epoch	-31	-27	-17	-15	-13
+ CBM-T, 0.5 epoch	-16	-20	-20	-21	-21
CBM-E, then CBM-T	-36	-34	-29	-28	-27
Null control (CBM-T)	–	≈ 0	≈ 0	–	≈ 0

D.2 The sampling-time baseline on DiT-B/2

Table 2 compares learned guidance with guidance applied at sampling time. To check that the sampling-time baseline is a working sampler rather than a weak one, we tune it on the DiT-B/2 native BTM run (Fig. 10).

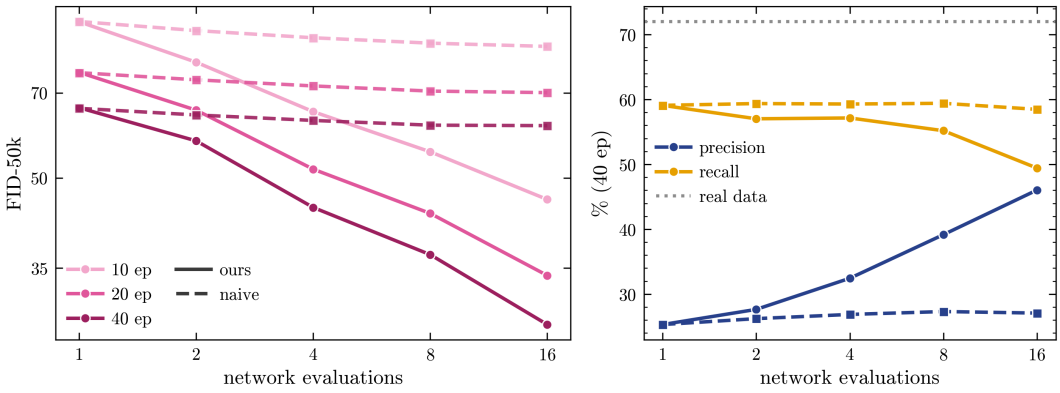


Figure 10: **More network calls, better images**, DiT-B/2 BTM. Left: FID against network calls for BTM checkpoints at 10, 20 and 40 epochs, sampled with Algorithm 3 (solid) or by re-applying the map to its own output (dashed). Right: precision and recall at 40 epochs; the dotted line is real data.

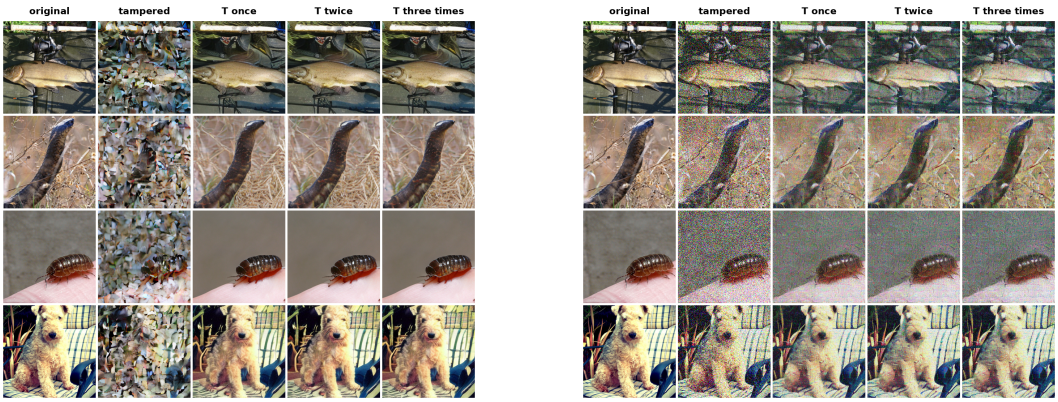
The naive way to spend more calls is to feed the map its own output again, but a map trained to jump from noise to data has little left to do once it lands on data: the map is close to idempotent (Fig. 11), and re-applying it barely changes FID (Fig. 10, dashed). Algorithm 3 instead adds fresh noise back and maps again, guided by a mismatched label. At every stage of training it improves FID steadily with more calls; at 40 epochs it goes from 66.0 with one call to 28.0 with sixteen (Table 14), while precision rises and recall falls, the signature of guidance. The baseline in Table 2 is therefore a tuned, working sampler; on the converged XL model it helps much less.

We sample the native BTM run at 10, 20 and 40 epochs on the grid $t_r \in \{0.35, 0.5\}$, $s \in \{0, 0.125, 0.25, 0.5, 1, 1.5, 2\}$ and 1, 2, 4 or 8 steps. For every budget we pick the setting with the lowest FID-10k on one set of sample seeds, and report FID on held-out seeds (Table 14).

Table 14: Selected sampler settings (re-noise time t_r, guidance scale s, steps) and held-out FID per budget of network calls, DiT-B/2 BTM.

Network calls	1	2	4	8	16
Setting (t_r, s, steps)	(–, 0, 1)	(–, 0.5, 1)	(0.35, 1, 2)	(0.5, 2, 4)*	(0.5, 2, 8)
BTM, 10 epochs	92.7	79.0	65.0	55.4	45.9
BTM, 20 epochs	75.9	65.4	51.8	43.5	34.0
BTM, 40 epochs	66.0	57.9	44.5	36.9	28.0

* $s = 1.5$ at 40 epochs.



(a) Latent-space corruption (b) Pixel-space noise

Figure 11: **The BTM map is close to idempotent.** Each row shows an original image, a corrupted version, and the map applied once, twice and three times. One application removes most of the corruption in (a) and part of it in (b); further applications change little.

D.3 Class sharpness under guided sampling

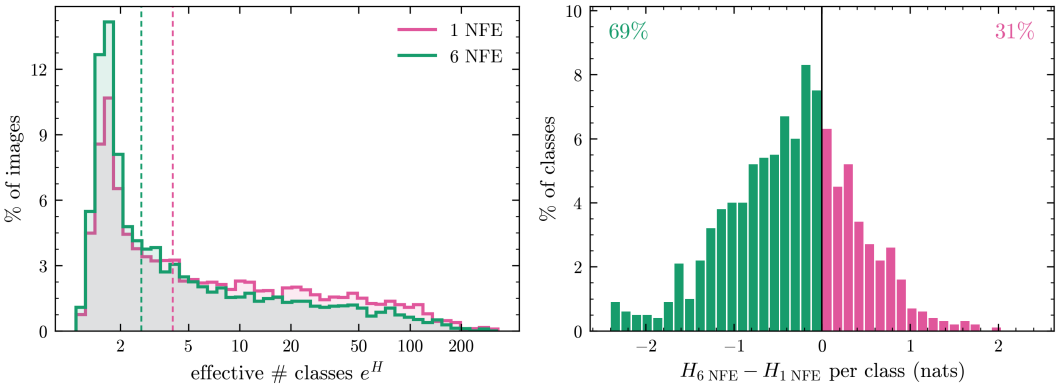


Figure 12: **Guided sampling also sharpens BTM.** The classifier analysis of Fig. 4, applied to BTM (DiT-XL/2) sampled with one call and with six calls of the guided sampler (Algorithm 3). Left: effective number of classes per sample; dashed lines mark the medians. Right: change in mean entropy per prompt class; six calls are sharper for 69% of classes.

With six calls of the guided sampler, the median effective number of classes of BTM's samples falls from 4.0 to 2.6, and mean entropy falls in 69% of prompt classes (Fig. 12): applied guidance also sharpens classifier predictions. This BTM baseline (median 4.0) comes from a different sample set from the one in Fig. 4 (median 4.4), so we do not compare the two figures directly.