GUIDANCE WITHOUT LABELS: FLOW MATCHING FROM COUPLED GAUSSIAN MIXTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow matching models are typically trained to transport a Gaussian source to the data distribution, with source and target samples paired independently. We replace the source with a fitted Gaussian mixture and couple each data point to a mixture component through an entropic optimal transport problem. We show that the resulting transport reduces to the standard Gaussian-source problem up to a guidance-like correction, which enables the reuse of a pretrained velocity field. This correction is a closed-form function of the fitted mixture plus a term that we learn with a small correction network. We test our model on published checkpoints for CIFAR-10 and CelebA-HQ. It improves FID at every step count, and on CelebA-HQ it reaches the quality of the baseline with about half the steps. The closed-form term alone, with no training, already delivers most of this improvement; the small correction network accounts for the rest.

1 INTRODUCTION

In generative modeling, one seeks to draw samples from a data distribution p_1 available only through a finite set of data points. Flow matching (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023) achieves this with a velocity field that transports a simple source distribution p_0 to p_1, learned from the conditional velocity of a single source-data pair. The framework leaves two design choices open, the source p_0 itself and the joint distribution with marginals p_0 and p_1, called the coupling. Standard practice fixes both, taking $p_0 = \mathcal{N}(\mathbf{0}, \mathbf{I})$ independent of the data distribution. That independence can make sampling inefficient, since every data point is paired with a starting point drawn without regard to where it has to end up. Near the source, a sample carries little information about where it will end up, so the field averages over many modes and the model needs many integration steps, most of them near the noise end where the trajectory decides which mode to commit to (Liu et al., 2023; Pooladian et al., 2023; Luo et al., 2026). Improving the coupling head-on is difficult. The optimal pairing between a Gaussian and the data is a continuous optimal transport map that can be neither computed nor certified, and minibatch approximations of it are biased toward independence (Pooladian et al., 2023; Tong et al., 2024).

We propose to improve both design choices at once. A source closer to the data shortens the transport, and a coupling that pairs each data point with a nearby source point shortens it further. We propose a Gaussian mixture model (GMM) with K components as the source, $p_0 = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\mu}_k, \tau^2 \mathbf{I})$, fitted to the data by maximum likelihood. GMMs have carried density estimation for half a century (Dempster et al., 1977; McLachlan & Peel, 2000). They are universal as they can approximate any continuous density to arbitrary accuracy as the number of components K grows (Bacharoglou, 2010; Nguyen et al., 2020). Sampling p_0 stays easy as it only requires one categorical draw of the component on top of the usual Gaussian one. On top of this, the categorical variable can be coupled to the data perfectly. This allows one to assign a probability distribution over the components for any data point without requiring any optimal-transport-based solvers.

Up to a change of variables and a rescaling of time, the path from a component of the GMM to the data is the standard Gaussian-source path. A velocity field pretrained with a Gaussian source can therefore be reused without any retraining. What the coupling adds is a single term: the gradient of the log-probability that the noisy image came from the drawn component. Most of it is a closed-form function of the mixture, and a light network suffices to learn the rest.

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.

The construction is reminiscent of classifier guidance, in which an externally trained classifier reads a class label off a noisy image and the gradient of its log-probability is added to what a pretrained model predicts (Song et al., 2021; Dhariwal & Nichol, 2021). In our approach, it is a consequence of the coupling, and needs no annotation since the components come out of the fit and they supply exact training targets. The granularity of the conditioning follows from K which we are free to choose, as opposed to classifier guidance which is fixed by the number of classes of the dataset. The distribution is left intact as well, since averaging over the component returns p_1 exactly, so the coupled model still targets the data distribution. Classifier guidance backpropagates through a classifier at every step, and classifier-free guidance needs no gradients but evaluates the base network a second time (Dhariwal & Nichol, 2021; Ho & Salimans, 2022). Both of these strategies at least double the cost of a step, whereas our correction is a single forward pass through a network far smaller than the base network. Earlier work that starts flow matching from a mixture retrains the velocity network (Jia et al., 2024; Issachar et al., 2025; Luo et al., 2026). We do not train it at all.

We validate the pipeline on the official Rectified Flow checkpoints (Liu et al., 2023) for CIFAR-10 and CelebA-HQ. The coupled model improves FID at every step count on both datasets, and on CelebA-HQ it reaches with 64 steps the FID of the baseline at 128, and with 128 steps the one at 256, halving the number of base-network evaluations. Most of this improvement costs nothing: the closed-form part of the correction alone, without any training, already delivers most of the gain and beats the baseline at nearly every step count. The correction network, with about half a percent of the parameters of the base network and under 5% of the cost of a sampling step, adds a further gain. Training it takes 1.4 hours on CIFAR-10 and 17 hours on CelebA-HQ on a single GPU, a small fraction of the cost of training the base network.

2 Related Work

Flow matching and couplings. Flow matching was introduced concurrently by Lipman et al. (2023), Liu et al. (2023), and Albergo & Vanden-Eijnden (2023). Unlike diffusion, where the forward noising process fixes both the source and its pairing with the data (Ho et al., 2020; Song et al., 2021), flow matching leaves the coupling free, since the training loss accepts any joint distribution of source and data with the right marginals. That freedom has been used for conditional generation, the source being drawn given the target (Albergo et al., 2024), and for straighter trajectories: minibatch optimal transport (Pooladian et al., 2023; Tong et al., 2024) pairs noise with data inside each batch, curvature can be penalized directly (Lee et al., 2023), and reflow (Liu et al., 2023) re-pairs noise with the model's own outputs and retrains. All of these act on *continuous* random variables. Here, by contrast, we act on *discrete* random variables, where optimality is provable and the solution is closed-form.

Guidance. Guidance controls what a generative model produces through an auxiliary signal, such as a class label or a text prompt. Classifier guidance adds to a pretrained score or velocity the gradient of an external classifier trained to predict the signal from the noisy image (Song et al., 2021; Dhariwal & Nichol, 2021), and classifier-free guidance removes the external classifier by training the generative network with and without the signal and combining the two predictions (Ho & Salimans, 2022). In our case, the signal is simply the mixture component the sample starts from. Unlike a semantic label or a text prompt, it requires no annotation, its training targets being closed-form posteriors of the mixture. Moreover, the simple form of our model yields explicit formulas for the classifier and its gradient, up to a single scalar function.

Mixture sources. Gaussian-mixture sources have been proposed before (Jia et al., 2024; Issachar et al., 2025; Luo et al., 2026). In all these approaches, the velocity network is trained from scratch under the new source. We instead view the coupling through the lens of guidance, under which the conditional velocity is the unchanged base velocity plus the gradient of a component classifier, a term that is explicit except for one scalar function. That identity lets us keep a pretrained base velocity network frozen and shifts the effort to the training of a small correction network. Different source mixtures, a sweep over K for instance, can therefore be tried without retraining the whole pipeline. We also prove that our coupling is the exact entropic-transport optimum. To our knowledge, ours is the first coupling-side intervention that works on a pretrained velocity field without touching its weights.

3 A GAUSSIAN-MIXTURE SOURCE AND ITS OPTIMAL COUPLING

3.1 SETUP

Let $\mathbf{X}_0 \sim p_0$ and $\mathbf{X}_1 \sim p_1$ on $\mathbb{R}^d$, where p_1 is assumed to have a density $\mathbf{x}_1 \mapsto p_1(\mathbf{x}_1)$ and finite variance. A *coupling* $\Gamma \in \Pi(p_0, p_1)$ is a joint law for the pair $(\mathbf{X}_0, \mathbf{X}_1)$ whose marginals are p_0 and p_1, with $\Pi(p_0, p_1)$ denoting the set of all couplings. Flow matching transports the source p_0 to the data p_1 along $\mathbf{X}_t = t\mathbf{X}_1 + (1-t)\mathbf{X}_0$. The learned object is the velocity field

$$\mathbf{v}_t(\mathbf{x}) = \mathbb{E}_{(\mathbf{X}_0,\mathbf{X}_1)\sim\Gamma}\big[\mathbf{X}_1 - \mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}\big], \tag{1}$$

which generates the marginals, so that integrating $\mathbf{v}_t(\mathbf{x})$ from $\mathbf{x}_0 \sim p_0$ yields $\mathbf{x}_t \sim p_t = \mathrm{Law}(\mathbf{X}_t)$ for every t. Training parametrizes $\mathbf{v}_t$ with a neural network $\mathbf{v}_\theta(\cdot, t)$ and minimizes the regression $\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}([0,1]),\,(\mathbf{X}_0,\mathbf{X}_1)\sim\Gamma}\|\mathbf{v}_\theta(\mathbf{X}_t, t) - (\mathbf{X}_1 - \mathbf{X}_0)\|^2$, whose minimizer is (1).

We reserve the letter $\mathbf{u}$ for the standard Gaussian-source problem, with state $\mathbf{Z}_t = t\mathbf{X}_1 + (1-t)\boldsymbol{\epsilon}$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ independent of $\mathbf{X}_1$, marginal ρ_t, and velocity $\mathbf{u}_t$. Tweedie's formula gives the score-velocity identity, which for a Gaussian source is the one-sided interpolant relation of Albergo et al. (2023, Sec. 3.2), proved in Appendix A.1 for completeness,

$$\mathbf{u}_t(\mathbf{z}) = \frac{1}{t}\Big(\mathbf{z} + (1-t)\,\nabla \log \rho_t(\mathbf{z})\Big). \tag{2}$$

3.2 THE ENTROPIC COUPLING IS THE MIXTURE POSTERIOR

We take as source a Gaussian mixture with K components and a shared isotropic scale $\tau > 0$,

$$p_0 = \sum_{k=1}^{K} \pi_k\, \mathcal{N}(\boldsymbol{\mu}_k, \tau^2\mathbf{I}), \qquad \mathbf{X}_0 = \boldsymbol{\mu}_\kappa + \tau\boldsymbol{\epsilon}, \tag{3}$$

sampled by drawing a component $\kappa \sim \mathrm{Cat}(\boldsymbol{\pi})$ (i.e., $\Pr(\kappa = k) = \pi_k$) and a noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Since $\mathbf{X}_0$ is a deterministic function of $(\kappa, \boldsymbol{\epsilon})$, choosing the coupling Γ amounts to choosing a joint law for the triplet $(\kappa, \boldsymbol{\epsilon}, \mathbf{X}_1)$. If one draws all three independently, nothing links a data point to its starting point and the interpolation lines are free to cross. On the other hand, coupling $\mathbf{X}_0$ as a whole to $\mathbf{X}_1$ is a continuous optimal transport problem, which is no more tractable for a mixture than for a Gaussian.

Between the two lies the construction of this paper, which couples the discrete component κ to the data and draws the noise $\boldsymbol{\epsilon}$ independently. A *component coupling* is a coupling $\Gamma \in \Pi(\mathrm{Cat}(\boldsymbol{\pi}), p_1)$ between the component and the data, with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ drawn independently of $(\kappa, \mathbf{X}_1)$. Since $\kappa \sim \mathrm{Cat}(\boldsymbol{\pi})$ and the noise is independent, the map $\mathbf{X}_0 = \boldsymbol{\mu}_\kappa + \tau\boldsymbol{\epsilon}$ gives $\mathbf{X}_0 \sim p_0$ exactly, so every component coupling induces an element of $\Pi(p_0, p_1)$.

Two properties make this the productive restriction. It is a K-dimensional problem rather than a d-dimensional one, solved exactly below (Proposition 1) with no transport problem left over. Moreover, because the noise stays independent, each component's conditional path remains a standard Gaussian-source problem, which is what lets a pretrained network supply the velocity $\mathbf{u}_t$ (Section 4).

Among the component couplings we seek the one of least transport cost. Substituting $\mathbf{X}_0 = \boldsymbol{\mu}_\kappa + \tau\boldsymbol{\epsilon}$ and expanding, this cost reads

$$\begin{aligned}
\mathbb{E}\|\mathbf{X}_1 - \mathbf{X}_0\|^2 &= \mathbb{E}\big\|\mathbf{X}_1 - \boldsymbol{\mu}_\kappa - \tau\boldsymbol{\epsilon}\big\|^2 \\
&= \mathbb{E}\|\mathbf{X}_1 - \boldsymbol{\mu}_\kappa\|^2 - 2\tau\,\mathbb{E}\big\langle \mathbf{X}_1 - \boldsymbol{\mu}_\kappa,\, \boldsymbol{\epsilon}\big\rangle + \tau^2\,\mathbb{E}\|\boldsymbol{\epsilon}\|^2 \\
&= \mathbb{E}\|\mathbf{X}_1 - \boldsymbol{\mu}_\kappa\|^2 + \tau^2 d,
\end{aligned} \tag{4}$$

where the cross term vanishes because $\boldsymbol{\epsilon}$ has mean $\mathbf{0}$ and is independent of $(\kappa, \mathbf{X}_1)$, and $\mathbb{E}\|\boldsymbol{\epsilon}\|^2 = d$. What remains is a *semi-discrete* transport of the atoms $\{\boldsymbol{\mu}_k\}$ with capacities $\boldsymbol{\pi}$ onto p_1.

When solved exactly, such a problem returns a hard assignment that sends each data point to a single component. Finding this solution means solving numerically for the K dual variables that enforce the capacities (Peyré & Cuturi, 2019). We work instead with its *entropic relaxation*, the standard

device in computational optimal transport (Cuturi, 2013; Peyré & Cuturi, 2019). To that end, we add a penalty on how far the assignment strays from the prescribed component frequencies,

$$\min_{\Gamma\in\Pi(\mathrm{Cat}(\boldsymbol{\pi}),p_1)} \mathbb{E}_{(\kappa,\mathbf{X}_1)\sim\Gamma}\|\mathbf{X}_1-\boldsymbol{\mu}_\kappa\|^2 \;+\; \eta\,\mathbb{E}_{\mathbf{X}_1\sim p_1}\mathrm{KL}\big(\Gamma(\cdot\mid\mathbf{X}_1)\,\|\,\mathrm{Cat}(\boldsymbol{\pi})\big),\quad \eta>0. \tag{5}$$

The penalty makes the problem strictly convex, so its solution is unique and available in closed form given the duals.

Proposition 1. *For any $\eta>0$, problem (5) has the unique solution*

$$\Gamma_\eta(k\mid\mathbf{x}_1)=\mathrm{Softmax}_k\Big(\log\pi_k+\frac{\psi_k-\|\mathbf{x}_1-\boldsymbol{\mu}_k\|^2}{\eta}\Big), \tag{6}$$

where the duals $\boldsymbol{\psi}\in\mathbb{R}^K$ are determined by the $\mathrm{Cat}(\boldsymbol{\pi})$-marginal constraint. Moreover, if

$$\mathbb{E}_{\mathbf{X}_1\sim p_1}\big[p_0(k\mid\mathbf{X}_1)\big]=\pi_k,\quad k=1,\dots,K, \tag{7}$$

and the temperature is set to $\eta=2\tau^2$, then the duals vanish, $\boldsymbol{\psi}=\mathbf{0}$, so that the optimal coupling is the posterior responsibility

$$\Gamma(k\mid\mathbf{x}_1)=p_0(k\mid\mathbf{x}_1)=\mathrm{Softmax}_k\Big(\log\pi_k-\tfrac{\|\mathbf{x}_1-\boldsymbol{\mu}_k\|^2}{2\tau^2}\Big). \tag{8}$$

The proof of Proposition 1 is in Appendix A.3. When the temperature is set optimally, the optimal coupling is available in closed form with no numerical solve for $\boldsymbol{\psi}$ of the kind semi-discrete transport normally requires (Cuturi, 2013; Peyré & Cuturi, 2019). Sampling from Γ is immediate: draw a training sample $\mathbf{X}_1\sim p_1$, and then κ from the closed-form conditional distribution $\Gamma(\cdot\mid\mathbf{X}_1)$. Problem (5) is a known reformulation of the mixture likelihood (Rigollet & Weed, 2018, Theorem 4.2), and (8) is its solution in the semi-relaxed form (Vayer & Lasalle, 2025, Proposition 2.3).

Remark 1. The construction needs the covariance in (3) to be isotropic and shared. With an anisotropic covariance, the conditional paths are no longer standard Gaussian-source problems, and the pretrained velocity cannot be reused. With one scale τ_k per component, the posterior weighs the squared distance to $\boldsymbol{\mu}_k$ by $1/(2\tau_k^2)$, and no temperature η makes the responsibility optimal for (5). One shared isotropic τ matches (5) exactly at $\eta=2\tau^2$, making the likelihood fit and the transport explicitly solvable in one computation.

Choosing $\eta=2\tau^2$ eliminates the need to compute the dual variables, provided that Condition (7) holds. Proposition 2 shows that this condition is automatically satisfied at any converged maximum-likelihood fit with all K mixture weights strictly positive. Its proof is in Appendix A.4.

Proposition 2. *Fix the means $\{\boldsymbol{\mu}_k\}_{k=1}^K$ and the scale τ of the GMM p_0. Consider the expected log-likelihood*

$$L(\boldsymbol{\pi})=\mathbb{E}_{\mathbf{X}_1\sim p_1}[\log p_0(\mathbf{X}_1)] \tag{9}$$

as a function of $\boldsymbol{\pi}$ on the open simplex, i.e., $\pi_k>0$ for all $k=1,\dots,K$ and $\sum_{k=1}^K\pi_k=1$. Then every stationary point of L under the simplex constraint satisfies Condition (7).

3.3 The coupled velocity is the uncoupled one plus a classifier gradient

Any component coupling partitions the data into *slices*. Since $\kappa\sim\mathrm{Cat}(\boldsymbol{\pi})$, Bayes' rule on Γ, the joint law of $(\kappa,\mathbf{X}_1)$, gives the law of the data conditioned on the drawn component,

$$p_1^{(k)}(\mathbf{x}_1)\coloneqq p_{\mathbf{X}_1\mid\kappa=k}(\mathbf{x}_1)=\frac{\Gamma(k\mid\mathbf{x}_1)\,p_1(\mathbf{x}_1)}{\pi_k},\qquad \sum_{k=1}^K\pi_k\,p_1^{(k)}=p_1. \tag{10}$$

Because $\boldsymbol{\epsilon}$ is independent of $(\kappa,\mathbf{X}_1)$, conditioning on the component $\kappa=k$ leaves it standard normal and independent of $\mathbf{X}_1$, so the corresponding joint distribution is $p_1^{(k)}\otimes\mathcal{N}(\mathbf{0},\mathbf{I})$. In other words, conditioned on $\kappa=k$, the data and the noise are independent. Therefore, conditioned on $\kappa=k$, $\mathbf{Z}_t\coloneqq t\mathbf{X}_1+(1-t)\boldsymbol{\epsilon}$ is an instance of the standard Gaussian-source problem with data law $p_1^{(k)}$, whose marginal we write $\rho_t^{(k)}$ and whose velocity we write $\mathbf{u}_t^{(k)}$. Writing $q_t(k\mid\mathbf{z})\coloneqq\Pr(\kappa=k\mid\mathbf{Z}_t=\mathbf{z})$ for the classifier, Bayes' rule and (2) applied to each slice give (Appendix A.2)

$$\mathbf{u}_t^{(k)}(\mathbf{z})=\mathbf{u}_t(\mathbf{z})+\frac{1-t}{t}\nabla_{\mathbf{z}}\log q_t(k\mid\mathbf{z}). \tag{11}$$

Equation (11) has the form of classifier guidance (Song et al., 2021; Dhariwal & Nichol, 2021), with κ in the role of the class label and $q_t(k \mid \mathbf{z})$ in the role of a classifier reading that label off a noisy image. Two things differ from the usual construction. First, the classifier is internal. Ordinary guidance trains a separate network against annotations supplied from outside the generative model, whereas the target of q_t is a closed-form function of the mixture, as Proposition 5 makes explicit. Second, there is no strength parameter. Guidance implementations multiply the gradient by a scale chosen by hand, whereas (11) is an equality between two exact velocity fields, so $\frac{1-t}{t}$ is the only coefficient under which the field generates the slice $p_1^{(k)}$. Mixing over κ returns p_1, as shown in Proposition 3, whose proof is in Appendix A.5.

Proposition 3. *Sampling $\kappa \sim \mathrm{Cat}(\boldsymbol{\pi})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ independently, computing $\mathbf{x}_0 = \boldsymbol{\mu}_\kappa + \tau\boldsymbol{\epsilon}$, and integrating the exact conditional velocity field $\mathbf{v}_t(\cdot \mid k)$ to $t = 1$ produces a sample $\mathbf{x}_1^{(\kappa)} \sim p_1^{(\kappa)}$, and mixing those slice samples over κ produces a sample $\mathbf{x}_1 \sim p_1$.*

4 From the Guidance Identity to a Trainable Model

A pretrained network supplies the base velocity and the coupling adds the correction of (11). We first derive the change of variables that enables the reuse, then isolate the single object the reuse leaves unknown, and then choose how to model it.

4.1 The Formula of the Velocity

Define

$$c(t) = t + (1-t)\tau, \quad s(t) = \frac{t}{t + (1-t)\tau}. \tag{12}$$

Proposition 4 shows that, conditioned on a mixture component $\kappa = k$, the GMM interpolation can be expressed in terms of the standard Gaussian-source interpolation, up to a reparameterization of time. Its proof is in Appendix A.6.

Proposition 4. *Under any component coupling, conditioned on $\kappa = k$,*

$$\mathbf{X}_t \mid \{\kappa = k\} = c(t)\mathbf{Z}_{s(t)} + (1-t)\boldsymbol{\mu}_k, \tag{13}$$

and the conditional velocity $\mathbf{v}_t(\mathbf{x} \mid k) = \mathbb{E}[\mathbf{X}_1 - \mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}, \kappa = k]$ is

$$\mathbf{v}_t(\mathbf{x} \mid k) = \frac{1-\tau}{c(t)}\big(\mathbf{x} - (1-t)\boldsymbol{\mu}_k\big) + \frac{\tau}{c(t)}\,\mathbf{u}^{(k)}_{s(t)}\left(\frac{\mathbf{x} - (1-t)\boldsymbol{\mu}_k}{c(t)}\right) - \boldsymbol{\mu}_k. \tag{14}$$

The marginal velocity is the posterior mixture

$$\mathbf{v}_t(\mathbf{x}) = \sum_{k=1}^{K} p_t(k \mid \mathbf{x})\mathbf{v}_t(\mathbf{x} \mid k), \quad p_t(k \mid \mathbf{x}) \coloneqq \Pr(\kappa = k \mid \mathbf{X}_t = \mathbf{x}). \tag{15}$$

Conditioned on $\kappa = k$, Equation (13) shows that the mixture path is the standard Gaussian-source path, up to an affine transformation and a rescaling of time. The conditional velocity (14) therefore only needs the slice velocity $\mathbf{u}^{(k)}_{s(t)}$, which by (11) is the pretrained velocity plus a classifier gradient, so the pretrained network is reused as is and only the gradient remains to be determined. The network is evaluated at the shifted time $s(t)$, which is exactly the timestep shift of Esser et al. (2024) with shift parameter $1/\tau$, chosen empirically there and derived here.

4.2 What has to be learned

Proposition 5. *Let $\sigma_t^2 = t^2\tau^2 + (1-t)^2$ and set*

$$\mathbf{m}_k(\mathbf{z}, t) = \frac{(1-t)^2\boldsymbol{\mu}_k + t\tau^2\mathbf{z}}{\sigma_t^2}, \quad \nu_t = \frac{\tau^2(1-t)^2}{\sigma_t^2}, \quad F_t(\mathbf{m}) = \mathbb{E}_{\mathbf{X}\sim\mathcal{N}(\mathbf{m},\,\nu_t\mathbf{I})}\left[\frac{p_1(\mathbf{X})}{p_0(\mathbf{X})}\right]. \tag{16}$$

Then the classifier is

$$q_t(k \mid \mathbf{z}) = \mathrm{Softmax}_k\,\ell_k(\mathbf{z}, t), \quad \ell_k(\mathbf{z}, t) = \log\pi_k - \frac{\|\mathbf{z} - t\boldsymbol{\mu}_k\|^2}{2\sigma_t^2} + \log F_t\big(\mathbf{m}_k(\mathbf{z}, t)\big), \tag{17}$$

and the correction of (11) *is*

$$\mathbf{u}_t^{(k)}(\mathbf{z}) - \mathbf{u}_t(\mathbf{z}) = \frac{1-t}{\sigma_t^2}\Big(\boldsymbol{\mu}_k - \sum_{j=1}^{K} q_t(j \mid \mathbf{z})\,\boldsymbol{\mu}_j\Big) + \frac{(1-t)\tau^2}{\sigma_t^2}\Big(\big[\nabla \log F_t\big](\mathbf{m}_k) - \sum_{j=1}^{K} q_t(j \mid \mathbf{z})\big[\nabla \log F_t\big](\mathbf{m}_j)\Big). \tag{18}$$

The proof of Proposition 5 is in Appendix A.7. Equations (17) and (18) are available in closed-form from the mixture parameters, up to the scalar function F_t. This is the only remaining quantity that needs to be modeled.

4.3 Parameterization

Everything in Proposition 5 is explicit except F_t, which appears as $\log F_t$ in the logits (17) and as $\nabla \log F_t$ in the correction (18). The most direct design would learn the single scalar function $\log F_t$ and compute the gradient from it. We do not, for two reasons. First, computing the gradient at every sampling step would be cost prohibitive. Second, learning a good potential is not learning a good gradient: a regression on the potential controls its values but not its derivatives, and an error that is small yet oscillates fast leaves the potential nearly exact while perturbing the gradient by an arbitrary amount. Accordingly, in energy-based diffusion models, parameterizing the score directly outperforms differentiating a learned energy (Salimans & Ho, 2021).

For the design, we drop the term $\log F_t$ and represent the second term of (18) with a neural network

$$\hat{\mathbf{u}}_t^{(k)}(\mathbf{z}) = \mathbf{u}_t(\mathbf{z}) + \frac{1-t}{\sigma_t^2}\Big(\boldsymbol{\mu}_k - \sum_{j} q_t^{\mathrm{ref}}(j \mid \mathbf{z})\,\boldsymbol{\mu}_j\Big) + \frac{(1-t)\tau^2}{\sigma_t^2}\,\mathbf{W}_\phi\big(\mathbf{m}_k,\, \mathbf{z},\, q_t^{\mathrm{ref}}(\cdot \mid \mathbf{z}),\, t\big), \tag{19}$$

where

$$q_t^{\mathrm{ref}}(k \mid \mathbf{z}) = \mathrm{Softmax}_k\left(\log \pi_k - \frac{\|\mathbf{z} - t\boldsymbol{\mu}_k\|^2}{2\sigma_t^2}\right). \tag{20}$$

The mean-squared-error training of $\mathbf{W}_\phi$ can correct for the missing $\log F_t$ term inside q_t, since the error it induces in the first term of (19) is absorbed into the regression target of $\mathbf{W}_\phi$.

The convolutional network $\mathbf{W}_\phi$ is small, with $(\mathbf{m}_k, \mathbf{z})$ as spatial input and its conditioning vector splits evenly between a sinusoidal embedding of t and a learned map of $q_t^{\mathrm{ref}}(\cdot \mid \mathbf{z})$. Its architecture and initialization details are in Appendix B.1.

The network $\mathbf{W}_\phi$ is trained by the flow matching regression with the component drawn from the coupling,

$$\mathcal{L}(\phi) = \mathbb{E}\left\|\hat{\mathbf{u}}_t^{(\kappa)}(\mathbf{Z}_t) - (\mathbf{X}_1 - \boldsymbol{\epsilon})\right\|^2, \quad \mathbf{X}_1 \sim p_1, \quad \kappa \sim \Gamma(\cdot \mid \mathbf{X}_1), \quad t \sim \mathcal{U}(0,1), \tag{21}$$

with $\mathbf{Z}_t = t\mathbf{X}_1 + (1-t)\boldsymbol{\epsilon}$. Its minimizer is the exact slice velocity (11), which is what lets the training absorb the missing $\log F_t$ term of q_t^{ref}.

Finally, setting $\mathbf{W}_\phi = \mathbf{0}$ in (19) yields a model that requires no training at all, the correction reducing to its closed-form term under q_t^{ref}. We report its results alongside the trained model in Section 5, under the name "GMM untrained", to show that even without training the coupling and q_t^{ref} already improve generation.

5 Experiments

Every mixture is fitted by maximizing the log-likelihood

$$\mathbb{E}_{\mathbf{X}_1 \sim p_1}\big[\log p_0(\mathbf{X}_1)\big] \tag{22}$$

by full-batch gradient descent with Adam, learning the means, the weights and the single shared scale τ together (Appendix B.2). Sampling uses a uniform Euler grid throughout, in t for the Gaussian baseline and in $s(t)$ of Equation (12) for every GMM model.

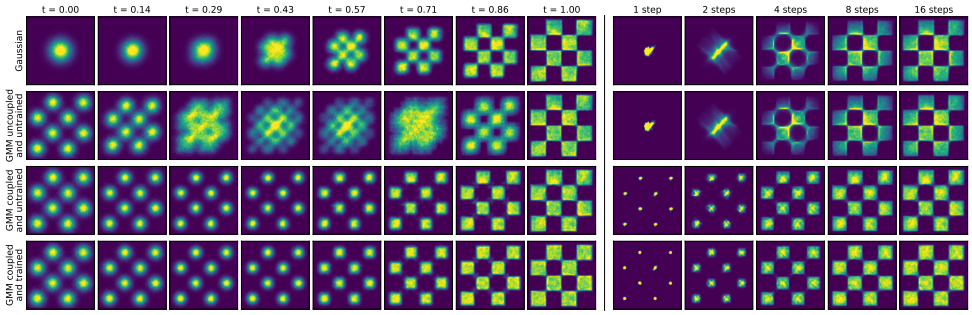


Figure 1: Flow matching toward the checkerboard density with different sources and couplings, one model per row. Left: marginals at intermediate times t along a 256-step Euler integration. Right: the final distribution at $t = 1$ when sampling with a small number of Euler steps.

5.1 A TWO-DIMENSIONAL ILLUSTRATION

We run the whole construction for $d = 2$, where trajectories can be seen. The target is the checkerboard density commonly used in the flow matching literature. The base velocity network is an MLP trained with the standard Gaussian source. We fit the Gaussian mixture with $K = 64$, deliberately mismatched with the eight squares of the density so that the source does not know the mode structure. The correction network is an MLP with 4% of the base velocity network parameters.

Figure 1 reports two experiments on the same four models. On the left, every model is integrated with 256 Euler steps, and the marginals are shown at intermediate times. On the right, every model is integrated with a small number of Euler steps and only the final distribution at $t = 1$ is shown. The Gaussian row is standard flow matching, with the source drawn independently of the target. The three remaining rows start from the fitted mixture; although it has 64 components, maximum likelihood puts them successfully on the eight squares. The uncoupled row needs no correction, since under the independent coupling every slice equals p_1 and the conditional field of (14) reduces to the base network. The two coupled rows add the correction of (18), first with $\mathbf{W}_\phi = \mathbf{0}$, then with the trained $\mathbf{W}_\phi$.

The figure disentangles the three ingredients of the construction, the source, the coupling, and the learned correction, and the four rows show that most of the improvement comes from the coupling. The source alone changes nothing, the uncoupled path dissolves the structure it starts with, every component transporting to every square. It arrives at essentially the same distribution as the Gaussian model at $t = 1$. The coupling changes everything. The eight clusters never dissolve, they expand into their squares. For the coupled models, the checkerboard is already recognizable from four steps, while the Gaussian and uncoupled models lag at every step count. The trained $\mathbf{W}_\phi$ improves slightly on the closed form by filling the final squares more uniformly. The experiment shows that generation is improved just by fitting a GMM and adding a closed-form term to the velocity.

5.2 SETUP OF THE IMAGE EXPERIMENTS

The image experiments run on the official Rectified Flow checkpoints (Liu et al., 2023): DDPM++ on CIFAR-10 (61.8M parameters, trained with horizontal flips) and NCSN++ on CelebA-HQ 256 (65.6M, trained without). Throughout, each training set is the one the base network was trained on: the CIFAR-10 network saw horizontal flips, so its training set is the flip-symmetrized one with 100k images, while the CelebA-HQ network saw none, so its training set is the 30k original images. The gradient of (22) is accumulated in chunks since the whole dataset does not hold in memory.

We sweep K over powers of four and select the model with the highest log-likelihood on the test set of CIFAR-10. It peaks at $K = 4096$ and overfits at $K = 16384$ (Table 1), so we keep the former, about 24 training images per component. CelebA-HQ has no test split, so we match the average number of training images per component: $K = 1024$ for 30k images. The learned scale is $\tau = 0.313$ on CIFAR-10 and $\tau = 0.342$ on CelebA-HQ.

K	64	256	1024	**4096**	16384
train	-0.412	-0.358	-0.305	-0.259	-0.178
test	-0.412	-0.366	-0.331	$\mathbf{-0.319}$	-0.330

Table 1: CIFAR-10 log-likelihood per dimension of the fitted mixture, on train and test set.

The total-variation gap between the two sides of (7) falls below 10^{-4}, the hypothesis of Proposition 1 holds up to numerical accuracy.

One network $\mathbf{W}_\phi$ is trained per dataset, with the same architecture (Appendix B.1) and the same recipe on both (Appendix B.2). The network $\mathbf{W}_\phi$ has 0.352M parameters on CIFAR-10 ($K = 4096$) and 0.328M on CelebA-HQ ($K = 1024$), about half a percent of the base velocity network. Training minimizes (21). Its cost is small and dominated by the frozen base network, which supplies one forward pass per training sample while receiving no gradient. On one V100, training takes 1.4 hours on CIFAR-10 and 17 hours on CelebA-HQ 256. Trying a different source, a new K for instance, therefore costs a mixture fit and a new $\mathbf{W}_\phi$, with the base network untouched.

5.3 Few-step generation

Our pipeline reproduces the published numbers up to the known discrepancy between TF-hub and torch Inception features (Appendix B.2). We evaluate FID against the full training set, the published protocol for these checkpoints, with 50k generated samples on CIFAR-10 and 10k on CelebA-HQ, plus the Inception Score (Salimans et al., 2016) on CIFAR-10.

Table 2 reports the results on both datasets. The coupled model improves FID at every step count. On CelebA-HQ, its 64-step FID of 11.24 already beats the baseline's 128-step 11.45, and its 128-step 9.80 the baseline's 256-step 10.01, so the coupling more than halves the number of base-network evaluations needed to reach either quality. The margin narrows as the step count grows, which is what the theory predicts, since with exact fields both models target the same marginal (Proposition 3). Appendix B.3 reports the untrained model at every K of the sweep. Larger mixtures help most at small step counts, and the differences between them close as the step count grows.

Euler steps	1	2	4	8	16	32	64	128	256
CelebA-HQ 256, FID ↓									
Gaussian	258.40	182.43	137.16	89.01	48.53	25.08	15.00	11.45	10.01
GMM untrained	186.94	123.86	93.61	63.60	37.12	19.91	12.81	10.63	9.63
GMM trained	**182.97**	**115.04**	**87.38**	**58.90**	**33.40**	**16.69**	**11.24**	**9.80**	**9.42**
CIFAR-10, FID ↓									
Gaussian	371.25	173.07	52.95	17.20	8.59	5.47	4.13	3.43	3.05
GMM untrained	151.43	**85.62**	31.36	14.49	8.48	5.65	4.14	3.32	2.98
GMM trained	**151.33**	87.01	**30.52**	**14.10**	**8.08**	**5.19**	**3.91**	**3.21**	**2.95**
CIFAR-10, Inception Score ↑									
Gaussian	1.13	2.92	6.17	8.11	8.67	8.98	9.17	9.28	**9.40**
GMM untrained	3.77	5.46	7.59	**8.42**	**8.86**	9.00	9.18	9.27	9.37
GMM trained	**3.83**	**5.52**	**7.67**	**8.42**	8.81	**9.07**	**9.20**	**9.37**	9.35

Table 2: FID and Inception Score across Euler step counts, 10k samples on CelebA-HQ 256 ($K = 1024$) and 50k on CIFAR-10 ($K = 4096$), against the full training sets.

Figure 2 shows samples conditioned on a given component. The samples inherit the palette and rough composition of the component mean and nothing more; identity, class and background vary freely within the slice.

5.4 The model does not memorize the training set

Since the source is fitted to the training images, a natural worry is that the model overfits the training set. We draw 1000 samples from our trained coupled models and 1000 from the Gaussian models, all with 256-step Euler. We record the pixel-space distance between every sample and its nearest

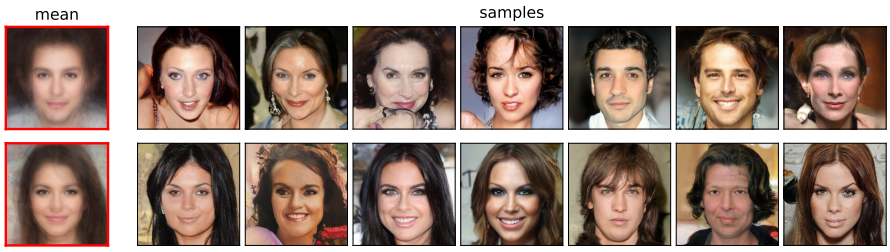


Figure 2: The mean of a single component (left) and seven samples of the coupled model conditioned on it, 256 Euler steps, on CelebA-HQ. The CIFAR-10 counterpart is in Appendix B.3.

training image and report the statistics in Table 3; the statistics in the Inception latent space, which agree, are in Appendix B.3. We also show the closest pairs in Figure 3.

On both datasets and in both spaces, the coupled and Gaussian distributions are very similar at every reported statistic. On CIFAR-10, the statistics also show that both are similar to the distance distribution between the training and test sets. The coupling does not move samples toward the training set.

	mean	std	min	p_1	p_5	median	max
CelebA-HQ, pixel L^2							
GMM coupled	162.0	22.9	95.8	110.9	125.5	161.8	231.4
Gaussian	159.7	22.7	95.3	111.2	123.9	159.2	241.3
CIFAR-10, pixel L^2							
GMM coupled	18.35	3.98	6.00	9.12	11.66	18.51	30.92
Gaussian	18.10	4.24	6.69	8.39	11.24	18.15	30.09
test set	18.11	4.37	0.51	8.02	10.67	18.21	36.10

Table 3: Distance of 1000 samples per model to their nearest training image, in pixel space.

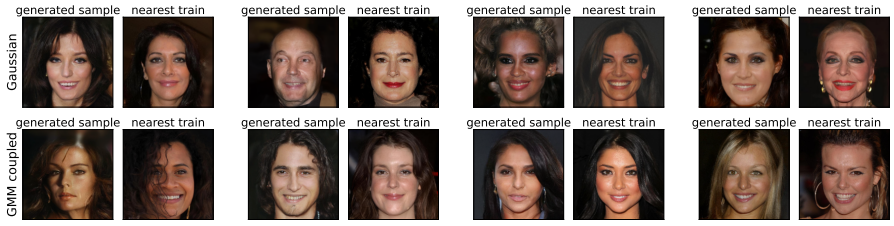


Figure 3: The four generated samples closest to the training set, out of 1000, in pixel space, each next to its nearest training image, on CelebA-HQ. The CIFAR-10 counterpart is in Appendix B.3.

6 Conclusion

We replaced the Gaussian source of flow matching with a fitted Gaussian mixture, coupled to the data through an optimal entropic coupling. A pretrained velocity network is reused without retraining and the target distribution is left intact. The coupling adds a correction that is mostly closed-form, and the closed-form part alone already delivers most of the improvement; a small correction network learns the rest. On two published checkpoints, the coupled model improves FID at every step count, reaches the quality of the baseline on CelebA-HQ with half the steps, and does not memorize the training set. Since the base network stays untouched, the source becomes a design parameter: a different mixture costs a mixture fit and a correction network, not a full retraining. Combining the coupling with accelerations that act on the velocity itself, such as distillation or reflow, and moving the construction to the latent space of larger models, are natural next steps.

Reproducibility Statement

All experiments use the two publicly released Rectified Flow checkpoints of Liu et al. (2023), the DDPM++ model for CIFAR-10 and the NCSN++ model for CelebA-HQ 256, with their released configurations and no change to their weights. The datasets, their preprocessing and the flip-symmetrization of the CIFAR-10 training set are described in Section 5.2. The mixture fit is described in Section 5 and Appendix B.2, and the selection of K and the fitted scales in Section 5.2; the architecture of the correction network is given in Appendix B.1 and its full training recipe, together with the evaluation protocol and the parity check against the published FID, in Appendix B.2. The two-dimensional experiment of Section 5.1 is fully specified there and runs on a CPU in minutes. The code for fitting the mixtures, training the correction networks and sampling, with the configuration of every table, is available at `https://github.com/CoupledGMMFlow/coupled-gmm-flow`.

The Use of Large Language Models

Large language models were used as an assistant in four ways. To aid and polish the writing, by rephrasing sentences. For retrieval and discovery, to find and check related work. For research execution, to help write and document the evaluation, sampling and orchestration code, and to launch and monitor the evaluation runs under the authors' instructions. For the mathematical claims, to assist with derivations and to check the proofs, which the authors verified.

References

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations (ICLR)*, 2023.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: a unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Michael S. Albergo, Mark Goldstein, Nicholas M. Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. In *International Conference on Machine Learning (ICML)*, volume 235, pp. 921–937, 2024.

Athanassia G. Bacharoglou. Approximation of probability distributions by convex mixtures of Gaussian measures. *Proceedings of the American Mathematical Society*, 138(7):2619–2628, 2010.

Sinho Chewi, Jonathan Niles-Weed, and Philippe Rigollet. Statistical optimal transport. *arXiv preprint arXiv:2407.18163*, 2024.

Marco Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980.

Patrick Esser, Sumith Kulal, Andreas Blattmann, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Noam Issachar, Mohammad Salama, Raanan Fattal, and Sagie Benaim. Designing a conditional prior distribution for flow-based generative models. *Transactions on Machine Learning Research (TMLR)*, 2025.

Nanshan Jia, Tingyu Zhu, Haoyu Liu, and Zeyu Zheng. Structured diffusion models with mixture of Gaussians as prior distribution. *arXiv preprint arXiv:2410.19149*, 2024.

Sangyun Lee, Beomsu Kim, and Jong Chul Ye. Minimizing trajectory curvature of ODE-based generative models. In *International Conference on Machine Learning (ICML)*, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

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 (ICLR)*, 2023.

Gaoxiang Luo, Frank Cole, Sihang Zhang, Yuxiang Wan, Yulong Lu, and Ju Sun. Flow matching for multimodal distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23260–23271, June 2026.

Geoffrey McLachlan and David Peel. *Finite Mixture Models*. Wiley, 2000.

TrungTin Nguyen, Hien D. Nguyen, Faicel Chamroukhi, and Geoffrey J. McLachlan. Approximation by finite mixtures of continuous density functions that vanish at infinity. *Cogent Mathematics & Statistics*, 7(1), 2020.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: straightening flows with minibatch couplings. In *International Conference on Machine Learning (ICML)*, 2023.

Philippe Rigollet and Jonathan Weed. Entropic optimal transport is maximum-likelihood deconvolution. *Comptes Rendus Mathématique*, 356(11–12):1228–1235, 2018.

Tim Salimans and Jonathan Ho. Should EBMs model the energy or the score? In *Energy Based Models Workshop, ICLR*, 2021.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

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 (ICLR)*, 2021.

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. *Transactions on Machine Learning Research (TMLR)*, 2024.

Titouan Vayer and Etienne Lasalle. A note on the relations between mixture models, maximum-likelihood and entropic optimal transport. *arXiv preprint arXiv:2501.12005*, 2025.

A PROOFS

A.1 PROOF OF THE SCORE-VELOCITY IDENTITY (2)

The marginal of $\mathbf{Z}_t = t\mathbf{X}_1 + (1-t)\boldsymbol{\epsilon}$ is $\rho_t(\mathbf{z}) = \mathbb{E}_{\mathbf{X}_1 \sim p_1}\big[\mathcal{N}(\mathbf{z};\, t\mathbf{X}_1,\, (1-t)^2\mathbf{I})\big]$, and by (1) the velocity field is $\mathbf{u}_t(\mathbf{z}) = \mathbb{E}[\mathbf{X}_1 - \boldsymbol{\epsilon} \mid \mathbf{Z}_t = \mathbf{z}]$. Next, write

$$\mathbf{z} = \mathbb{E}[\mathbf{Z}_t \mid \mathbf{Z}_t = \mathbf{z}] = t\mathbb{E}[\mathbf{X}_1 \mid \mathbf{Z}_t = \mathbf{z}] + (1-t)\mathbb{E}[\boldsymbol{\epsilon} \mid \mathbf{Z}_t = \mathbf{z}], \tag{23}$$

which is equivalent to

$$\mathbb{E}[\boldsymbol{\epsilon} \mid \mathbf{Z}_t = \mathbf{z}] = \frac{1}{1-t}\Big(\mathbf{z} - \mathbb{E}[t\mathbf{X}_1 \mid \mathbf{Z}_t = \mathbf{z}]\Big). \tag{24}$$

We also recall Tweedie's formula, which in our setting reads

$$\mathbb{E}[t\mathbf{X}_1 \mid \mathbf{Z}_t = \mathbf{z}] = \mathbf{z} + (1-t)^2 \nabla \log \rho_t(\mathbf{z}). \tag{25}$$

The formulas above imply that

$$\begin{aligned}
\mathbf{u}_t(\mathbf{z}) &= \mathbb{E}[\mathbf{X}_1 \mid \mathbf{Z}_t = \mathbf{z}] - \mathbb{E}[\boldsymbol{\epsilon} \mid \mathbf{Z}_t = \mathbf{z}] \\
&= \frac{1}{t}\mathbb{E}[t\mathbf{X}_1 \mid \mathbf{Z}_t = \mathbf{z}] - \frac{1}{1-t}\Big(\mathbf{z} - \mathbb{E}[t\mathbf{X}_1 \mid \mathbf{Z}_t = \mathbf{z}]\Big) \\
&= \frac{1}{t}\Big(\mathbf{z} + (1-t)\,\nabla \log \rho_t(\mathbf{z})\Big),
\end{aligned} \tag{26}$$

which is the desired result. □

A.2 DERIVATION OF THE GUIDANCE IDENTITY (11)

The slice marginal $\rho_t^{(k)}$ is the conditional density of $\mathbf{Z}_t$ given $\kappa = k$, so the law of total probability gives $\sum_{k=1}^K \pi_k \rho_t^{(k)} = \rho_t$, and Bayes' rule implies that

$$\rho_t^{(k)}(\mathbf{z}) = \frac{q_t(k \mid \mathbf{z})}{\pi_k}\, \rho_t(\mathbf{z}). \tag{27}$$

The slice velocities are now related to the marginal one in closed form. Taking logarithms in (27) and differentiating in $\mathbf{z}$, the constant $\log \pi_k$ drops and

$$\nabla \log \rho_t^{(k)}(\mathbf{z}) = \nabla \log \rho_t(\mathbf{z}) + \nabla_{\mathbf{z}} \log q_t(k \mid \mathbf{z}). \tag{28}$$

The slice problem is itself a standard Gaussian-source problem, so (2) applies to it and yields (11). □

A.3 PROOF OF PROPOSITION 1

We have that $\mathbb{E}_{\mathbf{X}_1 \sim p_1} \mathrm{KL}\left(\Gamma(\cdot \mid \mathbf{X}_1) \,\|\, \mathrm{Cat}(\boldsymbol{\pi})\right) = \mathrm{KL}(\Gamma \,\|\, \mathrm{Cat}(\boldsymbol{\pi}) \otimes p_1)$. Therefore, Problem (5) is equivalent to the standard entropic optimal transport (EOT) problem

$$\min_{\Gamma \in \Pi(\mathrm{Cat}(\boldsymbol{\pi}), p_1)} \mathbb{E}_{(\kappa, \mathbf{X}_1) \sim \Gamma} \|\mathbf{X}_1 - \boldsymbol{\mu}_\kappa\|^2 \;+\; \eta\, \mathrm{KL}(\Gamma \,\|\, \mathrm{Cat}(\boldsymbol{\pi}) \otimes p_1). \tag{29}$$

Under the assumption that $\mathrm{Cat}(\boldsymbol{\pi})$ and p_1 have finite second-order moments (this holds automatically if p_1 is compactly supported), which is the case here, the existence and uniqueness of the solution to Problem (29) is a standard result (Chewi et al., 2024). Moreover, with respect to the product of the counting and the Lebesgue measure, the optimal solution has density

$$\Gamma(k, \mathbf{x}_1) = \pi_k p_1(\mathbf{x}_1) e^{(\varphi(\mathbf{x}_1) + \psi_k - \|\mathbf{x}_1 - \boldsymbol{\mu}_k\|^2)/\eta}, \tag{30}$$

where $\varphi(\mathbf{x}_1)$ and ψ_k are dual variables (Chewi et al., 2024, Proposition 4.3), also referred to as *EOT potentials*. They satisfy the Schrödinger equation (Chewi et al., 2024, Section 4.2)

$$\varphi(\mathbf{x}_1) = -\eta \log \sum_{i=1}^K \pi_i e^{(\psi_i - \|\mathbf{x}_1 - \boldsymbol{\mu}_i\|^2)/\eta}. \tag{31}$$

Plugging (31) into (30) yields the closed form expression

$$\begin{aligned}\Gamma(k \mid \mathbf{x}_1) = \frac{\Gamma(k, \mathbf{x}_1)}{p_1(\mathbf{x}_1)} &= \pi_k e^{(\varphi(\mathbf{x}_1)+\psi_k-\|\mathbf{x}_1-\boldsymbol{\mu}_k\|^2)/\eta} \\ &= \frac{\pi_k e^{(\psi_k-\|\mathbf{x}_1-\boldsymbol{\mu}_k\|^2)/\eta}}{\sum_{i=1}^K \pi_i e^{(\psi_i-\|\mathbf{x}_1-\boldsymbol{\mu}_i\|^2)/\eta}} \\ &= \mathrm{Softmax}_k\left(\log \pi_k + \frac{\psi_k - \|\mathbf{x}_1 - \boldsymbol{\mu}_k\|^2}{\eta}\right). \end{aligned} \tag{32}$$

The dual variable ψ_k corresponds to the Lagrange multiplier of the $\mathrm{Cat}(\boldsymbol{\pi})$-marginal constraint

$$\int_{\mathbb{R}^d} \Gamma(k, \mathbf{x}_1)\, \mathrm{d}\mathbf{x}_1 = \pi_k \iff \mathbb{E}_{\mathbf{X}_1 \sim p_1}[\Gamma(k \mid \mathbf{X}_1)] = \pi_k \tag{33}$$

for all $k = 1, \ldots, K$.

Next, set $\eta = 2\tau^2$ and $\boldsymbol{\psi} = \mathbf{0}$. Then $\Gamma(k \mid \mathbf{x}_1) = p_0(k \mid \mathbf{x}_1)$, the mixture posterior (8). Its p_1-marginal is automatic, and Condition (7) is exactly the remaining $\mathrm{Cat}(\boldsymbol{\pi})$-marginal constraint (33). Thus, under (7), the choice $\boldsymbol{\psi} = \mathbf{0}$ satisfies the marginal constraint that determines the optimal coupling. Consequently, the posterior responsibility is the entropic optimal coupling, and uniqueness of the solution to (29) shows that it is the unique optimum. □

A.4 Proof of Proposition 2

Since $p_0(k \mid \mathbf{x}_1) = \pi_k \mathcal{N}(\mathbf{x}_1; \boldsymbol{\mu}_k, \tau^2\mathbf{I})/p_0(\mathbf{x}_1)$, differentiating the log-likelihood with respect to π_k gives

$$\frac{\partial}{\partial \pi_k} \mathbb{E}_{p_1}\big[\log p_0(\mathbf{X}_1)\big] = \mathbb{E}_{p_1}\left[\frac{\mathcal{N}(\mathbf{X}_1; \boldsymbol{\mu}_k, \tau^2\mathbf{I})}{p_0(\mathbf{X}_1)}\right] = \frac{\mathbb{E}_{p_1}\big[p_0(k \mid \mathbf{X}_1)\big]}{\pi_k}. \tag{34}$$

We optimize over the open simplex, so there are no inequality constraints to include in the Lagrangian, we only introduce a multiplier for the normalization constraint. The Lagrangian is

$$\mathcal{L}(\boldsymbol{\pi}, \lambda) = \mathbb{E}_{p_1}\big[\log p_0(\mathbf{X}_1)\big] - \lambda\left(\sum_{k=1}^K \pi_k - 1\right), \tag{35}$$

with $\lambda \in \mathbb{R}$. Its stationary points satisfy

$$\frac{\mathbb{E}_{p_1}\big[p_0(k \mid \mathbf{X}_1)\big]}{\pi_k} - \lambda = 0 \tag{36}$$

for all $k = 1, \ldots, K$, or equivalently,

$$\mathbb{E}_{p_1}\big[p_0(k \mid \mathbf{X}_1)\big] = \lambda \pi_k, \quad k = 1, \ldots, K. \tag{37}$$

Summing (37) over k and using $\sum_{k=1}^K p_0(k \mid \mathbf{x}_1) = 1$ and $\sum_{k=1}^K \pi_k = 1$ gives $\lambda = 1$, which yields Condition (7). □

A.5 Proof of Proposition 3

Conditioned on $\kappa = k$, the construction of Section 3.3 given by

$$\mathbf{X}_t = t\mathbf{X}_1 + (1-t)\mathbf{X}_0, \quad \mathbf{X}_0 = \boldsymbol{\mu}_k + \tau\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{38}$$

is standard flow matching between $\mathcal{N}(\boldsymbol{\mu}_k, \tau^2\mathbf{I})$ and the slice $p_1^{(k)}$. By the marginal-generation property of the exact velocity field (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023), integrating the corresponding exact conditional velocity field

$$\mathbf{v}_t(\mathbf{x} \mid k) = \mathbb{E}[\mathbf{X}_1 - \mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}, \kappa = k] \tag{39}$$

from $t = 0$ to $t = 1$ with initial point $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_k, \tau^2\mathbf{I})$ produces $\mathbf{x}_1 \sim p_1^{(k)}$. Mixing over $\kappa \sim \mathrm{Cat}(\boldsymbol{\pi})$ gives $\sum_{k=1}^K \pi_k p_1^{(k)} = p_1$ by the definition of the slices (10). □

An analytical expression of $\mathbf{v}_t(\mathbf{x} \mid k)$ is computed in Proposition 4.

A.6 Proof of Proposition 4

Conditioned on $\kappa = k$, we have that $\mathbf{X}_t = t\mathbf{X}_1 + (1-t)\boldsymbol{\mu}_k + (1-t)\tau\,\boldsymbol{\epsilon}$. With $c(t) = t + (1-t)\tau$ and $s(t) = t/c(t)$,

$$\begin{aligned} c(t)\mathbf{Z}_{s(t)} + (1-t)\boldsymbol{\mu}_k &= c(t)\Big[\tfrac{t}{c(t)}\mathbf{X}_1 + \big(1 - \tfrac{t}{c(t)}\big)\boldsymbol{\epsilon}\Big] + (1-t)\boldsymbol{\mu}_k \\ &= t\mathbf{X}_1 + (1-t)\tau\boldsymbol{\epsilon} + (1-t)\boldsymbol{\mu}_k \\ &= \mathbf{X}_t \mid \{\kappa = k\}, \end{aligned} \tag{40}$$

which is (13). Next, we compute the conditional velocity

$$\mathbf{v}_t(\mathbf{x} \mid k) = \mathbb{E}[\mathbf{X}_1 - \mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}, \kappa = k] = \mathbb{E}[\mathbf{X}_1 - \boldsymbol{\mu}_k - \tau\boldsymbol{\epsilon} \mid \mathbf{X}_t = \mathbf{x}, \kappa = k], \tag{41}$$

due to (38). We can verify that

$$\mathbf{X}_1 - \boldsymbol{\mu}_k - \tau\boldsymbol{\epsilon} = (1-\tau)\mathbf{Z}_{s(t)} + \frac{\tau}{c(t)}(\mathbf{X}_1 - \boldsymbol{\epsilon}) - \boldsymbol{\mu}_k, \tag{42}$$

so that

$$\mathbf{v}_t(\mathbf{x} \mid k) = (1-\tau)\mathbb{E}[\mathbf{Z}_{s(t)} \mid \mathbf{X}_t = \mathbf{x}, \kappa = k] + \frac{\tau}{c(t)}\mathbb{E}[\mathbf{X}_1 - \boldsymbol{\epsilon} \mid \mathbf{X}_t = \mathbf{x}, \kappa = k] - \boldsymbol{\mu}_k. \tag{43}$$

By (13), given $\kappa = k$, $\mathbf{X}_t$ is a deterministic and invertible function of $\mathbf{Z}_{s(t)}$, so that the events

$$\{\mathbf{X}_t = \mathbf{x} \mid \kappa = k\} = \{\mathbf{Z}_{s(t)} = \mathbf{z} \mid \kappa = k\}, \quad \mathbf{z} = \frac{\mathbf{x} - (1-t)\boldsymbol{\mu}_k}{c(t)} \tag{44}$$

are equal. Therefore,

$$\begin{aligned} \mathbf{v}_t(\mathbf{x} \mid k) &= (1-\tau)\mathbf{z} + \frac{\tau}{c(t)}\mathbb{E}[\mathbf{X}_1 - \boldsymbol{\epsilon} \mid \mathbf{Z}_{s(t)} = \mathbf{z}, \kappa = k] - \boldsymbol{\mu}_k \\ &= (1-\tau)\mathbf{z} + \frac{\tau}{c(t)}\mathbf{u}^{(k)}_{s(t)}(\mathbf{z}) - \boldsymbol{\mu}_k \\ &= \frac{1-\tau}{c(t)}\big(\mathbf{x} - (1-t)\boldsymbol{\mu}_k\big) + \frac{\tau}{c(t)}\mathbf{u}^{(k)}_{s(t)}\left(\frac{\mathbf{x} - (1-t)\boldsymbol{\mu}_k}{c(t)}\right) - \boldsymbol{\mu}_k. \end{aligned} \tag{45}$$

The mixture form of the marginal velocity is the law of total probabilities over κ. □

A.7 Proof of Proposition 5

Since $\mathbf{Z}_t$ depends on κ only through $\mathbf{X}_1$, the chain $\kappa \to \mathbf{X}_1 \to \mathbf{Z}_t$ is Markov, so conditioning on $\mathbf{X}_1$ inside the conditional expectation and using the tower property,

$$q_t(k \mid \mathbf{z}) = \mathbb{E}\big[\Gamma(k \mid \mathbf{X}_1) \mid \mathbf{Z}_t = \mathbf{z}\big] = \mathbb{E}\big[p_0(k \mid \mathbf{X}_1) \mid \mathbf{Z}_t = \mathbf{z}\big]. \tag{46}$$

Therefore,

$$q_t(k \mid \mathbf{z}) = \int_{\mathbb{R}^d} p_0(k \mid \mathbf{x}_1) p(\mathbf{x}_1 \mid \mathbf{z})\,\mathrm{d}\mathbf{x}_1. \tag{47}$$

The first factor is a GMM posterior, given by

$$p_0(k \mid \mathbf{x}_1) = \frac{\pi_k\,\mathcal{N}(\mathbf{x}_1; \boldsymbol{\mu}_k, \tau^2\mathbf{I})}{p_0(\mathbf{x}_1)}. \tag{48}$$

Recalling that ρ_t is the density of $\mathbf{Z}_t$, the second factor is

$$p(\mathbf{x}_1 \mid \mathbf{z}) = \frac{p_{\mathbf{Z}_t \mid \mathbf{X}_1 = \mathbf{x}_1}(\mathbf{z}) p_1(\mathbf{x}_1)}{\rho_t(\mathbf{z})} = \frac{p_1(\mathbf{x}_1)\,\mathcal{N}(\mathbf{z}; t\mathbf{x}_1, (1-t)^2\mathbf{I})}{\rho_t(\mathbf{z})}, \tag{49}$$

due to the fact that $\mathbf{Z}_t \mid \{\mathbf{X}_1 = \mathbf{x}_1\} = t\mathbf{x}_1 + (1-t)\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We then compute that

$$\mathcal{N}(\mathbf{x}_1; \boldsymbol{\mu}_k, \tau^2\mathbf{I})\,\mathcal{N}(\mathbf{z}; t\mathbf{x}_1, (1-t)^2\mathbf{I}) = \mathcal{N}(\mathbf{z};\, t\boldsymbol{\mu}_k,\, \sigma_t^2\mathbf{I})\,\mathcal{N}\big(\mathbf{x}_1;\, \mathbf{m}_k(\mathbf{z}, t),\, \nu_t\mathbf{I}\big), \tag{50}$$

with σ_t^2, $\mathbf{m}_k(\mathbf{z}, t)$ and ν_t as in (16). Therefore,

$$\begin{aligned} q_t(k \mid \mathbf{z}) &= \frac{\pi_k \mathcal{N}(\mathbf{z};\, t\boldsymbol{\mu}_k,\, \sigma_t^2 \mathbf{I})}{\rho_t(\mathbf{z})} \int_{\mathbb{R}^d} \frac{p_1(\mathbf{x}_1)}{p_0(\mathbf{x}_1)} \mathcal{N}\big(\mathbf{x}_1;\, \mathbf{m}_k(\mathbf{z}, t),\, \nu_t \mathbf{I}\big)\, \mathrm{d}\mathbf{x}_1 \\ &= \frac{\pi_k \mathcal{N}(\mathbf{z};\, t\boldsymbol{\mu}_k,\, \sigma_t^2 \mathbf{I})}{\rho_t(\mathbf{z})} \mathbb{E}_{\mathbf{X} \sim \mathcal{N}(\mathbf{m}_k(\mathbf{z}, t),\, \nu_t \mathbf{I})} \left[\frac{p_1(\mathbf{X})}{p_0(\mathbf{X})} \right] \\ &= \frac{\pi_k \mathcal{N}(\mathbf{z};\, t\boldsymbol{\mu}_k,\, \sigma_t^2 \mathbf{I})}{\rho_t(\mathbf{z})} F_t\big(\mathbf{m}_k(\mathbf{z}, t)\big), \end{aligned} \tag{51}$$

with F_t as in (16). Since $\sum_{k=1}^K q_t(k \mid \mathbf{z}) = 1$, we have that

$$\rho_t(\mathbf{z}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{z};\, t\boldsymbol{\mu}_k,\, \sigma_t^2 \mathbf{I}) F_t(\mathbf{m}_k(\mathbf{z}, t)), \tag{52}$$

so that

$$\begin{aligned} q_t(k \mid \mathbf{z}) = \frac{\pi_k \mathcal{N}(\mathbf{z};\, t\boldsymbol{\mu}_k,\, \sigma_t^2 \mathbf{I}) F_t\big(\mathbf{m}_k(\mathbf{z}, t)\big)}{\sum_{i=1}^K \pi_i \mathcal{N}(\mathbf{z};\, t\boldsymbol{\mu}_i,\, \sigma_t^2 \mathbf{I}) F_t\big(\mathbf{m}_i(\mathbf{z}, t)\big)} &= \frac{\pi_k \exp\Big(-\frac{\|\mathbf{z} - t\boldsymbol{\mu}_k\|^2}{2\sigma_t^2}\Big) F_t\big(\mathbf{m}_k(\mathbf{z}, t)\big)}{\sum_{i=1}^K \pi_i \exp\Big(-\frac{\|\mathbf{z} - t\boldsymbol{\mu}_i\|^2}{2\sigma_t^2}\Big) F_t\big(\mathbf{m}_i(\mathbf{z}, t)\big)} \\ &= \mathrm{Softmax}_k\, \ell_k(\mathbf{z}, t) \end{aligned} \tag{53}$$

with $\ell_k(\mathbf{z}, t)$ as in (17).

For the correction, recall that by (11),

$$\mathbf{u}_t^{(k)}(\mathbf{z}) - \mathbf{u}_t(\mathbf{z}) = \frac{1 - t}{t} \nabla_{\mathbf{z}} \log q_t(k \mid \mathbf{z}). \tag{54}$$

We compute

$$\nabla_{\mathbf{z}} \log q_t(k \mid \mathbf{z}) = \nabla_{\mathbf{z}} \ell_k(\mathbf{z}, t) - \sum_{j=1}^K q_t(j \mid \mathbf{z})\, \nabla_{\mathbf{z}} \ell_j(\mathbf{z}, t), \tag{55}$$

and

$$\nabla_{\mathbf{z}} \ell_k(\mathbf{z}, t) = \frac{t\boldsymbol{\mu}_k - \mathbf{z}}{\sigma_t^2} + \frac{t\tau^2}{\sigma_t^2} [\nabla \log F_t](\mathbf{m}_k(\mathbf{z}, t)). \tag{56}$$

Plugging (56) into (55) and multiplying by the prefactor $\frac{1-t}{t}$ of (11) yields (18). □

B Experimental Details

B.1 Architecture of the Correction Network

The network $\mathbf{W}_\phi$ maps its six input channels, the concatenation of $\mathbf{m}_k$ and $\mathbf{z}$, through a 3×3 convolution to 64 channels. The result is average-pooled by a factor of 2 and passed through four residual blocks that run at the pooled resolution; their output is upsampled back by nearest-neighbor interpolation and summed with the full-resolution activation of the input convolution, a skip that restores the detail the pooling discards while making every block four times cheaper. A group normalization, a SiLU, and a 3×3 convolution back to 3 channels produce the output.

Each residual block is conditioned in the style of adaptive layer normalization (Peebles & Xie, 2023): the block contains two convolutions, each preceded by a group normalization without learned affine parameters, and a linear map of the conditioning vector produces a per-channel scale and shift for both normalizations. The conditioning vector has 16 dimensions and is the concatenation of two halves. The time half is a sinusoidal embedding of t passed through a two-layer MLP. The posterior half is $q_t^{\mathrm{ref}}(\cdot \mid \mathbf{z})$ passed through a two-layer MLP; the posterior is the only input whose dimension depends on K, so this projection is the only K-dependent part of $\mathbf{W}_\phi$. The probabilities are fed rather than the logits, a linear map of a probability vector being a convex combination of K learned vectors and therefore bounded at every t, while the spread of the logits grows without bound as the posterior sharpens. The output convolution is initialized at zero, so at initialization $\mathbf{W}_\phi = \mathbf{0}$ and the model starts exactly as the untrained model of Section 4.3.

B.2 Training Recipes and Evaluation Protocol

The mixtures are fitted with Adam at learning rate 10^{-2} for 2000 full-batch steps without decay. The means are initialized at random, the weights uniformly, and the scale τ at a value far above the spread of the data, about $3 \cdot 10^3$. At that scale every responsibility is uniform, so every component receives the full gradient of the likelihood from the first step and none collapses to zero weight, a common failure of mixture fits with many components. As the means move onto the data, τ decreases by itself to its final value.

The correction network $\mathbf{W}_\phi$ is trained by minimizing (21) with Adam at learning rate $2 \cdot 10^{-4}$, batch size 16, 100k steps with a 1k-step linear warmup and cosine decay to zero, gradient clipping at 1, t uniform on the release's range $[10^{-3}, 1]$, and an EMA of $\mathbf{W}_\phi$ with decay 0.9999, which is what sampling deploys.

Before the evaluations of Section 5.3, we verified that our pipeline reproduces the published numbers. With the release's own adaptive RK45 solver (Dormand & Prince, 1980, Dormand–Prince, tolerances 10^{-5}) we obtain FID 2.82 on CIFAR-10 against a published 2.58, the residual gap being the known discrepancy between TF-hub and torch Inception features (Parmar et al., 2022).

B.3 Additional Results

Tables 4 and 5 report the FID of the untrained model of Section 4.3 at every mixture size K of the sweep of Section 5, each with its own maximum-likelihood mixture and the same sampler as Table 2; the trained model is reported in Table 2 at the selected K. On CelebA-HQ, the FID improves with K at every step count up to 64 steps, and every mixture beats the Gaussian at every step count. On CIFAR-10, the FID improves with K up to 4 steps, with the exception of the single-step FID of the smallest mixture, which is worse than the Gaussian one. On both datasets, the differences between the mixture sizes shrink as the step count grows, which is what Proposition 3 predicts since every mixture targets the same marginal. Table 6 reports the nearest-neighbor statistics of Section 5.4 in the Inception latent space, and Figures 4 and 5 the CIFAR-10 counterparts of Figures 2 and 3.

Euler steps	1	2	4	8	16	32	64	128	256
Gaussian	258.40	182.43	137.16	89.01	48.53	25.08	15.00	11.45	10.01
GMM untrained, $K = 4$	252.49	174.30	122.69	83.35	48.33	24.38	14.24	10.99	9.80
GMM untrained, $K = 16$	242.64	163.23	111.51	75.50	44.80	22.94	13.97	10.54	9.77
GMM untrained, $K = 64$	233.41	153.49	106.58	71.06	41.19	21.54	13.59	10.60	9.66
GMM untrained, $K = 256$	218.41	141.49	99.99	66.99	38.61	20.94	13.01	**10.42**	9.73
GMM untrained, $K = 1024$	**186.94**	**123.86**	**93.61**	**63.60**	**37.12**	**19.91**	**12.81**	10.63	**9.63**

Table 4: CelebA-HQ 256, FID ↓ of the untrained model ($\mathbf{W}_\phi = \mathbf{0}$) for every mixture size K, 10k samples against the full training set, uniform Euler grid in $s(t)$. The $K = 1024$ row is the one of Table 2.

Euler steps	1	2	4	8	16	32	64	128	256
Gaussian	371.25	173.07	52.95	17.20	8.59	**5.47**	4.13	3.43	3.05
GMM untrained, $K = 4$	415.10	142.93	43.32	15.61	**8.27**	5.50	**4.01**	3.34	3.02
GMM untrained, $K = 16$	357.96	130.65	39.43	15.12	8.40	5.51	4.04	3.40	3.03
GMM untrained, $K = 64$	312.40	120.00	36.35	14.81	8.39	5.57	4.02	3.38	2.98
GMM untrained, $K = 256$	250.91	108.81	34.59	14.89	8.47	5.54	4.05	3.38	2.97
GMM untrained, $K = 1024$	197.89	95.62	32.97	14.59	8.49	5.58	4.05	3.34	**2.95**
GMM untrained, $K = 4096$	**151.43**	**85.62**	**31.36**	**14.49**	8.48	5.65	4.13	**3.32**	2.98

Table 5: CIFAR-10, FID ↓ of the untrained model ($\mathbf{W}_\phi = \mathbf{0}$) for every mixture size K, 50k samples against the full training set, uniform Euler grid in $s(t)$. The $K = 4096$ row is the one of Table 2.

	mean	std	min	p_1	p_5	median	max
CelebA-HQ, latent L^2							
GMM coupled	9.18	1.95	5.12	5.92	6.56	8.87	17.23
Gaussian	9.11	1.94	5.30	5.83	6.43	8.83	18.09
CIFAR-10, latent L^2							
GMM coupled	12.09	2.51	5.59	6.48	7.91	12.33	18.24
Gaussian	11.99	2.53	5.56	6.31	7.86	12.22	18.18
test set	11.66	2.65	2.37	5.69	7.37	11.65	20.48

Table 6: Distance of 1000 samples per model to their nearest training image in the Inception latent space. The CIFAR-10 test-set rows are a reference of novel images.



Figure 4: CIFAR-10: the mean of a single component (left) and seven samples of the coupled model conditioned on it, 256 Euler steps.

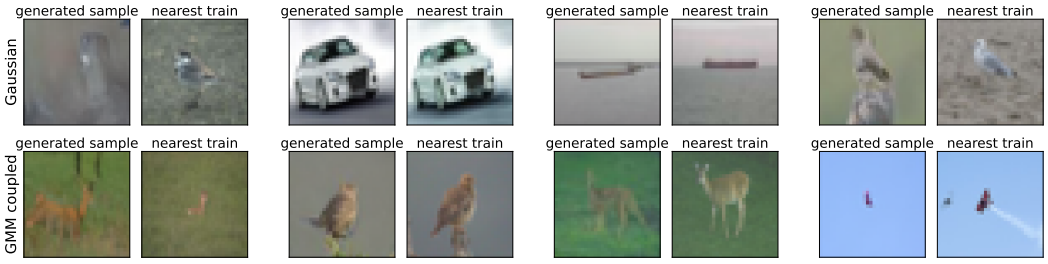


Figure 5: CIFAR-10: the four generated samples closest to the flip-symmetrized training set, out of 1000, by pixel L^2, each next to its nearest training image.