
ScoreFusion: Fusing Score-based Generative Models via Kullback–Leibler Barycenters

Hao Liu*
Stanford University

Junze Tony Ye*
Stanford University

Jose Blanchet
Stanford University

Nian Si
HKUST

Abstract

We introduce ScoreFusion, a theoretically grounded method for fusing multiple pre-trained diffusion models that are assumed to generate from auxiliary populations. ScoreFusion is particularly useful for enhancing the generative modeling of a target population with limited observed data. Our starting point considers the family of KL barycenters of the auxiliary populations, which is proven to be an optimal parametric class in the KL sense, but difficult to learn. Nevertheless, by recasting the learning problem as score matching in denoising diffusion, we obtain a tractable way of computing the optimal KL barycenter weights. We prove a dimension-free sample complexity bound in total variation distance, provided that the auxiliary models are well-fitted for their own task and the auxiliary tasks combined capture the target well. The sample efficiency of ScoreFusion is demonstrated by learning handwritten digits. We also provide a simple adaptation of a Stable Diffusion denoising pipeline that enables sampling from the KL barycenter of two auxiliary checkpoints; on a portrait generation task, our method produces faces that enhance population heterogeneity relative to the auxiliary distributions.

1 INTRODUCTION

Our goal in this paper is to propose and analyze a general method (which we call ScoreFusion) for fusing multiple pre-trained diffusion models that are assumed to simulate auxiliary populations.

There are multiple needs that motivate this goal. For example, it is well known that diffusion models rely on large datasets (often involving high-dimensional features) and there simply may not be enough data to train a diffusion model from a target population (Saharia et al., 2022; Wang et al., 2023; Kuznetsova et al., 2020; Schuhmann et al., 2022). Another motivation is that we may wish (at inference time) to sample from a region that has a low probability in the target population, but such a region may be targeted with the help of auxiliary models. As we will demonstrate, ScoreFusion addresses precisely these types of needs.

ScoreFusion starts from the idea that if the auxiliary populations are well chosen, then the target population could be well represented by some KL-weighted-barycenter of the auxiliary populations for a suitable choice of weights, which has an analytical closed form expression (Claici et al., 2020; Banerjee et al., 2005). The ScoreFusion method then tries to find the distribution that optimizes the fit to the target population within this parametric family, based on a limited number of samples. From a statistical standpoint, ScoreFusion reduces the problem of fitting a non-parametric distribution (a task that is extremely challenging to do with a limited sample size) to that of fitting a parametric family (a much more manageable task with a moderate sample size). Moreover, the parametric family is not arbitrary, but derived from a key optimality principle, namely, the KL-barycenter criterion.

Another common barycenter criterion that we could have used is the Wasserstein barycenter (Cuturi and Doucet, 2014; Peyré et al., 2016; Solomon et al., 2015; Claici et al., 2018; Janati et al., 2020). However, computing Wasserstein barycenters is generally challenging (Peyré and Cuturi, 2019; Benamou et al., 2015; Staib et al., 2017; Genevay et al., 2016). This is why we utilize the Kullback–Leibler (KL) barycenter.

Next, we proceed to optimize the weights of the KL-barycenter parametric family to minimize the empirical score-matching loss computed from a limited collection of target samples. Although this formulation is elegant—essentially reducing to a maximum likelihood

* Equal contribution. Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

estimation problem that is convex (as a function of the parametric weights, see Equation (9))—applying gradient descent poses challenges, particularly in high-dimensional settings, due to the need for complex numerical integration.

Fortunately, diffusion generative modeling is particularly useful in our setting, and the way we use them underscores a novel way to apply diffusion modeling. The success of diffusion models (Song et al., 2021b; Ho et al., 2020; Sohl-Dickstein et al., 2015) has sparked a well-developed infrastructure to train those models, with many pre-trained or fine-tuned checkpoints available on websites like Hugging Face and Civitai. We take advantage of these resources and show that if the auxiliary target populations have been fitted using them, then our learning task is greatly simplified. To be more precise, using auxiliaries trained with diffusion and solving Problem (10) is a simpler way of learning the KL-barycenter parametric family.

Our method can be viewed as an example of ensemble learning (Dietterich et al., 2002; Yang et al., 2024). We combine the idea of ensemble learning and score-based generative model. Moreover, our approach is optimal in the sense of KL divergence, and our weights can be easily learned on limited data. It is related to empirical methods like checkpoint merging for diffusion models (Automatic1111, 2022; HuggingFace, 2023b), which are not based on an optimality principle as we do (Equation (10)) and lack theoretical basis.

A Taylor expansion argument is presented in Biggs et al. (2024), which tries to connect interpolation in the parameter space (checkpoint merging) of the denoiser U-Net to that in its function space (ScoreFusion sampling). The catch is, the Taylor expansion argument is valid only if the parameters of the merging neural networks differ by small amounts. So our paper provides theoretical basis to checkpoint merging in the context of small parameter perturbations. But we also show in experiments that the KL-barycenter approach is different and can produce more heterogeneity in populations that have relatively low representation.

The sampling aspect of ScoreFusion is also related to concurrent methods of synthesizing semantic concepts to create novel images. For example, Richardson et al. (2024), Li et al. (2024), and Xiong et al. (2024) proposed different ways of combining concepts in text-to-image generation. The difference between their approaches and ours is the specific component of the generative pipeline where the interpolation happens: our KL barycenter controls interpolation in the probability space of observable outcomes (e.g. the pixel space, in the case of generative vision), whereas theirs combine concepts in the latent embedding spaces.

Our main contributions are summarized as:

- We demonstrate that KL barycenter fusion of auxiliary models can be efficiently implemented when the auxiliary models are trained by score matching. In this case, the optimal score is linear in the auxiliary scores.
- We provide generalization bounds which split the error into four components. First, the error between the optimal KL barycenter and the target at time zero (whose direct implementation is difficult due to numerical integration). The second term corresponds to the sample complexity $O(n^{-1/4})$ and the third term is the approximation error incurred from converting the training into a regression problem. The fourth component reflects the quality of auxiliary score estimations.
- We demonstrate empirically that ScoreFusion successfully addresses the needs that motivate our goal, as mentioned earlier. We demonstrate ScoreFusion’s sample efficiency on MNIST digits measured in both image fidelity and class proportions. Further, we show ScoreFusion’s ability to sample from low probability regions on the task of generating professional portraits.

The rest of the paper is organized as follows. Section 2 reviews the background of KL barycenter and diffusion models. Section 3 details our proposed fusion methods. Section 4 provides convergence results for our methods. Section 5 presents experiment results and comparisons with baseline methods. Finally, Section 6 concludes the paper with future directions. All proofs, code, and additional experimental details are relegated to the Supplementary Materials.

2 PRELIMINARIES AND SETUP

2.1 Notations

The following notation will be used. Given two functions $f, g : D \rightarrow \mathbb{R}$, we say $f \lesssim g$ if there exists a constant $C > 0$ such that for all $x \in D$, $f(x) \leq Cg(x)$. When $x \rightarrow a$, where $a \in [-\infty, \infty]$, we say $f(x) = \mathcal{O}(g(x))$ if there exists a constant $C > 0$ such that for all x close enough to a , $|f(x)| \leq Mg(x)$. In asymptotic cases, we use \mathcal{O} and \lesssim interchangeably. $f \sim g$ if and only if $f \lesssim g$ and $g \lesssim f$. $C([0, T] : \mathbb{R}^d)$ is the space of all continuous functions on \mathbb{R}^d equipped with the uniform topology. In this paper, we consider a Polish spaces S , which could be \mathbb{R}^d or $C([0, T] : \mathbb{R}^d)$. For a Polish space S equipped with Borel σ -algebra $\mathcal{B}(S)$, we denote $\mathcal{P}(S)$ as the space of probability measures on S equipped with the topology of weak convergence. In a normed vector space $(X, \|\cdot\|)$, $\|\cdot\|$ denotes

the corresponding norm. $\|\cdot\|_p$ denotes the standard L^p norm. Given a matrix \mathbf{A} , we use \mathbf{A}^T to denote its transpose. We denote $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_k)^T \in [0, 1]^k$. We use Δ_k to present the k -dimensional probability simplex, i.e., $\Delta_k = \{\boldsymbol{\lambda} \in [0, 1]^k : \sum_{i=1}^k \lambda_i = 1\}$.

2.2 Barycenter Problems and Kullback–Leibler Divergence

Given a set of probability measures $P_1, \dots, P_k \in \mathcal{P}(S)$ on a Polish space S and a measure of dissimilarity (e.g. a metric or a divergence) between two elements in $\mathcal{P}(S)$, D , we define the barycenter problem with respect to D and weight $\boldsymbol{\lambda}$ as the optimization problem $\min_{\mu \in \mathcal{P}(S)} \sum_{i=1}^k \lambda_i D(\mu, P_i) \quad \text{s.t. } \boldsymbol{\lambda} \in \Delta_k$, where P_1, \dots, P_k are called the reference measures. With a fixed choice of weight and reference measures, the solution of the barycenter problem is denoted as $\mu_{\boldsymbol{\lambda}}$.

Recall the definition of Kullback–Leibler (KL) divergence: suppose $P, Q \in \mathcal{P}(S)$, then $D_{\text{KL}}(P \| Q) = \int \log\left(\frac{dP}{dQ}\right) dP$ if $P \ll Q$ and $D_{\text{KL}}(P \| Q) = \infty$ otherwise; where $\frac{dP}{dQ}$ is the Radon-Nikodym derivative of P with respect to Q . In particular, if $S = \mathbb{R}^d$, P and Q are absolutely continuous random vectors (with respect to Lebesgue measure) in \mathbb{R}^d with densities p and q respectively, then $D_{\text{KL}}(P \| Q) = \int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$. If D is the KL divergence, we recover the KL barycenter problem (Claici et al., 2020). In fact, for any Polish space S , the KL barycenter problem is strictly convex hence has at most one solution.

2.3 Background on Diffusion Models

Our score fusion method depends the generative diffusion model driven by stochastic differential equations (SDEs) developed in Song et al. (2021b); Ho et al. (2020); Sohl-Dickstein et al. (2015). In this section, we review the background of generative diffusion model.

2.3.1 Forward Process

We begin with the unsupervised learning setup. Given an unlabeled dataset i.i.d. from a distribution p_0 , the forward diffusion process is defined as the differential form

$$dX(t) = f(t, X(t))dt + g(t)dW(t), X(0) \sim p_0, \quad (1)$$

where $f : \mathbb{R}_+ \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a vector-valued function, $g : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is a scalar function, and $W(t)$ denotes a standard d -dimensional Brownian motion. From now on, we assume the existence and denote by $p_t(x)$ the marginal density function of $X(t)$, and let $p_{t|s}(X(t)|X(s))$ be the transition kernel from $X(s)$ to

$X(t)$, for $0 \leq s \leq t \leq T < \infty$, where T is the terminal time for the forward process (time horizon). If $f(t, x) = -ax$ and $g(t) = \sigma$ with $a > 0$ and $\sigma > 0$, then Equation (1) becomes a linear SDE with Gaussian transition kernels

$$dX(t) = -aX(t)dt + \sigma dW(t), X(0) \sim p_0, \quad (2)$$

which is an Ornstein-Uhlenbeck (OU) process. If T is large enough, then p_T is close to $\pi \sim \mathcal{N}\left(\mathbf{0}, \frac{\sigma^2}{2a}\mathbf{I}\right)$, a Gaussian distribution with mean $\mathbf{0}$ (vector) and covariance matrix $\frac{\sigma^2}{2a}\mathbf{I}$. The forward process can be viewed as the following dynamic: given the data distribution, we gradually add noise to it such that it becomes a known distribution in the long run.

2.3.2 Backward Process (Denoising)

If we reverse a diffusion process in time, then under some mild conditions (see, for example, Cattiaux et al. (2022); Föllmer (1985)) which are satisfied for all processes under consideration in this work, we still get a diffusion process. To be more precise, we want to have a process \tilde{X} such that for $t \in [0, T]$, $\tilde{X}(t) = X(T-t)$. From the Fokker–Planck equation and the log trick (Anderson, 1982), the corresponding reverse process for Process (1) is

$$\begin{aligned} d\tilde{X}(t) &= g^2(T-t)\nabla \log p_{T-t}\left(\tilde{X}(t)\right) dt \\ &\quad - f(T-t, \tilde{X}(t))dt + g(T-t)dW(t), \tilde{X}(0) \sim p_T, \end{aligned} \quad (3)$$

where ∇ represents taking derivative with respect to the space variable x . We call the term $\nabla \log p_t(x)$ as the (Stein) score function. If the forward process is an OU process, then the reverse process is $\tilde{X}(0) \sim p_T$ and

$$d\tilde{X}(t) = \left(a\tilde{X}(t) + \sigma^2 \nabla \log p_{T-t}\left(\tilde{X}(t)\right)\right) dt + \sigma dW(t). \quad (4)$$

If the backward SDE can be simulated (which is typically done via Euler–Maruyama method, see details in Supplementary Material Section A.2), we can generate samples from the distribution p_0 . We can view simulating the backward SDE as the denoising step from pure noise to the groundtruth distribution.

2.3.3 Score Estimation

The only remaining task is score estimation for $\nabla \log p_t(x)$. There are many ways to achieve this, and some of them are equivalent up to constants that is independent of the training parameters. In this paper, we choose the time-dependent score matching loss used in Song et al. (2021a):

$$\mathcal{L}(\theta; \gamma) := \mathbb{E}_{t \sim \mathcal{U}[0, T]} \left[\gamma(t) \mathbb{E}_{X(t) \sim p_t} \left[\|s_{t,\theta}(X(t)) - \nabla \log p_t(X(t))\|_2^2 \right] \right], \quad (5)$$

where $\gamma : [0, T] \rightarrow \mathbb{R}_+$ is a weighting function, and $s_{t,\theta} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a score estimator $s_{t,\theta}$, usually chosen as a neural network. Then score estimation is done by the empirical loss using SGD (Li et al., 2023).

There are many performance measures. Suppose $D(\cdot, \cdot)$ is a measure of dissimilarity in $\mathcal{P}(S)$, then we say $D(\mu, \hat{\mu})$ is a generalization error with respect to D , where μ is the target distribution and $\hat{\mu}$ is the distribution of the generated samples.

Recently, several analysis about the generative properties of diffusion models has been done; however, even in the case of compactly supported target distributions and sufficient smoothness regularity, the basic diffusion model encounters the curse of dimensionality. Therefore, a large amount of target data is needed to generate high quality samples. For a detailed discussion, see Supplementary Material Section A.3.

3 KL BARYCENTER AND FUSION METHODS

In Section 3.1, we propose and analytically solve two types of KL barycenter problems. These solutions will lead to the development of our fusion methods, which is detailed in Section 3.2.

3.1 KL Barycenter Problems

Distribution-level KL barycenter. We first consider a simple case, where the KL barycenter problem is solved on an Euclidean space, i.e., $S = \mathbb{R}^d$. The barycenter obtained in this scenario is referred to as a distribution-level KL barycenter. Theorem 1 gives the analytical solution.

Theorem 1. Suppose $\{\mu_1, \dots, \mu_k\} \subset \mathcal{P}(\mathbb{R}^d)$ and for each $i = 1, \dots, k$, μ_i is absolutely continuous with respect to the Lebesgue measure, with densities p_1, \dots, p_k respectively. Then, the distribution-level KL barycenter μ_λ is unique with density $p_\lambda(x) = \frac{\prod_{i=1}^k p_i(x)^{\lambda_i}}{\int_{\mathbb{R}^d} \prod_{i=1}^k p_i(x)^{\lambda_i} dx}$.

Process-level KL barycenter. Our second barycenter problem is performed when the state space is the continuous-function space, i.e., $S = C([0, T] : \mathbb{R}^d)$. This context yields a process-level KL barycenter. When the underlying measures are represented by SDEs, we offer a closed-form solution for the process-level KL barycenter in Theorem 2.

Theorem 2. Suppose for each $i = 1, 2, \dots, k$, the i -th SDE has the form with $X_i(0) \sim \mu_i$,

$$dX_i(t) = a_i(t, X(t)) dt + \sigma(t) dW_i(t)$$

and has a unique strong solution. The law of solution of each SDE is denoted as $P_i \in \mathcal{P}(C([0, T] : \mathbb{R}^d))$. We

further assume, for each $i = 1, 2, \dots, k$, μ_i has an absolutely continuous density with respect to the Lebesgue measure and a_i uniformly bounded, then process-level KL barycenter can be represented as the SDE with $X(0) \sim \mu$ and

$$dX(t) = a(t, X(t)) dt + \sigma(t) dW(t),$$

where $a(t, x) = \sum_{i=1}^k \lambda_i a_i(t, x)$, μ is the distribution-level KL barycenter of reference measures μ_1, \dots, μ_k , and W is a standard Brownian motion.

In this paper, fusing k distributions is viewed as computing a KL barycenter with optimized weights. Given k well-trained reference generative models, our fusing method optimizes the weights $\lambda_1, \dots, \lambda_k$ to approximate a target distribution.

3.2 Fusion Methods

Recall that in our task, we are given k datasets with abundant samples, and our goal is to generate samples for a target dataset (with limited available data). Therefore, in this section, we denote the target measure as ν and we assume that we are given k reference diffusion generative models and they are able to generate samples from k different reference measures μ_1, \dots, μ_k , respectively. Specifically, each reference measure corresponds to an auxiliary backward diffusion process with $\tilde{X}_i(0) \sim p_T^i$ and

$$d\tilde{X}_i(t) = \left(a\tilde{X}_i(t) + \sigma^2 s_{T-t, \theta^*}^i(\tilde{X}_i(t)) \right) dt + \sigma dW_i(t), \quad (6)$$

where s_{T-t, θ^*}^i is a well-trained score function for the i -th reference measure. we introduce two fusion algorithms and related generalization error results.

In practice, the discretized version of the SDE (6) is used. Specifically, we employ a small time-discretization step h and a total of N time steps (hence $T = Nh$). Since p_T^i is close to the Gaussian distribution π , the SDE (6) is approximated by $\hat{X}(0) \sim \pi$ and for all $t \in [lh, (l+1)h]$,

$$d\hat{X}_i(t) = \left(a\hat{X}_i(t) + \sigma^2 s_{T-lh, \theta^*}^i(\hat{X}_i(lh)) \right) dt + \sigma dW(t). \quad (7)$$

As shown in Li et al. (2023) Lemma 1, when the target distribution is compactly supported, then with high probability, the trajectory is also compactly supported, thus the score term is uniformly bounded. Thus, given a weight λ , Theorem 2 implies that the corresponding process-level KL barycenter follows the SDE: for all

$t \in [lh, (l+1)h]$, $\hat{Y}(0) \sim \pi$,

$$\begin{aligned} d\hat{Y}(t) &= \sigma^2 \sum_{i=1}^k \lambda_i s_{T-lh,\theta^*}^i(\hat{Y}(lh)) dt \\ &+ a\hat{Y}(t)dt + \sigma dW(t). \end{aligned} \quad (8)$$

We denote the distribution of the terminal variable $\hat{Y}(T)$ as \hat{p}_{λ} , which will later serve as the distribution of generated sample.

The key component in our diffusion method is to find an optimal λ^* such that the \hat{p}_{λ^*} is as close as the target measure ν as possible. To achieve this goal, we propose two fusion methods that relies on two different optimization problems.

The first method directly optimizes on the probability measure defined on the Euclidean space, which is based on Theorem 1. We notice that a similar idea of fusing component distributions via KL barycenter compared with vanilla fusion has been proposed in Claić et al. (2020), which uses averaging KL divergence as a metric to recover the mean-field approximation of posterior distribution of the fused global model.

Namely, we consider the following convex problem

$$\begin{aligned} \min_{\lambda \in \Delta_k} D_{\text{KL}}(\nu \| \mu_{\lambda}) \quad (9) \\ = \min_{\lambda \in \Delta_k} \left\{ \mathbb{E}_{\nu} \left[\log q(X) - \sum_{i=1}^k \lambda_i \log p_i(X) \right] \right. \\ \left. + \log \left(\int \prod_{i=1}^k p_i(y)^{\lambda_i} dy \right) \right\}, \end{aligned}$$

where p_1, \dots, p_k denote the densities of the reference measures and $q(x)$ denote the density of target distribution ν . We refer to this fusion method as *vanilla fusion*. Suppose we have an accurate estimation of the densities p_i s. We then use Frank-Wolfe method to solve Problem (9) and get an optimal λ^* . In the Frank-Wolfe method, the gradient term can be approximated by sample mean estimators from target data ν (See Remark 2 in Supplementary Material Section C.1.3.). To generate samples, we plug in the λ^* to (8) and simulate the SDE. However, the diffusion generative models usually cannot directly estimate the densities p_1, \dots, p_k . Therefore, for complicated high-dimension distributions, it is usually hard to directly apply vanilla fusion. Therefore, we propose a practical alternative, process-level method called **ScoreFusion**. The numerical results in Section 5 were generated by employing Algorithm 1.

In our second method, we first build a forward process starting from the target dataset, according to (2). We denote this forward process as $\tilde{X}^{\nu}(t)$ and the corresponding density as $p_t^{\nu}(x)$. Then, we modify the loss

function (5) as a linear regression problem

$$\begin{aligned} \tilde{\mathcal{L}}(\lambda; \theta^*, \gamma) &= \mathbb{E}_{t \sim \mathcal{U}[0, \tilde{T}]} \left[\gamma(t) \left(\mathbb{E}_{\tilde{X}^{\nu}(t) \sim p_t^{\nu}} \left[\right. \right. \right. \\ &\left. \left. \left. \left\| \sum_{i=1}^k \left(\lambda_i s_{t,\theta^*}^i(\tilde{X}^{\nu}(t)) \right) - \nabla \log p_t^{\nu}(\tilde{X}^{\nu}(t)) \right\|_2^2 \right] \right], \end{aligned} \quad (10)$$

where we choose $\tilde{T} \ll T$. The intuition behind the choice of \tilde{T} is that we want to learn an optimal λ^* such that p_{λ^*} is close to the target ν . Therefore, when $\tilde{T} \ll T$ (the forward process has not inject much noise), the $\hat{\lambda}$ obtained from the training is affected less by the noise. Theoretically, choosing $\tilde{T} = 0$ is optimal, but this is hard to implement. Algorithm 1 with $\tilde{T} = 0$ can be viewed as a variant of vanilla fusion since the learning is only performed on the distribution level (p_0), and extremely small \tilde{T} causes numerical instability in practice, which makes sense given the numerical integration and density estimations needed in the vanilla fusion. The optimization problem associated with our second method is $\min_{\lambda \in \Delta_k} \tilde{\mathcal{L}}(\lambda; \theta^*, \gamma)$. The details are in Algorithm 1.

Algorithm 1 ScoreFusion

- 1: **Input:** Calibration data \mathcal{D} , pre-trained score functions $s_{t,\theta^*}^1, \dots, s_{t,\theta^*}^k$. Hyperparameter \tilde{T} .
 - 2: **Output:** Samples from a distribution $\hat{\nu}_T$.
 - 3: **I. Calibration Phase**
 - 4: Randomly initialize non-negative $\lambda_1, \dots, \lambda_k$ s.t. $\sum \lambda_i = 1$.
 - 5: **repeat**
 - 6: Run forward process $\tilde{X}^{\nu}(t)$ using a mini-batch from \mathcal{D} .
 - 7: Evaluate the loss function (10) and back-propagate onto $\lambda_1, \dots, \lambda_k$.
 - 8: ▷ λ_i 's are softmaxed to enforce the probability simplex constraint
 - 9: **until** converged. Save the optimal $\lambda^* = \{\lambda_1^*, \lambda_2^*, \dots, \lambda_k^*\}$.
 - 10: **II. Sampling Phase**
 - 11: $s_{t,\lambda^*}(\hat{Y}(t)) := \sum_{i=1}^k \lambda_i^* s_{t,\theta^*}^i(\hat{Y}(t))$.
 - 12: Simulate the backward SDE (8) with $s_{t,\lambda^*}(\cdot)$ starting from a Gaussian prior and generate samples.
-

4 CONVERGENCE RESULTS

This section details the convergence results for our proposed fusion methods. We focus on sample complexities, quantified by the necessary number of samples in the target dataset, in terms of total variation distance. We show that the sample complexities of our methods are dimension-free, given that the auxiliary

processes are accurately fitted to their reference distributions and together offer adequate information for the target distribution. To begin with, we assume all distributions are compactly supported.

Assumption 1. *The target and reference distributions are all compactly supported in $\mathbb{K} \subset \mathbb{R}^d$ with absolutely continuous densities. We assume that their second moments are bounded by $M \in (0, \infty)$.*

Proposition 1. *Under Assumption 1, Problem (9) is convex in λ .*

Proposition 1 implies that Problem (9) is easy to solve given that the reference densities can be estimated. We further require Assumption 2 below, which guarantees that each auxiliary process is accurately trained in the sense that the score function at each time step is well-fitted.

Assumption 2. *For each $1, 2, \dots, k$ and for all $t \in [0, T]$, $\nabla \log p_t^i$ is L -Lipschitz with $L \geq 1$, the step size $h = T/N$ satisfies $h \lesssim 1/L$, and the inverse of $\nabla \log p_t^i$ is also Lipschitz; for each $1, \dots, k$ and $l = 0, 1, \dots, N$, $\mathbb{E}_{p_{lh}^i} \left[\left\| s_{lh, \theta^*}^i - \nabla \log p_{lh}^i \right\|_2^2 \right] \leq \epsilon_{score}^2$ with small ϵ_{score} .*

Assumption 2 is widely used in the diffusion model literature (see, for example, Chen et al. (2023)).

To proceed, we denote λ^* and Λ^* to be the solutions of Problems 9 and 10, respectively. The corresponding barycenters are denoted as μ_{λ^*} and μ_{Λ^*} . Assumption 3 below states that the theoretical optimal barycenters are close to the target measure, which ensures all reference distributions together are able to provide sufficient information for the target distribution.

Assumption 3. $D_{KL}(\nu \parallel \mu_{\lambda^*}) \leq \epsilon_0^2$ and $D_{KL}(\nu \parallel \mu_{\Lambda^*}) \leq \epsilon_1^2$, with small ϵ_0 and ϵ_1 .

Based on Assumptions 1, 2 and 3, we provide convergence results for the vanilla fusion and ScoreFusion (Algorithm 1) in Theorems 3 and 4, respectively.

Theorem 3. *Suppose that Assumptions 1, 2, and 3 are satisfied. We further assume for each fixed $\lambda \in \Delta_k$, $TV(\mu_\lambda, \hat{\mu}_\lambda) \leq \epsilon_2$, where $\hat{\mu}_\lambda$ is the barycenter of the output distributions of k auxiliary processes. Then, for $\delta > 0$ and $\delta \ll 1$, the output distribution of the vanilla fusion method, $\hat{\nu}_D$, we have with probability at least $1 - \delta$,*

$$TV(\nu, \hat{\nu}_D) \lesssim \underbrace{\epsilon_0}_{\text{quality of combined auxiliaries}} + \underbrace{\epsilon_2}_{\text{auxiliary density estimation}} + \underbrace{\mathcal{O} \left(\left(\log \left(\frac{1}{\delta} \right) \right)^{1/4} n^{-1/4} \right)}_{\text{mean estimation error}} + SE,$$

where SE is the error of auxiliary score estimation, defined as

$$SE := \left[\exp(-T) \max_{i=1,2,\dots,k} \sqrt{D_{KL}(p_T^i \parallel \pi)} + \sigma \sqrt{kT} (\epsilon_{score} + L\sqrt{dh} + Lh\sqrt{M}) \right].$$

Theorem 4. *Suppose that Assumptions 1, 2, and 3 are satisfied. Then, for $\delta > 0$ and $\delta \ll 1$, for the output distribution of Algorithm 1, $\hat{\nu}_P$, with probability at least $1 - \delta$,*

$$TV(\nu, \hat{\nu}_P) \lesssim \underbrace{(\sigma + 1)\epsilon_1}_{\text{quality of combined auxiliaries}} + \underbrace{\sigma \sqrt{k} \mathcal{O}(\tilde{T}^{1/4})}_{\text{approximation of time 0}} + \underbrace{\mathcal{O} \left(\sigma \left(\log \left(\frac{1}{\delta} \right) \right)^{1/4} n^{-1/4} \right)}_{\text{sampling error}} + SE.$$

Theorems 3 and 4 demonstrate dimension-free sample complexities given that auxiliaries are well approximated and auxiliaries all combined capture the features of target well. More specifically, each bound in Theorems 3 and 4 has 4 terms, which represents different sources of error.

The quality of combined auxiliaries is the essential assumption in both Theorems 3 and 4. The sampling error in Theorem 4 reflects the fact that with the help of diffusion models, the optimization in fact becomes linear in terms of scores, making the problem easier and escape the curse of dimensionality. The approximation to time $t = 0$ term replaces the vanilla fusion with a small controllable noise but makes the implementation much easier. It worth noticing that there is a tradeoff between choosing \tilde{T} : the smaller \tilde{T} , the more accurate the optimal weights are, but the more probably that the algorithm will encounter numerical instability. Finally, the score estimation term of the auxiliaries can be small with a careful choice of discretization time steps and accurate auxiliary score approximation (see Remark 3 in Supplementary Material Section C.2).

5 EXPERIMENTS

In this section, we use images to illustrate some of the key features highlighted in our contributions (third bullet point) in the Introduction. Additional conceptual experiments are given in the Supplementary Material. We release our code at <https://github.com/junzeye/ScoreFusion>.

5.1 Calibrating MNIST Digits Distribution

EMNIST (Cohen et al., 2017) is an enriched version of the well-known MNIST dataset in 1x28x28 grayscale. We selected five non-overlapping subsets ($D_i, i = 1, \dots, 5$), composed entirely of the digits 7 and 9 but with varying mixtures: (10%, 90%), (30%, 70%), (70%, 30%), (90%, 10%), and (60%, 40%), respectively. Four auxiliary score networks are trained from scratch on D_1, \dots, D_4 with enough data to ensure adequate validation loss convergence. D_5 is reserved as the target dataset for calibrating (finetuning) both ScoreFusion and the baseline models. In our experiment, we vary the quantity of class-balanced samples from D_5 made available for the model to learn.

Our first baseline method trains a score network from scratch using only limited target data, illustrating the difficulty of density estimation in the low-data regime. Our second baseline directly fine-tunes (with early-stopping) one of the pre-trained auxiliary models on the target data by unfreezing its checkpoint; the chosen auxiliary model was pre-trained on D_3 , whose digits mix (70-30) is the closest to D_5 in a Wasserstein sense. Lastly, the KL barycenter weights of ScoreFusion are obtained by optimizing exclusively a linear projection layer on top of the four auxiliary score networks.

Table 1: Mean NLL (bits/dim) under different sample sizes. A smaller value is better.

Sample size	2^6	2^8	2^{10}	2^{12}
B1	7.186 ± 0.019	6.235 ± 0.016	5.725 ± 0.024	4.979 ± 0.028
B2	4.779 ± 0.042	4.786 ± 0.034	4.769 ± 0.032	4.763 ± 0.034
Frozen B2		4.768 ± 0.024		
Ours	4.733 ± 0.029	4.733 ± 0.018	4.718 ± 0.022	4.715 ± 0.021

We evaluate both fidelity and diversity of samples generated by each method at inference time. Fidelity is measured by the negative log likelihood (NLL) of a holdout test dataset under the trained model (Ho et al., 2020). To quantify digits distribution, we use a high-accuracy digits classifier (Kabir et al., 2022) to classify 1024 samples generated by each method.

Tables 1 and 2 show summary statistics of our evaluations. We relegate training details and sample images to the Appendix. Across all four tested sample sizes, ScoreFusion achieves a lower NLL than the other two baselines. Moreover, despite the alignment of digits proportions not being hard-coded in the score matching loss minimization, Table 2 shows that with as few as 64 samples, ScoreFusion already learns a generative model whose digits proportions closely align with the ground-truth 60-40 split. At the same time, Baseline 2 is slow to calibrate its digits proportions even as we increase the fine-tuning data size.

5.2 Sampling a New Facial Distribution

SDXL 1.0 is the newest model in the Stable Diffusion family (Stability-AI, 2024), capable of generating realistic images in 1024x1024 resolution. We downloaded two fine-tuned SDXL checkpoints to use them as our auxiliary models (AlDahoul et al., 2024a,b). These two models were each finetuned by their creators to generate human portraits who look like White males and East Asian females, respectively, in Figure 1. For consistency, we use the same text prompt in all images generations: “*a photo of a mathematics scientist, looking at the camera, ultra quality, sharp focus*”. Importantly, this prompt only specifies a person’s profession, leaving their other traits unspecified.



Figure 1: **Top:** Generations by the 1st auxiliary model alone, resembling the White male phenotype. **Bottom:** Generations by the 2nd auxiliary model, resembling the Asian female phenotype. These are model generations **without** using the KL barycenter sampler.

Since the MNIST experiment already tests the sample efficiency of training with ScoreFusion, we focus on investigating its inference-time behavior in the SDXL experiment. Specifically, we hardcode the barycenter weights $\lambda \in \Delta^1$, use the same prompt as before, and sample images from the KL barycenter distribution of the two auxiliary models. As a baseline, we also sampled images from a model created from linearly merging the checkpoints of the two auxiliary models, as is the empirical practice in Automatic1111 (2022); Biggs et al. (2024).

The side-by-side in Figure 2 suggests a caveat of Biggs et al. (2024)’s Taylor expansion argument to approximate interpolation in the parameter space of the score network by the function space. Samples on the left are visibly different from those on the right, indicating that the error term introduced by Taylor approximation amplifies non-trivially after denoising diffusion. The observation lends further evidence that our approach fuses a distribution meaningfully distinct from that of weight-averaging two auxiliary models.

We also conduct an ablation study on the samples’ sensitivity to λ . Figure 3 qualitatively compares how

Table 2: Class proportions of 1024 handwritten digits generated by each differently trained model, estimated by a high-accuracy MNIST classifier. “B1” and “B2” refer to Baselines 1 and 2 respectively. “Others” are samples assigned to a class that is neither 7 nor 9. $2^6, \dots, 2^{12}$ refer to number of training data from the target distribution.

Digit	Target	2^6		2^8		2^{10}		2^{12}				
		B1	B2	Ours	B1	B2	Ours	B1	B2	Ours	B1	B2
7	60%	47.9%	72.4%	55.6%	66.8%	65.5%	57.5%	65.5%	65.1%	56.6%	66.7%	65.5%
9	40%	10.3%	23.2%	39.4%	23.8%	29.9%	38.0%	26.7%	30.6%	39.8%	27.9%	30.4%
Others	0	41.8%	4.4%	5.0%	9.4%	4.6%	4.5%	7.8%	4.3%	3.6%	5.4%	4.1%



Figure 2: **Left:** i.i.d. samples from KL barycenter. **Right:** i.i.d. samples from checkpoint merging. Interpolation weights are $\lambda = (0.5, 0.5)$ for both. The same text prompt as Figure 1 was used: “*a photo of a mathematics scientist, looking at the camera, ultra quality, sharp focus*”. Both approaches enhance ethnic diversity relative to the monolithic representations in Figure 1, but the KL barycenter approach also produces samples that embody a more ambiguous and rarer representation of gender and ethnicity, mitigating stereotypes.

KL barycenter and checkpoint merging respectively responds to variations in λ . The KL barycenter shows a smooth spectrum of gender expressions and physical appearances that move beyond stereotypical portrayals, embodying qualities that may be read as gender-neutral. This distinction highlights our approach’s ability to encourage the sampling from the *tails* of the auxiliary distributions, which one could not efficiently obtain by resampling from the auxiliary models.

Besides visual evaluations, we quantify gender and ethnic representations by using the CLIP encoder as a human surrogate (Radford et al., 2021), running it on 512 image samples generated by each method. Figure 4 shows a joint 2D scatter plot of the CLIP distances of samples of different generators to two opposite gender concepts, confirming the qualitative comment for Figure 2. Figure 5 shows a KDE-smoothed CLIP scatter plot for the ablation study, complementing Figure 3. Implementation details, more image samples and CLIP

plots for semantic probing are given in Appendix G.

6 CONCLUSION

In this paper, we propose an ensemble method based on KL barycenter that can be easily implemented if the auxiliary score estimations are obtained from diffusion. Our method not only simplifies a parametric training in the low data regime, but also contributes a mathematically grounded algorithm for visual generative models. We provide a theoretical analysis of the sample complexity, showing that it is dimension-free given accurate auxiliary score estimation and closeness between optimal KL barycenter and the target distribution. The numerical experiments further demonstrate that our method performs well in the low data regime and show the difference between our method and checkpoint merging. This method can also extend to other gradient flow variants, which we leave for future work.



Figure 3: **Top row:** KL barycenter. **Bottom row:** Checkpoint merging.

The same Gaussian noise was used to seed all twelve images, the only difference being the interpolation approach (top vs bottom) and interpolation weight; from left to right, $\lambda_2 \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ and $\lambda_1 = 1 - \lambda_2$. $\lambda_2 = 0$ and $\lambda_2 = 1$ each reduce to an original auxiliary (biased) SDXL model. Observe that the bottom row samples show an abrupt identity shift between $\lambda_2 = 0.2$ and 0.4 , whereas the top row shows a smoother transition from one demographic visual concept to another.

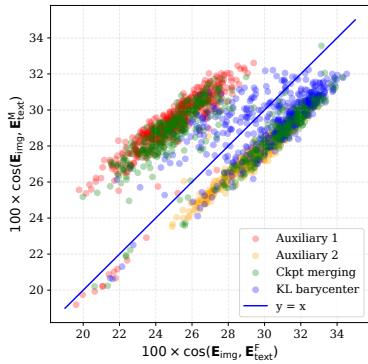


Figure 4: Empirical distribution of each model, projected onto a 2D gender semantic space. x and y coordinates are their CLIP scores. E_{img} are CLIP embeddings of each sample. E_{text}^F and E_{text}^M are embeddings of “*a photo of a female scientist*” and “*a photo of a male scientist*”. Gender neutrality and diversity can be interpreted as the middle region between the two auxiliary models’ unimodal clusters; KL barycenter samples from this unexplored region, whereas checkpoint merging induces a bimodal mixtures distribution.

Acknowledgements

The material in this paper is partly supported by the Air Force Office of Scientific Research under award number FA9550-20-1-0397 and ONR N000142412655. Support from NSF 2229012, 2312204, 2403007 is also gratefully acknowledged. J.Y. would like to thank Michael Y. Li, Yuhui Zhang, and Zeyu Wang for helpful feedback on the experiments. This work was partly

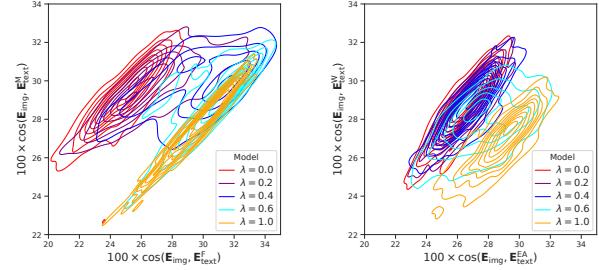


Figure 5: KDE contours of the KL barycenter distribution under various λ_2 (denoted as λ in legends) values, estimated using a bandwidth of 0.8. **Left:** E_{text}^F , E_{text}^M are text embeddings of “*a photo of a female scientist*” and “*a photo of a male scientist*”. **Right:** E_{text}^A , E_{text}^W are text embeddings of “*a photo of an East Asian scientist*” and “*a photo of a White scientist*”.

conducted using the Stanford Yen Cluster, provided by the Data, Analytics, and Research Computing (DARC) group at the Stanford Graduate School of Business. We are also grateful to the anonymous reviewers for suggesting numerous improvements.

References

- Nouar AlDahoul, Talal Rahwan, and Yasir Zaki. Asian female profession model, 2024a. URL https://huggingface.co/NYUAD-ComNets/Asian_Female_Profession_Model.
- Nouar AlDahoul, Talal Rahwan, and Yasir Zaki. White male profession model, 2024b. URL https://huggingface.co/NYUAD-ComNets/White_Male_Profession_Model

https://huggingface.co/NYUAD-ComNets/White_Male_Profession_Model.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows*. Springer Science & Business Media, 2005.

Brian D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Automatic1111. Stable Diffusion Web UI, August 2022. URL <https://github.com/AUTOMATIC1111/stable-diffusion-webui>.

Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382, 2005.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

Benjamin Biggs, Arjun Seshadri, Yang Zou, Achin Jain, Aditya Golatkar, Yusheng Xie, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Diffusion soup: Model merging for text-to-image diffusion models. *arXiv preprint arXiv:2406.08431*, 2024.

Gábor Braun, Alejandro Carderera, Cyrille W. Combettes, Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Sebastian Pokutta. Conditional gradient methods. *arXiv preprint arXiv:2211.14103*, 2022.

Patrick Cattiaux, Giovanni Conforti, Ivan Gentil, and Christian Léonard. Time reversal of diffusion processes under a finite entropy condition. *arXiv preprint arXiv:2104.07708*, 2022.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R. Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.1121*, 2023.

Jaemin Cho, Abhay Zala, and Mohit Bansal. Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3043–3054, 2023.

Sebastian Claici, Edward Chien, and Justin Solomon. Stochastic wasserstein barycenters. In *International Conference on Machine Learning*, pages 1141–1150, 2018.

Sebastian Claici, Mikhail Yurochkin, Soumya Ghosh, and Justin Solomon. Model fusion with kullback-leibler divergence. In *International Conference on Machine Learning*. PMLR, 2020.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693, 2014.

Thomas G Dietterich et al. Ensemble learning. *The handbook of brain theory and neural networks*, 2(1):110–125, 2002.

Lawrence C. Evans. *Partial Differential Equations*. American Mathematical Society, 2010.

H Föllmer. An entropy approach to the time reversal of diffusion processes. *Stochastic differential systems (Marseille-Luminy, 1984)*, 69:156–163, 1985.

I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Dover Publications, 2000.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. *Advances in Neural Information Processing Systems*, 29:3432–3440, 2016.

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

Daniel Hsu, Clayton Sanford, and Rocco A. Servedio Emmanouil V. Vlatakis Gkaragkounis. On the approximation power of two-layer networks of random relus. In *Proceedings of Machine Learning Research*, volume 134, pages 1–39. 34th Annual Conference on Learning Theory, 2021.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

HuggingFace. Load loras for inference.
https://huggingface.co/docs/diffusers/main/en/tutorials/using_peft_for_inference, 2023a.
Accessed: 2024-10-14.

HuggingFace. Merge loras, 2023b. URL
https://huggingface.co/docs/diffusers/main/en/using-diffusers/merge_loras. Accessed:
2024-09-28.

Sebastian Jaimungal and Silvana M. Pesenti.
Kullback-leibler barycentre of stochastic processes.
arXiv preprint arXiv:2407.04860v1, 2024.

Hicham Janati, Marco Cuturi, and Alexandre Gramfort. Debiased sinkhorn barycenters. In *International Conference on Machine Learning*, pages 4692–4701, 2020.

HM Dipu Kabir, Moloud Abdar, Abbas Khosravi, Seyed Mohammad Jafar Jalali, Amir F Atiya, Saeid Nahavandi, and Dipti Srinivasan. Spinalnet: Deep neural network with gradual input. *IEEE Transactions on Artificial Intelligence*, 2022.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. In *International Journal of Computer Vision. ICCV*, 2020.

Jun Li, Zedong Zhang, and Jian Yang. TP2O: Creative text pair-to-object generation using balance swap-sampling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Puheng Li, Zhong Li, Huishuai Zhang, and Jiang Bian. On the generalization properties of diffusion models. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. NeurIPS, 2023.

Sasha Luccioni, Christopher Akiki, Margaret Mitchell, and Yacine Jernite. Stable bias: Evaluating societal representations in diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, pages 2664–2672, 2016.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL
<https://arxiv.org/abs/2103.00020>.

Elad Richardson, Kfir Goldberg, Yuval Alaluf, and Daniel Cohen-Or. Conceptlab: Creative concept generation using vlm-guided diffusion prior constraints. *ACM Transactions on Graphics*, 43(3):34:1–34:14, 2024. doi: 10.1145/3659578.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, , and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.

Chitwan Saharia, William Chan, Saurabh Saxena†, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems 36*. NeurIPS, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *Proceedings of the 32nd International Conference on Machine Learning*, 37:2256–2265, 2015.

Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, and Leonidas Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics*, 34(4):66, 2015.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2021)*. NeurIPS, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modelling through stochastic differential equations. In *International Conference on Learning Representations (ICLR 2021)*. ICLR, 2021b.

Stability-AI. Announcing sdxl 1.0, 2024. URL <https://stability.ai/news/stable-diffusion-sdXL-1-announcement>.

Matthew Staib, Sebastian Claici, Justin Solomon, and Stefanie Jegelka. Parallel streaming wasserstein barycenters. In *Advances in Neural Information Processing Systems*, volume 30, pages 2647–2658, 2017.

Ramon van Handel. Probability in high dimension, apc 550 lecture notes, December 2016.

Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, and Mingyuan Zhou. Patch diffusion: Faster and more data-efficient training of diffusion models. *arXiv preprint arXiv:2304.12526*, 2023.

Zeren Xiong, Zedong Zhang, Zikun Chen, Shuo Chen, Xiang Li, Gan Sun, Jian Yang, and Jun Li. Novel object synthesis via adaptive text-image harmony. In *Advances in Neural Information Processing Systems*, volume 37, pages 139085–139113, 2024.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*, 2024.

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]

- (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
- (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes]
 - (b) Complete proofs of all theoretical results. [Yes]
 - (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Yes]
 - (b) The license information of the assets, if applicable. [Yes]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
 - (d) Information about consent from data providers/curators. [Not Applicable]
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
 - (a) The full text of instructions given to participants and screenshots. [Not Applicable]
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

Section A gives additional information about the basic diffusion models, including the time reversal formulas, time discretization steps, and the current generalization error results. B and C provide missing proofs in Section 3.1 and 4 of the main text, respectively. D discusses a related work in quantitative finance. E clarifies some experiment notations. F provides implementation details and additional results of the MNIST experiment. G provides implementation details of the Stable Diffusion experiment, including the CLIP distance computation (G.2.4), the ablation study setup (G.3), more image samples (G.4), and a heuristic mathematical explanation of the observation (G.5). Lastly, H provides results from a stylized Gaussian mixtures experiment.

A MORE ABOUT BASIC DIFFUSION MODELS

A.1 About the Time Reversal Formula

Note that Equations (3) and (4) are still represented as a “forward” processes. If we replace $W(t)$ by $\tilde{W}(t)$, where $\tilde{W}(t)$ is a standard d -dimensional Brownian motion which flows backward from time T to 0, then Equation (3) becomes

$$d\hat{X}(t) = \left(f(T-t, \hat{X}(t)) - g^2(T-t) \nabla \log p_{T-t}(\hat{X}(t)) \right) dt + g(T-t) d\tilde{W}(t), \hat{X}(T) \sim p_T,$$

which is the reverse SDE presented in Song et al. (2021b). Hence for the forward OU process, the reverse process has another representation by

$$d\hat{X}(t) = \left(-a\hat{X}(t) - \sigma^2 \nabla \log p_{T-t}(\hat{X}(t)) \right) dt + \sigma d\tilde{W}(t), \hat{X}(T) \sim p_T. \quad (11)$$

A.2 Discretization and Backward Sampling

In this section, we follow the scheme in Chen et al. (2023).

Given n samples $X_0^{(1)}, \dots, X_0^{(n)}$ from p_0 (data distribution), we train a neural network with the loss function (5). Let $h > 0$ be the step size of the time discretization, and there are N steps, hence $T = Nh$. We assume that for each time $l = 0, 1, \dots, N$, the score estimation s_{lh, θ^*} of $\nabla \log p_t$ is obtained. In order to simulate the reverse SDE (3), we first replace the score function $\nabla \log p_{T-t}$ with the estimate s_{T-t, θ^*} . Next, for each $t \in [lh, (l+1)h]$, the value of this coefficient in the SDE at time lh , which yields the new time-discretized SDE with each $t \in [lh, (l+1)h]$,

$$d\hat{X}(t) = \left(-f(T-t, \hat{X}(t)) + g^2(T-t) s_{T-t, \theta^*}(\hat{X}_{kh}) \right) dt + g(T-t) dW(t) \quad (12)$$

and $\hat{X}(0) \sim \Pi$, where Π is the (theoretical) stationary distribution of the forward process (1).

There are several details in this implementation. In practice, when we use OU process as the forward, then Equation (12) becomes

$$d\hat{X}(t) = \left(a\hat{X}(t) + \sigma^2 s_{T-t, \theta^*}(\hat{X}_{kh}) \right) dt + \sigma dW(t), t \in [lh, (l+1)h],$$

with $\Pi = \pi$, which is a linear SDE. In particular, $X_{(l+1)h}$ conditioned on X_{lh} is Gaussian, so the sampling is easier.

In theory, we should use $\Pi \sim p_T$, which we have no access to. The above implementation takes advantage of $p_T \approx \Pi$ as T is large enough. This introduces a small initialization error.

A.3 About the Generalization Error of Basic Diffusion Model

In Li et al. (2023), a random feature model is considered as the score estimator. The basic intuition is that the generalization error with respect to the KL divergence, $D_{\text{KL}}(\mu \parallel \hat{\mu})$ is decomposed into three terms: the training error, approximation error of underlying random feature model, and the convergence error of stationary measures. Among these three, the third one is ignorable since the fast rate of convergence of an OU process (or, from log Sobolev inequality for Gaussian random variables in van Handel (2016)). The first one is also small since random feature model in this setting is essentially linear regression with least squares.

Moreover, as stated in Hsu et al. (2021), random feature model can approximate Lipschitz functions with compact supports. However, the approximation error can be large and cause curse of dimensionality if we choose $m \sim n$. To illustrate this, we make a more general statement including smoothness considerations.

To be more precise, we introduce the following setting. We use the basic diffusion model with a forward OU process. The score function $s_{t,\theta}(x)$ is parameterized by the random feature model with m random features:

$$s_{t,\theta}(x) = \frac{1}{m} A \sigma(Wx + Ue(t)) = \frac{1}{m} \sum_{j=1}^m a_j \sigma(w_j^T x + u_j^T e(t)),$$

where σ is the ReLU activation function, $A = (a_1, \dots, a_m) \in \mathbb{R}^{d \times m}$ is the trainable parameters, $W = (w_1, \dots, w_m)^T \in \mathbb{R}^{m \times d}$, $U = (u_1, \dots, u_m)^T \in \mathbb{R}^{m \times d_e}$ are initially sampled from some pre-chosen distributions (related to random features) and remain frozen during the training, and $e : \mathbb{R}_+ \rightarrow \mathbb{R}^{d_e}$ is the time embedding function. The precise description is given below.

Assume that a_j, w_j , and u_j are drawn i.i.d. from a distribution ρ , then as $m \rightarrow \infty$, from strong law of large numbers, with probability 1,

$$s_{t,\theta}(x) \rightarrow \bar{s}_{t,\bar{\theta}}(x) = \mathbb{E}_{(w,u) \sim \rho_0} [a(w,u) \sigma(w^T x + u^T e(t))], \quad (13)$$

where $a(w,u) = \frac{1}{\rho_0(w,u)} \int a \rho(a,w,u) da$ and $\rho_0(w,u) = \int \rho(a,w,u) da$. From the positive homogeneity of ReLU function, we may assume $\|u\| + \|w\| \leq 1$. The optimal solution is denoted by $\bar{\theta}^*$ when replacing $s_{t,\theta}(x)$ in loss objective with $\bar{s}_{t,\bar{\theta}}(x)$.

Define a kernel $K_{\rho_0}(x,y) = \mathbb{E}_{(w,u) \sim \rho_0} [\sigma(w^T x + u^T e(t)) \sigma(w^T y + u^T e(t))]$ and denote the induced reproducing kernel Hilbert space (RKHS) as $\mathcal{H}_{K_{\rho_0}}$; if there is no misunderstanding, we denote $\mathcal{H} := \mathcal{H}_{K_{\rho_0}}$. It follows that $\bar{s}_{t,\bar{\theta}} \in \mathcal{H}$ if and only if $\|\bar{s}_{t,\bar{\theta}}\|_{\mathcal{H}} = \mathbb{E}_{(w,u) \sim \rho_0} [\|a(w,u)\|_2^2] < \infty$.

In Hsu et al. (2021), a notion of approximation quality called minimum width of the neural network is defined to measure the minimum number of random features needed to guarantee an accurate enough approximation with high probability. The exact definition is given below.

Definition 1. Given $\epsilon, \delta > 0$ and a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ with bounded norm $\|f\|_{\alpha} < \infty$, where α is the measure in \mathbb{R}^d associated with the corresponding function space. We also denote $g^{(i)}(x) = \sigma(w^T x + u^T e(t))$. The minimum width $m_{f,\epsilon,\delta,\alpha,\rho_0}$ is defined to be the smallest $r \in \mathbb{Z}^+$ such that with probability at least $1 - \delta$ over $g^{(1)}, \dots, g^{(r)}$,

$$\inf_{g \in \text{span}(g^{(1)}, \dots, g^{(r)})} \|f - g\|_{\alpha} < \epsilon.$$

Moreover, for $s \geq 0$, $p \in [1, \infty]$, and $U \subset \mathbb{R}^d$ be an open and bounded set, $W^{s,p}(U)$ is the Sobolev space with order s, p consists of all locally integrable function f such that for each multiindex α with $|\alpha| \leq s$, weak derivative of f exists and has finite L^p norm (see Evans (2010)). If $p = 2$, we denote $W^{s,2}(U) = H^s(U)$ to reflect the fact that it is a Hilbert space now. Finally, recall that the space of all Lipschitz functions on U is the same as $W^{1,\infty}(U)$.

With these settings and definitions, we can state and prove the following generalization error for the basic diffusion model using random feature model.

Theorem 5. Suppose that the target distribution μ is continuously differentiable and has a compact support, we choose an appropriate random feature ρ_0 , and there exists a RKHS \mathcal{H} such that $\bar{s}_{0,\bar{\theta}^*} \in \mathcal{H}$. Assume that the initial loss, trainable parameters, the embedding function $e(t)$ and the weighting function $\gamma(t)$ are all bounded.

We further suppose that for all $t \in [0, T]$, the score function $\nabla \log p_t \in H^s(K) \cap W^{1,\infty}(K)$ and there exists $\gamma > 0$ such that $\|\nabla \log p_t\|_{H^s(K)} \leq \gamma$, where $K \subset \mathbb{R}^d$ is compact. Then for fixed $0 < \epsilon, \delta \ll 1$, with probability at least $1 - \delta$, we have

$$\begin{aligned} D_{KL}(\mu \parallel \hat{\mu}) &\lesssim \left(\frac{\tau^4}{m^3 n} + \frac{\tau^2}{mn} + \frac{\tau^3}{m^2} + \frac{1}{\tau} + \frac{1}{m} \right) \\ &\quad + \min \left(\left(\frac{s}{\log m} \right)^{s/2}, \left(\frac{d(m^{1/d} - 2)}{s\gamma^{2/s}} \right)^{-s/2} \right) + D_{KL}(p_T \parallel \pi), \end{aligned}$$

where τ is the training time (steps) in the gradient flow dynamics (see Li et al. (2023)), m is the number of random features, n is the sample size of the target distribution, π is the stationary Gaussian distribution, p_T is the distribution of the forward OU process at time T , μ is the target distribution, and $\hat{\mu}$ is the distribution of the generated samples.

Proof. The proof follows exactly the same as in the proof of Theorem 1 in Li et al. (2023). The only extra work is to compute the universal approximation error of the random feature model for Sobolev functions on a compact domain. From compacted supported assumption (Lemma 1 in Li et al. (2023)), the forward process defines a random path $(X(t), t)_{t \in [0, T]}$ contained in a compact rectangular domain in \mathbb{R}^{d+1} .

Theorem 35 in Hsu et al. (2021) states the existence of a random feature ρ_0 such that for any $f \in H^s(K)$ with $\|f\|_{H^s(K)} \leq \gamma$, $m_{f, \epsilon, \delta, \alpha, \rho_0} \lesssim \frac{s^2 \gamma^{2+4/s} d^2}{\epsilon^{2+4/s}} \log\left(\frac{1}{\delta}\right) \exp\left(\min\left(d \log\left(\frac{\gamma^2}{\epsilon^2 d} + 2\right), \frac{\gamma^2}{\epsilon^2} \log\left(\frac{d\epsilon^2}{\gamma^2} + 2\right)\right)\right)$, which implies the approximation error term. \square

Remark 1. The random feature model has two difficulties in implementation.

If m , T , and τ are large enough, then the generalization error is small regardless to the sample size n . However, the choice of random feature ρ_0 is hard in practice, especially in neither Hsu et al. (2021) nor Li et al. (2023) the method to choose ρ_0 is specified. Therefore, the assumption that ρ_0 is appropriately chosen is very strong.

Even if ρ_0 is appropriately chosen, if we let $m \sim n$ and try to find an optimal early stopping time as in Li et al. (2023), the term $\min\left(\left(\frac{s}{\log n}\right)^{s/2}, \left(\frac{d(n^{1/d} - 2)}{s\gamma^{2/s}}\right)^{-s/2}\right)$ still dominates and shows the curse of dimensionality.

B PROOF OF RESULTS IN SECTION 3.1

Before the proofs, we note the strict convexity of the KL barycenter problems via a simple lemma.

Lemma 1. For any Polish space S , the KL barycenter problem $\min_{\mu \in \mathcal{P}(S)} \sum_{i=1}^k \lambda_i D_{KL}(\mu \parallel P_i)$ s.t. $\sum_{i=1}^k \lambda_i = 1$ is strictly convex.

Proof. Let $t \in (0, 1)$ and $\mu_1, \mu_2 \in S$ such that $\mu_1 \ll P_i$ and $\mu_2 \ll P_i$, for each $i = 1, 2, \dots, k$, then

$$\begin{aligned} \sum_{i=1}^k \lambda_i D_{KL}(t\mu_1 + (1-t)\mu_2 \parallel P_i) &< \sum_{i=1}^k \lambda_i [tD_{KL}(\mu_1 \parallel P_i) + (1-t)D_{KL}(\mu_2 \parallel P_i)] \\ &= t \sum_{i=1}^k \lambda_i D_{KL}(\mu_1 \parallel P_i) + (1-t) \sum_{i=1}^k \lambda_i D_{KL}(\mu_2 \parallel P_i), \end{aligned}$$

where the inequality follows from the strictly convexity of KL divergence in terms of μ with fixed P_i . Therefore, the KL barycenter problem is strictly convex. \square

B.1 Proof of Theorem 1

Proof. It suffices to consider a probability measure $\mu \in \mathcal{P}(\mathbb{R}^d)$ with absolutely continuous density $q(x)$ (otherwise the KL divergence is ∞) and show the existence. If there is no confusion, we use the density and measure

interchangeably. We denote $\mathcal{P}_{\text{ac}}(\mathbb{R}^d)$ as the space of all absolutely continuous distributions and define a functional $F : \mathcal{P}_{\text{ac}}(\mathbb{R}^d) \rightarrow \mathbb{R}$ that for $x \in \mathbb{R}^d$,

$$F(q, x) = \sum_{i=1}^k \lambda_i q(x) \log \left(\frac{q(x)}{p_i(x)} \right).$$

Therefore, the barycenter problem becomes

$$\min_{\mu \in \mathcal{P}_{\text{ac}}(\mathbb{R}^d)} \int_{x \in \mathbb{R}^d} F(q, x) dx \quad \text{s.t. } \sum_{i=1}^k \lambda_i = 1 \text{ and } \int_{x \in \mathbb{R}^d} q(x) dx = 1,$$

which is a variational problem with a subsidiary condition ((Gelfand and Fomin, 2000)). Therefore, from calculus of variations, a necessary condition for q to be an extremal of the variational problem is for some constant m

$$\frac{\partial}{\partial q} F(q) + m = 0.$$

Hence, the optimal solution is

$$q^*(x) = \frac{\prod_{i=1}^k p_i(x)^{\lambda_i}}{\int \prod_{i=1}^k p_i(x)^{\lambda_i} dx}.$$

□

B.2 Proof of Theorem 2

Before the proof of Theorem 2, we review a consequence of Girsanov's theorem (Theorem 8 in Chen et al. (2023)). We will use a similar technique as in Chen et al. (2023) to prove Theorem 2.

Theorem 6. Suppose $Q \in \mathcal{P}(C([0, T] : \mathbb{R}^d))$. For $t \in [0, T]$, let $\mathcal{L}(t) = \int_0^t b(s) dB(s)$ and the stochastic exponential $\mathcal{E}(\mathcal{L})(t) = \exp \left(\int_0^t b(s) dB(s) - \frac{1}{2} \int_0^t \|b(s)\|_2^2 ds \right)$, where B is a Q -Brownian motion. Assume $\mathbb{E}_Q \left[\int_0^T \|b(s)\|_2^2 ds \right] < \infty$. Then \mathcal{L} is a square integrable Q -martingale. Moreover, if $\mathbb{E}_Q [\mathcal{E}(\mathcal{L})(T)] = 1$, then $\mathcal{E}(\mathcal{L})$ is a true Q -martingale and the process $B(t) - \int_0^t b(s) ds$ is a P -Brownian motion, where P is a probability measure such that $P = \mathcal{E}(\mathcal{L})(T)Q$.

In most applications of Girsanov's theorem, we need to check a sufficient condition to hold, known as Novikov's condition. In the context of Theorem 6, Novikov's condition is

$$\mathbb{E}_Q \left[\exp \left(\frac{1}{2} \int_0^T \|b(s)\|_2^2 ds \right) \right] < \infty. \quad (14)$$

Now we begin the proof of Theorem 2.

Proof. From Lemma 1, it suffices to show the existence. Let $\alpha \in \mathcal{P}(C([0, T] : \mathbb{R}^d))$ with initial distribution α_0 . We denote $\alpha(0)$ as the initial distribution of the process whose law is measure α as notation. From the chain rule of KL divergence, we have

$$\begin{aligned} \sum_{i=1}^k \lambda_i D_{\text{KL}} (\alpha \| P_i) &= \sum_{i=1}^k \lambda_i D_{\text{KL}} (\alpha_0 \| \mu_i) \\ &\quad + \mathbb{E}_{z \sim \alpha_0} \left[\sum_{i=1}^k \lambda_i D_{\text{KL}} (\alpha(.|\alpha(0)=z) \| P_i(.|P_i(0)=z)) \right], \end{aligned}$$

where the first term solves the KL barycenter problem with respect to the initial distributions, and the second term solves the KL barycenter problem with all reference processes have the same initial distribution. Therefore, to finish the proof, we can assume for each $i = 1, \dots, k$, $\mu_i \sim \mu$, the same initial distribution.

Since we are finding the minimizer of the weight sum of KL divergences, it is sufficient to assume that α is the law of a diffusion process which is a strong solution of an SDE with the same diffusion (volatility) coefficient as all reference processes:

$$dX(t) = a(t, X(t)) dt + \sigma(t) dB(t), X(0) \sim \mu,$$

where B is a standard Brownian motion, and otherwise the KL divergence would be ∞ . For now, we assume that $a(t, x)$ is uniformly bounded.

When applying Girsanov's theorem, it is more convenient to view different path measures on $\mathcal{P}(C([0, T] : \mathbb{R}^d)$ as the different laws of the same single stochastic process. For notational convenience, we denote the single process as $\{Z(t)\}_{t \in [0, T]}$.

For each $i = 1, \dots, k$, we can apply the Girsanov's theorem to $Q = \alpha$ and

$$b(t) = \frac{1}{\sigma(t)} (a_i(t, Z(t)) - a(t, Z(t)))$$

in the setting of Theorem 6. Therefore, under the measure $P = \mathcal{E}(\mathcal{L})(T)\alpha$, there exists a Brownian motion $\{\beta(t)\}_{t \in [0, T]}$ such that

$$dB(t) = \frac{1}{\sigma(t)} (a_i(t, Z(t)) - a(t, Z(t))) dt + d\beta(t).$$

Since under the measure α , with probability 1,

$$dZ(t) = a(t, Z(t)) dt + \sigma(t) dB(t), Z(0) \sim \mu,$$

then this also holds P -almost surely, which implies that P -almost surely, $Z(0) \sim \mu$, and

$$\begin{aligned} dZ(t) &= a(t, Z(t)) dt + \sigma(t) \left(\frac{1}{\sigma(t)} [a_i(t, Z(t)) - a(t, Z(t))] dt + d\beta(t) \right) \\ &= a_i(t, Z(t)) dt + \sigma(t) d\beta(t). \end{aligned}$$

In other words, $P \sim P_i$ in law.

Therefore,

$$\begin{aligned} D_{\text{KL}}(\alpha \parallel P_i) &= \mathbb{E}_\alpha \left[\log \left(\frac{d\alpha}{dP_i} \right) \right] \\ &= \mathbb{E}_\alpha \left[\log \left(\frac{1}{\mathcal{E}(\mathcal{L})(T)} \right) \right] \\ &= \frac{1}{2} \mathbb{E}_\alpha \left[\int_0^T \frac{1}{\sigma(t)^2} \|a_i(t, Z(t)) - a(t, Z(t))\|_2^2 dt \right] \\ &\quad + \mathbb{E}_\alpha \left[\int_0^T \frac{1}{\sigma(t)^2} (a(t, Z(t)) - a_i(t, Z(t))) dt \right] \\ &= \frac{1}{2} \mathbb{E}_\alpha \left[\int_0^T \frac{1}{\sigma(t)^2} \|a_i(t, Z(t)) - a(t, Z(t))\|_2^2 dt \right] \end{aligned}$$

since Ito integral with regular integrand is a true martingale.

Therefore, the objective function of process level KL barycenter problem becomes

$$\frac{1}{2} \sum_{i=1}^k \lambda_i \mathbb{E}_\alpha \left[\int_0^T \frac{1}{\sigma(t)^2} \|a(t, Z(t)) - a_i(t, Z(t))\|_2^2 dt \right],$$

given we assume that all of reference laws have the same initial distribution. Therefore, as a functional optimization problem, the minimizer $a^*(t, x) = \sum_{i=1}^k \lambda_i a_i(t, x)$, which is indeed uniformly bounded and optimal, thus this finishes the proof. \square

C PROOF OF RESULTS IN SECTION 4

C.1 Preliminaries and Basic Tools

C.1.1 Preliminaries

We include this subsection to present basic definitions and notations used in our proofs.

Definition 2. S is a Polish space equipped with Borel σ -algebra $\mathcal{B}(S)$, $\{P_n\}_{n \in \mathbb{N}} \subset \mathcal{P}(S)$ is a set of probability measures, we say P_n converges to $P \in \mathcal{P}(S)$ weakly if and only if for each bounded and continuous function $f : S \rightarrow \mathbb{R}$, as $n \rightarrow \infty$,

$$\int_S f(x) dP_n(x) \rightarrow \int_S f(x) dP(x).$$

Definition 3. Given two measurable spaces (X, \mathcal{F}) and (Y, \mathcal{G}) , $f : X \rightarrow Y$ is a measurable function, and (X, \mathcal{F}, μ) is a (positive) measure space. The pushforward of μ is defined to be a measure $f_{\#}\mu$ such that for any $B \in \mathcal{G}$,

$$f_{\#}\mu(B) = \mu(f^{-1}(B)).$$

Definition 4. A differentiable function $F : \mathbb{R}^d \rightarrow \mathbb{R}$ is called L -smooth if for any $x, y \in \mathbb{R}^d$,

$$|F(x) - F(y) - F'(y)(x - y)| \leq \frac{L}{2} \|y - x\|_2^2.$$

Definition 5. A stochastic process $\{X_t\}_{t \in [0, T]}$ is called a local martingale if there exists a sequence of nondecreasing stopping times $\{T_n\}_{n \in \mathbb{N}}$ such that $T_n \rightarrow T$ and $\{X_{t \wedge T_n}\}_{t \in [0, T]}$ is a true martingale.

Next we define some notations and stochastic processes that will be used in the following proofs.

Recall the process (6) is a backward SDE with score terms replaced by the estimations. We say for each $i = 1, 2, \dots, k$, process \bar{X}_i is the theoretical backward process with exact score terms:

$$d\bar{X}_i(t) = (a\bar{X}_i(t) + \sigma^2 \nabla \log p_{T-t}^i(\bar{X}_i(t))) dt + \sigma dW_i(t), \bar{X}_i(0) \sim p_T^i. \quad (15)$$

The corresponding forward process is denoted as X_i :

$$dX_i(t) = -aX_i(t)dt + \sigma dW(t), X_i(0) \sim p_i \sim \mu_i. \quad (16)$$

We denote the marginal density of $X_i(t)$ as p_t^i ; when $t = 0$, we use the notation $p_i \sim \mu_i$. Process (8) is a time-discretized SDE to be implemented in practice. It can be viewed as an approximation of the theoretical barycenter process (denoted as \tilde{Y}) of the backward SDEs of the form (15):

$$d\tilde{Y}(t) = \left(a\tilde{Y}(t) + \sigma^2 \sum_{i=1}^k \lambda_i \nabla \log p_{T-t}^i(\tilde{Y}(t)) \right) dt + \sigma dW(t), \tilde{Y}(0) \sim \gamma_T^d, \quad (17)$$

where γ_T^d is the distribution level KL barycenter at time T with respect to the reference measures $\{p_T^1, \dots, p_T^k\}$. When T is large, γ_T^d is approximated by π in Equation (8). In theory, there is corresponding forward process with respect to process (17):

$$dY(t) = -aX(t)dt + \sigma dW(t), Y(0) \sim \tilde{Y}(T). \quad (18)$$

For a fixed λ , we denote $p_{\lambda, t}$ as the marginal distribution of process (18) at time t ; when $t = 0$, we ignore the time subscript.

C.1.2 Basic Algorithms

In this section, we recall the Frank-Wolfe method (Braun et al., 2022), which is used to solve an optimization problem with L -smooth convex function $f : \mathcal{X} \rightarrow \mathbb{R}$ on a compact domain \mathcal{X} :

$$\min_{x \in \mathcal{X}} f(x) \quad (19)$$

To measure the error of the algorithm, we define for each $\tau \geq 1$, the primary gap is

$$h_\tau = h(x_\tau) = f(x_\tau) - f(x^*),$$

where x^* is the minimizer of problem (19).

Algorithm 2 (vanilla) Frank-Wolfe with function-agnostic step size rule (Braun et al., 2022)

```

1: Input: Start atom  $x_0 \in \mathcal{X}$ , objective function  $f$ , smoothness  $L$ 
2: Output: Iterates  $x_1, \dots, x_\tau \in \mathcal{X}$ 
3: for  $\tau = 0$  to  $\dots$  do
4:    $v_\tau \leftarrow \arg \min_{v \in \mathcal{X}} \langle \nabla f(x_\tau), v \rangle$ 
5:    $\gamma_\tau \leftarrow \begin{cases} 1 & \text{if } \tau = 1 \\ \frac{2}{\tau+3} & \text{if } \tau > 1 \end{cases}$ 
6:    $x_{\tau+1} \leftarrow x_\tau + \gamma_\tau(v_\tau - x_\tau)$ 
7: end for

```

C.1.3 Basic Lemmas

In this subsection, we first list some basic lemmas (Lemma 2 to 5) that serve as essential tools in our proofs. All proofs can be found in (Chen et al., 2023).

Lemma 2. Suppose that Assumption 1 and 2 hold. For each $i = 1, 2, \dots, k$, let $Z_i(t)$ denote the forward auxiliary process (16), then for all $t \geq 0$,

$$\mathbb{E} \left[\|Z_i(t)\|_2^2 \right] \leq d \vee M \text{ and } \mathbb{E} \left[\|\nabla \log p_t^i(Z_i(t))\|_2^2 \right] \leq Ld.$$

Lemma 3. Suppose that Assumption 1 holds. For each $i = 1, 2, \dots, k$, let $Z_i(t)$ denote the forward auxiliary process (16). For $0 \leq s < t$, let $\delta = t - s$. If $\delta \leq 1$, then

$$\mathbb{E} \left[\|Z_i(t) - Z_i(s)\|_2^2 \right] \lesssim \delta^2 M + \delta d.$$

Lemma 4. Consider a sequence of functions $f_n : [0, T] \rightarrow \mathbb{R}^d$ and a function $f : [0, T] \rightarrow \mathbb{R}^d$ such that there exists a nondecreasing sequence $\{T_n\}_{n \in \mathbb{N}} \subset [0, T]$ such that $T_n \rightarrow T$ as $n \rightarrow \infty$ and for each $t \leq T_n$, $f_n(t) = f(t)$, then for each $\epsilon > 0$, $f_n \rightarrow f$ uniformly over $[0, T - \epsilon]$.

Lemma 5. $f : [0, T] \rightarrow \mathbb{R}^d$ is a continuous function, and $f_\epsilon : [0, T] \rightarrow \mathbb{R}^d$ such that for each $\epsilon > 0$, $f_\epsilon(t) = f(t \wedge (T - \epsilon))$, then as $\epsilon \rightarrow 0$, $f_\epsilon \rightarrow f$ uniformly over $[0, T]$.

Next, we review and give two results related to the fusion algorithms.

Lemma 6. For any fixed $\lambda \in \Delta_k$, $\tilde{Y}(T) \sim \mu_\lambda$, the KL barycenter of $\{\mu_1, \dots, \mu_k\}$.

Proof. In this proof, we use the following notations: suppose $x, y \in \mathbb{R}^d$ and $0 \leq s \leq t \leq T$, we denote $p^i(x, t|y, s)$ as the transition density of the i th auxiliary process from time s to t . Similarly, $p^\lambda(x, t|y, s)$ as the transition density of the barycenter process from time s to t .

Let λ be fixed, then at each time $t \in [0, T]$,

$$\nabla \log(p_{\lambda, t}(x)) = \nabla \sum_{i=1}^n \lambda_i \log(p_t^i(x)).$$

Expanding LHS and RHS at the same time, we get

$$\nabla \log \left(\int p^\lambda(x, t|y, 0) p_\lambda(y) dy \right) = \nabla \sum_{i=1}^k \lambda_i \log \left(\int p^i(x, t|y, 0) p_i(y) dy \right),$$

Note that as $t \rightarrow 0$, $p^i(x, t|y, 0) \rightarrow \delta(x - y)$ and $p^\lambda(x, t|y, 0) \rightarrow \delta(x - y)$, where the limit is the delta function.

Therefore, from the compactness assumption and dominated convergence theorem,

$$\begin{aligned}\nabla \log p_{\lambda}(x) &= \lim_{t \rightarrow 0} \nabla \log \left(\int p_{\lambda}(x, t|y, 0) p_{\lambda}(y) dy \right) \\ &= \lim_{t \rightarrow 0} \nabla \sum_{i=1}^k \lambda_i \log \left(\int p^i(x, t|y, 0) p_i(y) dy \right) \\ &= \nabla \sum_{i=1}^k \lambda_i \log p_i(x).\end{aligned}$$

Therefore,

$$\begin{aligned}\log p_{\lambda}(x) &\propto \sum_{i=1}^k \lambda_i \log p_i(x) \\ &= \log \left(\prod_{i=1}^k p_i(x)^{\lambda_i} \right) \\ &= \log \left(\prod_{i=1}^k p_i(x)^{\lambda_i} \right).\end{aligned}$$

Since $p_{\lambda}(x)$ is a density function, then after normalization

$$p_{\lambda}(x) = \frac{\prod_{i=1}^k p_i(x)^{\lambda_i}}{\int \prod_{i=1}^k p_i(x)^{\lambda_i} dx},$$

which is the solution of KL barycenter problem with reference measures p_1, \dots, p_k . \square

Next we give the proof of Proposition 1.

Proof. Recall that the objective function for λ is

$$F(\lambda) = \mathbb{E}_{\nu} \left[\log \nu(X) - \sum_{i=1}^k \lambda_i \log p_i(X) \right] + \log \left(\int \prod_{i=1}^k p_i(y)^{\lambda_i} dy \right). \quad (20)$$

We note that the first term is linear in λ , so to show convexity, it is enough to show the second term is convex in λ . If we denote $h_i(x) = \log(p_i(x))$ for each $i = 1, 2, \dots, k$ and X as the uniform distribution on K , then

$$\begin{aligned}\log \left(\int \prod_{i=1}^k p_i(y)^{\lambda_i} dy \right) &= \log \left(\int_{\mathbb{K}} \prod_{i=1}^k p_i(y)^{\lambda_i} dy \right) \\ &= \log \left(\frac{1}{|\mathbb{K}|} \int_{\mathbb{K}} \exp \left(\sum_{i=1}^k h_i(y) \lambda_i \right) dy \right) + \log(|\mathbb{K}|) \\ &= \log \left(\mathbb{E} \left[\exp \left(\lambda^T Z \right) \right] \right) + \log(|\mathbb{K}|),\end{aligned}$$

where $Z = (h_1(X), \dots, h_k(X))$ and $|\mathbb{K}|$ is the Lebesgue measure of \mathbb{K} . Since log of moment generating function is convex, then second term in Equation (20) is convex in λ . \square

Remark 2. In theory, the first order condition of the convex optimization problem (9) is

$$\begin{aligned}
 \frac{\partial F}{\partial \lambda_i}(\boldsymbol{\lambda}) &= - \int \nu(x) h_i(x) dx + \frac{\partial}{\partial \lambda_i} \log \left(\int \prod_{l=1}^k p_l(y)^{\lambda_l} dy \right) \\
 &= -\mathbb{E}_\nu [h_i(X)] + \frac{\int \prod_{l=1}^k p_l(y)^{\lambda_l} \log p_i(y) dy}{\int \prod_{l=1}^k p_l(y)^{\lambda_l} dy} \\
 &= -\mathbb{E}_\nu [h_i(X)] + \frac{\int \exp \left(\sum_{l=1}^k \lambda_l h_l(y) \right) h_i(y) dy}{\int \exp \left(\sum_{l=1}^k \lambda_l h_l(y) \right) dy}.
 \end{aligned}$$

In practice, each h_i is replaced by the estimated auxiliary densities, and the second term is computed independent of the target data ν . However, the implementation is extremely hard since the numerical integration of the second term may have large error and the error is hard to control.

C.2 Proof of Theorem 3

Before the proof of the sample complexity of the whole algorithm, we first prove a lemma about the auxiliary score estimation errors. The proof is adapted from Chen et al. (2023).

Lemma 7. Suppose that Assumption 2 holds, $\boldsymbol{\lambda}$ is fixed, and the step size $h = T/N$ satisfies $h \lesssim 1/L$, where $L \geq 1$. Let $p_{\boldsymbol{\lambda}}$ and $\hat{p}_{\boldsymbol{\lambda}}$ denote the distribution of process (17) and (8) at time T , respectively. Then we have

$$TV(p_{\boldsymbol{\lambda}}, \hat{p}_{\boldsymbol{\lambda}}) \lesssim \exp(-T) \max_{i=1,2,\dots,k} \sqrt{D_{KL}(p_T^i \| \pi)} + \sigma \sqrt{kT} (\epsilon_{score} + L\sqrt{dh} + Lh\sqrt{M}).$$

Remark 3. To interpret the result, suppose $\max_{i=1,2,\dots,k} \sqrt{D_{KL}(p_T^i \| \pi)} \lesssim \text{poly}(d)$ and $M \leq d$, then for fixed ϵ , if we choose $T \sim \log \left(\max_{i=1,2,\dots,k} \sqrt{D_{KL}(p_T^i \| \pi)} / \epsilon \right)$ and $h \sim \frac{\epsilon^2}{L^2 \sigma^2 k d}$, and hiding the logarithmic factors, then with $N \sim \frac{L^2 \sigma^2 k d}{\epsilon^2}$, $SE \lesssim \epsilon + \epsilon_{score}$. In particular, if we want to choose the sampling error $SE \lesssim \epsilon$, it suffices to have $\epsilon_{score} \lesssim \epsilon$.

Proof. We denote the laws of process (17) and (8) as α and $\beta \in C([0, T] : \mathbb{R}^d)$, respectively. For simplicity of the proof, we define a fictitious diffusion satisfying the SDE with $\hat{Y}(0) \sim \gamma_T^d$:

$$d\hat{Y}(t) = \left(a\hat{Y}(t) + \sigma^2 \sum_{i=1}^k \lambda_i s_{T-lh,\theta^*}^i(\hat{Y}(lh)) \right) dt + \sigma dW_i(t), t \in [lh, (l+1)h]. \quad (21)$$

since in practice, it is always convenient to use Gaussian π as a prior. We denote law of process (21) as $\beta_T \in C([0, T] : \mathbb{R}^d)$.

We also denote the score estimators of the process (18) as $s_{lh,\theta^*}^{\boldsymbol{\lambda}}$. Similarly as before, we consider only one stochastic process $Z(t)_{t \in [0, T]}$ now to use Girsanov's theorem.

For $t \in [lh, (l+1)h]$, we have the discretization error \mathcal{L} with

$$\begin{aligned}
 \mathcal{L} &= \sigma^2 \mathbb{E}_\alpha \left[\|s_{T-lh,\theta^*}^\lambda(Z(lh)) - \nabla \log p_{\lambda,T-t}(Z(t))\|_2^2 \right] \\
 &= \sigma^2 \mathbb{E}_\alpha \left[\left\| \sum_{i=1}^k \lambda_i [s_{T-lh,\theta^*}^i(Z(lh)) - \nabla \log p_{T-t}^i(Z(t))] \right\|_2^2 \right] \\
 &\lesssim \sigma^2 \sum_{i=1}^k \lambda_i^2 \mathbb{E}_\alpha \left[\|s_{T-lh,\theta^*}^i(Z(lh)) - \nabla \log p_{T-t}^i(Z(t))\|_2^2 \right] \\
 &\lesssim \sigma^2 \sum_{i=1}^k \lambda_i^2 \mathbb{E}_\alpha \left[\|s_{T-lh,\theta^*}^i(Z(lh)) - \nabla \log p_{T-lh}^i(Z(lh))\|_2^2 \right] \\
 &\quad + \sigma^2 \sum_{i=1}^k \lambda_i^2 \mathbb{E}_\alpha \left[\|\nabla \log p_{T-lh}^i(Z(lh)) - \nabla \log p_{T-t}^i(Z(lh))\|_2^2 \right] \\
 &\quad + \sigma^2 \sum_{i=1}^k \lambda_i^2 \mathbb{E}_\alpha \left[\|\nabla \log p_{T-t}^i(Z(lh)) - \nabla \log p_{T-t}^i(Z(t))\|_2^2 \right] \\
 &\lesssim k\sigma^2 \left(\epsilon_{\text{score}}^2 + \mathbb{E}_\alpha \left[\left\| \nabla \log \left(\frac{p_{T-lh}^i}{p_{T-t}^i} \right) (Z(lh)) \right\|_2^2 \right] + L^2 \mathbb{E}_\alpha [\|Z(lh) - Z(t)\|_2^2] \right).
 \end{aligned}$$

From Lemma 16 in Chen et al. (2023), we have the bound for the second term since $L \geq 1$,

$$\begin{aligned}
 \mathbb{E}_\alpha \left[\left\| \nabla \log \left(\frac{p_{T-lh}^i}{p_{T-t}^i} \right) (Z(lh)) \right\|_2^2 \right] &\lesssim L^2 dh + L^2 h^2 \mathbb{E}_\alpha [\|Z(lh)\|_2^2] \\
 &\quad + (1 + L^2) h^2 \mathbb{E}_\alpha [\|\nabla \log p_{T-t}^i Z(lh)\|_2^2] \\
 &\lesssim L^2 dh + L^2 h^2 \mathbb{E}_\alpha [\|Z(lh)\|_2^2] \\
 &\quad + L^2 h^2 \mathbb{E}_\alpha [\|\nabla \log p_{T-t}^i Z(lh)\|_2^2].
 \end{aligned}$$

Moreover, from L -Lipschitz condition,

$$\begin{aligned}
 \|\nabla \log p_{T-t}^i Z(lh)\|_2^2 &\lesssim \|\nabla \log p_{T-t}^i Z(t)\|_2^2 + \|\nabla \log p_{T-t}^i Z(lh) - \nabla \log p_{T-t}^i Z(t)\|_2^2 \\
 &\lesssim \|\nabla \log p_{T-t}^i Z(t)\|_2^2 + L^2 \|Z(lh) - Z(t)\|_2^2
 \end{aligned}$$

Hence,

$$\begin{aligned}
 \mathcal{L} &= \sigma^2 \mathbb{E}_\alpha \left[\|s_{T-lh,\theta^*}^\lambda(Z(lh)) - \nabla \log p_{\lambda,T-t}(Z(t))\|_2^2 \right] \\
 &\lesssim k\sigma^2 \epsilon_{\text{score}}^2 + k\sigma^2 L^2 dh + k\sigma^2 L^2 h^2 \mathbb{E}_\alpha [\|Z(lh)\|_2^2] \\
 &\quad + k\sigma^2 L^2 h^2 \mathbb{E}_\alpha [\|\nabla \log p_{T-t}^i Z(t)\|_2^2] + k\sigma^2 L^2 \mathbb{E}_\alpha [\|Z(lh) - Z(t)\|_2^2].
 \end{aligned}$$

From Lemma 2 and Lemma 3, we have

$$\begin{aligned}
 \mathcal{L} &= \sigma^2 \mathbb{E}_\alpha \left[\|s_{T-lh,\theta^*}^\lambda(Z(lh)) - \nabla \log p_{\lambda,T-t}(Z(t))\|_2^2 \right] \\
 &\lesssim k\sigma^2 (\epsilon_{\text{score}}^2 + L^2 dh + L^2 h^2 (d + M) + L^3 dh^2 + L^2 (dh + Mh^2)) \\
 &\lesssim k\sigma^2 (\epsilon_{\text{score}}^2 + L^2 dh + L^2 h^2 M).
 \end{aligned}$$

Therefore,

$$\begin{aligned}
 \mathcal{L} &= \sigma^2 \sum_{l=0}^{N-1} \mathbb{E}_\alpha \left[\int_{lh}^{(l+1)h} \|s_{T-lh,\theta^*}^\lambda(Z(lh)) - \nabla \log p_{\lambda,T-t}(Z(t))\|_2^2 dt \right] \\
 &\lesssim \sigma^2 kT (\epsilon_{\text{score}}^2 + L^2 dh + L^2 h^2 M).
 \end{aligned}$$

Next, we claim that

$$D_{\text{KL}}(\alpha \parallel \beta_T) \lesssim k\sigma^2 T (\epsilon_{\text{score}}^2 + L^2 dh + L^2 h^2 M). \quad (22)$$

Then from triangle inequality, Pinsker's inequality, and data processing inequality,

$$\begin{aligned} \text{TV}(p_{\lambda}, \hat{p}_{\lambda}) &\leq \text{TV}(\alpha, \beta) \\ &\leq \text{TV}(\beta, \beta_T) + \text{TV}(\alpha, \beta_T) \\ &\leq \text{TV}(\pi, \gamma_T^d) + \text{TV}(\alpha, \beta_T) \\ &\lesssim \exp(-T) \max_{i=1,2,\dots,k} \sqrt{D_{\text{KL}}(p_T^i \parallel \pi)} + \sigma\sqrt{kT} (\epsilon_{\text{score}} + L\sqrt{dh} + Lh\sqrt{M}). \end{aligned}$$

Hence it suffices to prove Equation (22). We will use a localization argument and apply Girsanov's theorem. The notations are the same as in Theorem 6.

Let $t \in [0, T]$, $\mathcal{L}(t) = \int_0^t b(s) dB(s)$, where B is an α -Brownian motion and for $t \in [lh, (l+1)h]$,

$$b(t) = \sigma (s_{T-lh, \theta^*}^{\lambda} (Z(lh)) - \nabla \log p_{\lambda, T-t} (Z(t))).$$

Recall that

$$\mathbb{E}_{\alpha} \left[\int_0^T \|b(s)\|_2^2 ds \right] \lesssim kT\sigma^2 (\epsilon_{\text{score}}^2 + L^2 dh + L^2 h^2 M).$$

Since $\{\mathcal{E}(\mathcal{L})(t)\}_{t \in [0, T]}$ is a local martingale, then there exists a non-decreasing sequence of stopping times $T_n \rightarrow T$ such that $\{\mathcal{E}(\mathcal{L})(t \wedge T_n)\}_{t \in [0, T]}$ is a true martingale. Note that $\mathcal{E}(\mathcal{L})(t \wedge T_n) = \mathcal{E}(\mathcal{L}^n)(t)$, where $\mathcal{L}^n(t) = \mathcal{L}(t \wedge T_n)$, therefore

$$\mathbb{E}_{\alpha} [\mathcal{E}(\mathcal{L}^n)(T)] = \mathbb{E}_{\alpha} [\mathcal{E}(\mathcal{L}^n)(0)] = 1.$$

Applying Theorem 6 to $\mathcal{L}^n(t) = \int_0^t b(s) \mathbf{1}_{[0, T_n]}(s) dB(s)$, we have that under the measure $P^n = \mathcal{E}(\mathcal{L}^n)(T)\alpha$, there exists a Brownian motion β^n such that for all $t \in [0, T]$,

$$dB(t) = \sigma (s_{T-lh, \theta^*}^{\lambda} (Z(lh)) - \nabla \log p_{\lambda, T-t} (Z(t))) \mathbf{1}_{[0, T_n]}(t) dt + d\beta^n(t).$$

Since under α we have almost surely

$$dZ(t) = (aZ(t) + \sigma^2 \nabla \log p_{\lambda, T-t} (Z(t))) dt + \sigma dB(t), Z(0) \sim \gamma^d,$$

which also holds P^n -almost surely since $P^n \ll \alpha$. Therefore, P^n -almost surely, $Z(0) \sim \gamma^d$ and

$$\begin{aligned} dZ(t) &= [aZ(t) + \sigma^2 s_{T-lh, \theta^*}^{\lambda} (Z(lh))] \mathbf{1}_{[0, T_n]} dt \\ &\quad + [aZ(t) \nabla \log p_{\lambda, T-t} (Z(t))] \mathbf{1}_{[T_n, T]} dt + \sigma d\beta(t). \end{aligned}$$

In other words, P^n is the law of the solution of the above SDE. Plugging in the Radon-Nikodym derivatives, we get

$$\begin{aligned} D_{\text{KL}}(\alpha \parallel P^n) &= \mathbb{E}_{\alpha} \left[\log \left(\frac{d\alpha}{dP^n} \right) \right] \\ &= \mathbb{E}_{\alpha} \left[\log \left(\frac{1}{\mathcal{E}(\mathcal{L})(T_n)} \right) \right] \\ &= \mathbb{E}_{\alpha} \left[-\mathcal{L}(T_n) + \frac{1}{2} \int_0^{T_n} \|b(s)\|_2^2 ds \right] \\ &= \mathbb{E}_{\alpha} \left[\frac{1}{2} \int_0^{T_n} \|b(s)\|_2^2 ds \right] \\ &\leq \mathbb{E}_{\alpha} \left[\frac{1}{2} \int_0^T \|b(s)\|_2^2 ds \right] \\ &\lesssim kT\sigma^2 (\epsilon_{\text{score}}^2 + L^2 dh + L^2 h^2 M) \end{aligned}$$

since $\mathcal{L}(T_n)$ is a martingale and T_n is a bounded stopping time (apply optional sampling theorem).

Now consider a coupling of $(P^n)_{n \in \mathbb{N}}$, β_T : a sequence of stochastic processes $(Z^n)_{n \in \mathbb{N}}$ over the same probability space, a stochastic process Z and a single Brownian motion W over that space such that $Z(0) = Z^n(0)$ almost surely, $Z(0) \sim \gamma^d$,

$$\begin{aligned} dZ^n(t) &= [aZ^n(t) + \sigma^2 s_{T-lh,\theta^*}^\lambda(Z^n(lh))] \mathbf{1}_{[0,T_n]} dt \\ &\quad + [aZ^n(t) + \nabla \log p_{\lambda,T-t}(Z^n(t))] \mathbf{1}_{[T_n,T]} dt + \sigma dW(t), \end{aligned}$$

and

$$dZ(t) = [aZ(t) + \sigma^2 s_{T-lh,\theta^*}^\lambda(Z^n(lh))] dt + \sigma dW(t).$$

Hence law of Z^n is P^n and law of Z is β_T . The existence of such coupling is shown in Chen et al. (2023).

Fix $\epsilon > 0$, define the map $\pi_\epsilon : C([0, T] : \mathbb{R}^d) \rightarrow C([0, T] : \mathbb{R}^d)$ such that

$$\pi_\epsilon(\omega)(t) = \omega(t \wedge T - \epsilon).$$

Since for each $t \in [0, T_n]$, $Z^n(t) = Z(t)$, then from Lemma 4, we have $\pi_\epsilon(Z^n) \rightarrow \pi_\epsilon(Z)$ almost surely uniformly over $[0, T]$, which implies that $\pi_\epsilon \# P^n \rightarrow \pi_\epsilon \# \beta_T$ weakly.

Since KL divergence is lower semicontinuous, then from data processing inequality, we have

$$\begin{aligned} D_{\text{KL}}(\pi_\epsilon \# \alpha \| \pi_\epsilon \# \beta_T) &\leq \liminf_{n \rightarrow \infty} D_{\text{KL}}(\pi_\epsilon \# \alpha \| \pi_\epsilon \# P^n) \\ &\leq D_{\text{KL}}(\alpha \| P^n) \\ &\lesssim kT\sigma^2(\epsilon_{\text{score}}^2 + L^2dh + L^2h^2M). \end{aligned}$$

From Lemma 5, as $\epsilon \rightarrow 0$, $\pi_\epsilon(\omega) \rightarrow \omega$ uniformly over $[0, T]$. Hence, from Corollary 9.4.6 in Ambrosio et al. (2005), as $\epsilon \rightarrow 0$, $D_{\text{KL}}(\pi_\epsilon \# \alpha \| \pi_\epsilon \# \beta_T) \rightarrow D_{\text{KL}}(\alpha \| \beta_T)$. Therefore, from Pinsker's inequality,

$$D_{\text{KL}}(\alpha \| \beta_T) \lesssim kT\sigma^2(\epsilon_{\text{score}}^2 + L^2dh + L^2h^2M).$$

□

Before the proof, we introduce some notations that will only be used for the proof of Theorem 3. Recall that the vanilla fusion method requires two layers of approximation before running the Frank-Wolfe method: we use target samples to estimate an expectation and we also estimate the densities of auxiliaries. As a notation, we denote $\hat{p}_{\hat{\lambda}}$ as the distribution of the generated sample by vanilla fusion, which is $\hat{\nu}_D$ in Section 4. $\hat{\lambda}$ is the weight computed with n target samples, $p_{\hat{\lambda}}$ denotes the barycenter of $\{\mu_1, \dots, \mu_k\}$ with the weight $\hat{\lambda}$, and $\bar{p}_{\hat{\lambda}}$ denotes the barycenter of $\{\bar{p}_1, \dots, \bar{p}_k\}$ with the weight $\hat{\lambda}$, where $\{\bar{p}_1, \dots, \bar{p}_k\}$ is the collection of estimates of auxiliary densities. Note that $\bar{p}_{\hat{\lambda}} \sim \hat{\mu}_{\hat{\lambda}}$ in Section 4.

Proof. From triangle inequality, we have

$$\begin{aligned} \text{TV}(\nu, \hat{p}_{\hat{\lambda}}) &\leq \text{TV}(\nu, p_{\hat{\lambda}}) + \text{TV}(p_{\hat{\lambda}}, \bar{p}_{\hat{\lambda}}) + \text{TV}(\bar{p}_{\hat{\lambda}}, \hat{p}_{\hat{\lambda}}) \\ &:= I_1 + I_2 + I_3, \end{aligned}$$

where I_1 represents the error when computing using the Frank-Wolfe method, $I_2 \leq \epsilon_2$ by assumption, and I_3 is the error from auxiliary score estimations, which is bounded by Lemma 7.

Therefore it only remains to bound I_1 . From Pinsker's inequality,

$$I_1 = \text{TV}(\nu, p_{\hat{\lambda}}) \lesssim \sqrt{D_{\text{KL}}(\nu \| p_{\hat{\lambda}})},$$

hence it is enough to bound $D_{\text{KL}}(\nu \| p_{\hat{\lambda}})$. From the compactedness assumption, we note that the objective function F of problem (9) is \tilde{L} -smooth for some constant \tilde{L} . Since the simplex in real space is convex, we denote the diameter of constrain set as D .

Recall that in Remark 2 in Section C.1.3, the target gradient ∇F is given by

$$F_i(\boldsymbol{\lambda}) = -\mathbb{E}_\nu[h_i(X)] + \frac{\int \exp\left(\sum_{l=1}^k \lambda_l h_l(y)\right) h_i(y) dy}{\int \exp\left(\sum_{l=1}^k \lambda_l h_l(y)\right) dy}.$$

Thus we define \hat{F} as the estimator of F with the gradient estimated by

$$\nabla \hat{F}_i(\boldsymbol{\lambda}) = \frac{-1}{n} \sum_{j=1}^n h_i(x_j) + \frac{\int \exp\left(\sum_{l=1}^k \lambda_l h_l(y)\right) h_i(y) dy}{\int \exp\left(\sum_{l=1}^k \lambda_l h_l(y)\right) dy},$$

where x_j are i.i.d. samples from the target distribution ν . Obviously \hat{F} is also convex and \tilde{L} -smooth on a compact set with diameter D .

We denote $\hat{\boldsymbol{\lambda}}(\tau)$ as the weight computed after τ iterations with n target samples, then from a standard Frank-Wolfe error analysis (e.g. Theorem 2.2 in Braun et al. (2022)), we have

$$\hat{F}(\hat{\boldsymbol{\lambda}}(\tau)) - \hat{F}(\boldsymbol{\lambda}^*) \leq \frac{2\tilde{L}D^2}{\tau+3}.$$

From Fundamental Theorem of Calculus, there exists a curve $\gamma : [0, 1] \rightarrow \Delta_k$ such that

$$\gamma(0) = \hat{\boldsymbol{\lambda}}(\tau), \gamma(1) = \boldsymbol{\lambda}^*,$$

$$F(\hat{\boldsymbol{\lambda}}(\tau)) - F(\boldsymbol{\lambda}^*) = \int_\gamma \nabla F(z) dz,$$

and

$$\hat{F}(\hat{\boldsymbol{\lambda}}(\tau)) - \hat{F}(\boldsymbol{\lambda}^*) = \int_\gamma \nabla \hat{F}(z) dz,$$

where the right hand sides are line integrals. From Hoeffding's inequality, for a fixed $z \in \Delta_k$, with probability at least $1 - \delta$,

$$\nabla F(z) \lesssim \nabla \hat{F}(z) + \mathcal{O}\left(\left(\log\left(\frac{1}{\delta}\right)\right)^{1/2} n^{-1/2}\right).$$

Therefore,

$$\begin{aligned} F(\hat{\boldsymbol{\lambda}}(\tau)) - F(\boldsymbol{\lambda}^*) &= \int_\gamma \nabla F(z) dz \\ &\lesssim \int_\gamma \nabla \hat{F}(z) dz + \mathcal{O}\left(\left(\log\left(\frac{1}{\delta}\right)\right)^{1/2} n^{-1/2}\right) \\ &= \hat{F}(\hat{\boldsymbol{\lambda}}(\tau)) - \hat{F}(\boldsymbol{\lambda}^*) + \mathcal{O}\left(\left(\log\left(\frac{1}{\delta}\right)\right)^{1/2} n^{-1/2}\right) \\ &\leq \frac{2\tilde{L}D^2}{\tau+3} + \mathcal{O}\left(\left(\log\left(\frac{1}{\delta}\right)\right)^{1/2} n^{-1/2}\right). \end{aligned}$$

If we let $\tau \rightarrow \infty$, then

$$\begin{aligned} D_{\text{KL}}\left(\nu \parallel p_{\hat{\boldsymbol{\lambda}}(\tau)}\right) &\leq D_{\text{KL}}(\nu \parallel p_{\boldsymbol{\lambda}^*}) + \mathcal{O}\left(\left(\log\left(\frac{1}{\delta}\right)\right)^{1/2} n^{-1/2}\right) \\ &\lesssim \epsilon_0^2 + \mathcal{O}\left(\left(\log\left(\frac{1}{\delta}\right)\right)^{1/2} n^{-1/2}\right). \end{aligned}$$

Therefore, from Pinsker's inequality, with probability at least $1 - \delta$,

$$\begin{aligned} \text{TV}(\nu, \hat{p}_{\hat{\Lambda}}) &\lesssim \epsilon_0 + \epsilon_2 + \exp(-T) \max_{i=1,2,\dots,k} \sqrt{D_{\text{KL}}(p_T^i \| \pi)} + \sigma \sqrt{kT} (\epsilon_{\text{score}} + L\sqrt{dh} + Lh\sqrt{M}) \\ &\quad + \mathcal{O}\left(\left(\log\left(\frac{1}{\delta}\right)\right)^{1/4} n^{-1/4}\right). \end{aligned}$$

□

C.3 Proof of Theorem 4

Before the proof, we define some notations that will be used in this proof. $\hat{p}_{\hat{\Lambda}}$ denotes the output distribution of Algorithm 1, which is $\hat{\nu}_P$ in Section 4. For a fixed small $\tilde{T} \ll 1$, in the calibration phase of ScoreFusion, we denote the forward process as Z (which is \tilde{X}^ν in Section 4): for $t \in [0, \tilde{T}]$,

$$dZ(t) = -aZ(t)dt + \sigma dW(t), Z(0) \sim \nu. \quad (23)$$

We learn an optimal weight by solving problem (10). We still denote the marginal distribution of process (23) at time t for fixed Λ as p_Λ^ν . Even though in practice we do not use the backward process of process (23), the following two versions of backward processes will help in the proof of Theorem 4: for $t \in [0, \tilde{T}]$ with $\tilde{Z}(0) \sim \gamma_{\tilde{T}}^d \sim \hat{Z}(0)$, and fixed Λ ,

$$d\tilde{Z}(t) = \left(a\tilde{Z}(t) + \sigma^2 \nabla \log p_{T-t}^\nu(\tilde{Z}(t))\right) dt + \sigma dW(t), \tilde{Z}(\tilde{T}) \sim \nu, \quad (24)$$

and for $l = 0, 1, \dots, N_{\tilde{T}}$,

$$d\hat{Z}(t) = \left(a\hat{Z}(t) + \sigma^2 \sum_{i=1}^k \Lambda_i s_{T-lh, \theta^*}^i(\hat{Z}(lh))\right) dt + \sigma dW(t), t \in [lh, (l+1)h], \quad (25)$$

where $hN_{\tilde{T}} = \tilde{T}$. Process (25) is the time-discretization version of process (24) without the initialization error (since $\tilde{Z}(0) \sim \hat{Z}(0)$). We denote the law of process (24) and (25) as $\alpha_{\tilde{T}}$ and $\beta_{\tilde{T}} \in \mathcal{P}(C([0, T] : \mathbb{R}^d))$, respectively. For fixed Λ , we call $\tilde{Z}(\tilde{T}) \sim p_{\hat{\Lambda}}^{\tilde{T}}$ (which is in fact ν) and $\hat{Z}(\tilde{T}) \sim \hat{p}_{\hat{\Lambda}}^{\tilde{T}}$.

Proof. From triangle inequality, we have

$$\begin{aligned} \text{TV}(\nu, \hat{p}_{\hat{\Lambda}}) &\leq \text{TV}\left(\nu, \hat{p}_{\hat{\Lambda}}^{\tilde{T}}\right) + \text{TV}\left(\hat{p}_{\hat{\Lambda}}^{\tilde{T}}, p_{\hat{\Lambda}}^{\tilde{T}}\right) + \text{TV}\left(p_{\hat{\Lambda}}^{\tilde{T}}, p_{\Lambda^*}^{\tilde{T}}\right) + \text{TV}\left(p_{\Lambda^*}^{\tilde{T}}, p_{\Lambda^*}\right) + \text{TV}\left(p_{\Lambda^*}, p_{\hat{\Lambda}}\right) + \text{TV}\left(p_{\hat{\Lambda}}, \hat{p}_{\hat{\Lambda}}\right) \\ &\lesssim \text{TV}\left(\nu, \hat{p}_{\hat{\Lambda}}^{\tilde{T}}\right) + \text{TV}\left(p_{\hat{\Lambda}}, \hat{p}_{\hat{\Lambda}}\right) + \text{TV}\left(\nu, p_{\Lambda^*}\right) + \text{TV}\left(p_{\hat{\Lambda}}, p_{\Lambda^*}\right) \\ &\lesssim \text{TV}\left(\nu, \hat{p}_{\hat{\Lambda}}^{\tilde{T}}\right) + \text{TV}\left(p_{\hat{\Lambda}}, \hat{p}_{\hat{\Lambda}}\right) + \epsilon_1 + \text{TV}\left(p_{\hat{\Lambda}}, p_{\Lambda^*}\right). \end{aligned}$$

From Lemma 7, we bound the second term

$$\text{TV}\left(p_{\hat{\Lambda}}, \hat{p}_{\hat{\Lambda}}\right) \lesssim \exp(-T) \max_{i=1,2,\dots,k} \sqrt{D_{\text{KL}}(p_T^i \| \pi)} + \sqrt{kT} \sigma (\epsilon_{\text{score}} + L\sqrt{dh} + Lh\sqrt{M}).$$

To bound the first term, we use a Girsanov's theorem and approximation argument similar as in Section C.2 to get

$$\begin{aligned} D_{\text{KL}}\left(\nu \| \hat{p}_{\hat{\Lambda}}^{\tilde{T}}\right) &\lesssim D_{\text{KL}}(\alpha_{\tilde{T}} \| \beta_{\tilde{T}}) \\ &\lesssim \frac{1}{\tilde{T}} \sum_{l=0}^{N_{\tilde{T}}-1} \mathbb{E}_{\alpha_{\tilde{T}}} \left[\int_{lh}^{(l+1)h} \sigma^2 \|s_{T-lh, \theta^*}^{\Lambda}(Z(lh)) - \nabla \log p_{T-t}^\nu(Z(t))\|_2^2 dt \right] \\ &\lesssim \frac{1}{\tilde{T}} \int_0^{\tilde{T}} \left[\sigma^2 \mathbb{E}_{Z(t) \sim p_t^\nu} \left[\left\| \sum_{i=1}^k (\Lambda_i s_{t, \theta^*}^i(Z(t))) - \nabla \log p_t^\nu(Z(t)) \right\|_2^2 \right] \right] dt \\ &\lesssim \tilde{\mathcal{L}}(\hat{\Lambda}; \theta^*, \sigma^2) = \tilde{\mathcal{L}}(\Lambda^*; \theta^*, \sigma^2) + [\tilde{\mathcal{L}}(\hat{\Lambda}; \theta^*, \sigma^2) - \tilde{\mathcal{L}}(\Lambda^*; \theta^*, \sigma^2)] \\ &:= I_1 + I_2, \end{aligned}$$

where I_1 represents the approximation error and I_2 represents the excess risk.

We note that from the bi-Lipschitz assumption and the compact support assumption,

$$D_{KL}(p_{\hat{\Lambda}} \| p_{\Lambda^*}) \sim \tilde{\mathcal{L}}(\hat{\Lambda}; \theta^*, \sigma^2),$$

hence we only need to bound I_1 and I_2 then.

From McDiarmid's inequality, for $\delta > 0$, with probability at least $1 - \delta$,

$$I_2 \lesssim \mathcal{O}\left(\sigma^2 \left(\log\left(\frac{1}{\delta}\right)\right)^{1/2} (N_{\tilde{T}} n)^{-1/2}\right) \lesssim \mathcal{O}\left(\sigma^2 \left(\log\left(\frac{1}{\delta}\right)\right)^{1/2} n^{-1/2}\right)$$

since $\tilde{T} \lesssim T$ and $N_{\tilde{T}}$ is small.

Finally, we need to give a bound on I_1 . The intuition is that from continuity of a diffusion process, when h is small, then p_h and p_0 are similar. Since the backward fused process is constructed as a process whose drift term is a linear combination of the auxiliary drifts, then the approximation error of the linear regression should be small, given Assumption 3.

Fix $t \in [0, \tilde{T}]$, then from the Lipschitz and the compactedness assumption, the loss \mathcal{L} is

$$\begin{aligned} \mathcal{L} &= \tilde{\mathcal{L}}(\Lambda^*; \theta^*, \sigma^2) \\ &= \frac{\sigma^2}{\tilde{T}} \int_0^{\tilde{T}} \mathbb{E}_{Z(t) \sim p_t^\nu} \left[\left\| \sum_{i=1}^k \Lambda_i^* s_{t, \theta^*}^i(Z(t)) - \nabla \log p_t^\nu(Z(t)) \right\|_2^2 \right] dt \\ &\lesssim \frac{\sigma^2}{\tilde{T}} \int_0^{\tilde{T}} \mathbb{E}_{Z(t) \sim p_t^\nu} \left[\left\| \sum_{i=1}^k \Lambda_i^* s_{t, \theta^*}^i(Z(t)) - \sum_{i=1}^k \Lambda_i^* \nabla \log p_t^i(Z(t)) \right\|_2^2 \right] dt \\ &\quad + \frac{\sigma^2}{\tilde{T}} \int_0^{\tilde{T}} \mathbb{E}_{Z(t) \sim p_t^\nu} \left[\left\| \sum_{i=1}^k \Lambda_i^* \nabla \log p_t^i(Z(t)) - \nabla \log p_t^\nu(Z(t)) \right\|_2^2 \right] dt \\ &\lesssim \sigma^2 k \epsilon_{\text{score}}^2 + \sigma^2 \mathbb{E}_{Z(0) \sim \nu} \left[\left\| \sum_{i=1}^k \Lambda_i^* \nabla \log p_0^i(Z(0)) - \nabla \log p_0^\nu(Z(0)) \right\|_2^2 \right] dt \\ &\quad + \frac{\sigma^2}{\tilde{T}} \int_0^{\tilde{T}} \mathbb{E}_{Z(t) \sim p_t^\nu} \left[\left\| \sum_{i=1}^k \Lambda_i^* \nabla \log p_t^i(Z(t)) - \sum_{i=1}^k \Lambda_i^* \nabla \log p_0^i(Z(t)) \right\|_2^2 \right] dt \\ &\quad + \frac{\sigma^2}{\tilde{T}} \int_0^{\tilde{T}} \mathbb{E}_{Z(t) \sim p_t^\nu} \left[\|\nabla \log p_t^\nu(Z(t)) - \nabla \log p_0^\nu(Z(t))\|_2^2 \right] dt \\ &\lesssim \sigma^2 k \epsilon_{\text{score}}^2 + \sigma^2 \mathbb{E}_{Z(0) \sim \nu} [\|p_0^\nu(Z(0)) - p_{\Lambda^*}(Z(0))\|_2^2] \\ &\quad + \sigma^2 \mathbb{E}_{Z(\tilde{T}) \sim \gamma_{\tilde{T}}^d} \left[\|p_{\tilde{T}}^\nu(Z(\tilde{T})) - p_0^\nu(Z(\tilde{T}))\|_2^2 \right] + \max_{j=1,2,\dots,k} \sigma^2 \mathbb{E}_{Z(\tilde{T}) \sim \gamma_{\tilde{T}}^d} \left[\|p_{\tilde{T}}^j(Z(\tilde{T})) - p_0^j(Z(\tilde{T}))\|_2^2 \right] \\ &\lesssim \sigma^2 k \epsilon_{\text{score}}^2 + \sigma^2 D_{KL}(\nu \| p_{\Lambda^*}^*) + \max_{j=1,2,\dots,k} \sigma^2 D_{KL}(p_{\tilde{T}}^j \| p_0^j) \\ &\lesssim \sigma^2 k \epsilon_{\text{score}}^2 + \sigma^2 \epsilon_1^2 + \sigma^2 k \mathcal{O}\left(\left(\tilde{T}\right)^{1/2}\right). \end{aligned}$$

Therefore, from Pinsker’s inequality, with probability at least $1 - \delta$,

$$\begin{aligned} \text{TV}(\nu, \hat{p}_{\hat{\Lambda}}) &\lesssim \text{TV}\left(\nu, \hat{p}_{\hat{\Lambda}}^{\tilde{T}}\right) + \text{TV}\left(p_{\hat{\Lambda}}, \hat{p}_{\hat{\Lambda}}\right) + \epsilon_1 + \text{TV}\left(p_{\hat{\Lambda}}, p_{\Lambda^*}\right) \\ &\lesssim \sqrt{D_{\text{KL}}\left(\nu \| \hat{p}_{\hat{\Lambda}}^{\tilde{T}}\right)} + \text{TV}\left(p_{\hat{\Lambda}}, \hat{p}_{\hat{\Lambda}}\right) + \epsilon_1 + \sqrt{D_{\text{KL}}\left(p_{\hat{\Lambda}} \| p_{\Lambda^*}\right)} \\ &\lesssim (\sigma + 1)\epsilon_1 + \sigma\sqrt{k}\mathcal{O}\left(\tilde{T}^{1/4}\right) + \mathcal{O}\left(\sigma\left(\log\left(\frac{1}{\delta}\right)\right)^{1/4} n^{-1/4}\right) \\ &\quad + \exp(-T) \max_{i=1,2,\dots,k} \sqrt{D_{\text{KL}}\left(p_T^i \| \pi\right)} + \sigma\sqrt{kT}\left(\epsilon_{\text{score}} + L\sqrt{dh} + Lh\sqrt{M}\right), \end{aligned}$$

which finishes the proof. \square

D RELATED WORK IN FINANCE

After the first draft of our paper, a concurrent work has been published by Jaimungal and Pesenti (2024), which discusses the KL barycenter in the process level. Jaimungal and Pesenti (2024) considers a constrained optimization problem in the finance setting of merging experts’ ideas, while our KL barycenter problem is unconstrained to fuse several auxiliary processes. The KL barycenter problem in Jaimungal and Pesenti (2024) imposes additional constrained conditions and coefficients of the SDEs. However, at this stage, our assumption is enough for the purpose of combining auxiliary models in the setting of generative models and is easy to integrate with neural network architecture. Our solution of the barycenter problem (Theorem 2) is equivalent to Proposition 2.4 in Jaimungal and Pesenti (2024) since the Radon-Nikodym derivative is 1 if our setting is plugged in. Moreover, the two papers discuss two different problems, use different approaches, and derive different results: in Jaimungal and Pesenti (2024), the constrained optimization is solved via the dynamic programming approach and it is not related to statistics or machine learning, while in our case, we utilize the optimality of in the sense of KL barycenter to design a new machine learning method (ScoreFusion) and derive the sample complexity bound.

E TERMINOLOGY CLARIFICATION

- When we use the phrase “diffusion model” in this paper, or in discussing how to “fuse” a number of them, the underlying object is the U-Net (Ronneberger et al., 2015) that parametrizes the score function of the time-reversed Ornstein–Uhlenbeck process, i.e., the $s_{t,\theta}(X)$ that parametrizes the $\nabla \log p_t(X)$ term in Equation 11. Although our notation is based on the stochastic differential equation (SDE) view of diffusion modeling, we note that denoising diffusion probabilistic modeling (DDPM; Ho et al. (2020)) can be formulated as a time-discretized version of SDE diffusion, as shown in Song et al. (2021b); the U-Net (i.e. $s_{t,\theta}(X)$) in SDE plays a similar role as the $\epsilon_\theta(X, t)$ function in DDPM.
- In this paper, we often used the term “MNIST” to refer to the digits dataset; but to be exact, what we meant is the extended version of the original MNIST dataset called EMNIST (Cohen et al., 2017). It is curated by the same institution (NIST), but containing a significantly larger set of digit samples. Since MNIST is the better-known name for the digits dataset, we referred to it as such in the text. In our code base, however, you would see us referencing the dataset as EMNIST, its real name.

F MNIST DIGITS EXPERIMENT

F.1 Model Architecture

Our parametrization of $s_{t,\theta}(X)$ employs a U-Net architecture composed of 8 convolutional layers (4 for encoding and 4 for decoding) with group normalization applied after each convolution, totaling $1M$ trainable parameters. Its code is modified from the **ScoreNet** class in the GitHub repository of Song et al. (2021b). The model also has skip connections between corresponding layers in the encoder and decoder. Temporal information is encoded via Gaussian random Fourier projections, followed by a dense layer to produce time embeddings, which are injected into each layer of the network through fully connected layers. This allows the model to condition its outputs on time steps effectively while maintaining multi-resolution feature extraction.

F.2 Implementation Details

Calibration of ScoreFusion (Part I of Algorithm 1 in the main text) can be understood as optimizing the normalized weights of an additional linear layer that sum over the $s_{t,\theta_i}(X)$ score tensors. The PyTorch workflow is implemented by our `FusionNet` class in `src/model_EMNIST.py`, provided in our Github repository. To ensure a fair comparison, the baselines and the auxiliary score models share the same U-Net architecture dimensions. The only difference between a Baseline 1 (B1) instance and an auxiliary instance is whether they are pre-trained or not. Each Baseline 2 (B2) instance is instantiated from a pre-trained auxiliary instance, and its parameters are fine-tuned on the target data until its test loss starts to increase.

We follow the standard machine learning convention of splitting each dataset into train, validation, and test sets with stratified sampling to ensure class balance. The ratio of training data to validation data is 4 : 1. We use the ground truth digit labels only for data-splitting, hiding them from the model during training. Model training taking more than an hour was run on two NVIDIA A40 GPUs in a computing cluster, while lightweight tasks were run on Google Colab using one A100 GPU.

Auxiliary and baseline U-Nets use the same trainer function `generic_train(*args)`, implemented in `src/model_EMNIST.py`. Auxiliary models are trained from scratch on 25000 MNIST images. Baseline 1 instances are also trained from scratch, with training data ranging from 2^6 up to 2^{14} . At training time, we supply an additional 25% data as the validation set. In short, auxiliary and baseline training share the training workflow, U-Net architecture, and score matching loss `loss_fn(*args)` as defined in `src/model_EMNIST.py`. Test evaluations are conducted on 5000 held-out images with a batch size of 200. The trainer contains the usual PyTorch pipeline of mini-batch + ADAM optimizer, plus three adaptations:

- ADAM learning rate (`lr`) is 1e-4 for all auxiliary score training. Baseline 1 training follows the `lr` schedule in format (`lr`, [sample size using this `lr`]): (1e-3, [64, 128]), (2e-4, [256, 512, 1024]), (1e-4, [4096, 16384]). We set a higher `lr` for smaller datasets to accelerate loss decrease; empirically, it goes a lot slower on smaller datasets. The batch size is $\min(128, TrainSize)$ for both training and validation loss.
- `ExponentialMovingAverage` (see `src/training.py`) is used for all checkpoint updating, the decay rate being a default 0.999.
- Early-stopping of both baseline and auxiliary training is determined by the same algorithm to ensure consistency. A detailed description is given in below.

Early Stopping To balance overfitting reduction and adequate learning, our experiments early-stop the model learning when validation loss exceeds the lowest realized validation loss by $\geq 50\%$ for more than 50 consecutive epochs, implemented by the `EarlyStopper` class in `src/training.py`. We examine the fine-tuning / training loss curves of Baseline 2 and 1 U-Nets on ≤ 1000 MNIST images and provide one such plot in Figure 6. These inspections suggest that there is no under-training of neither Baseline 1 nor Baseline 2.

Baseline 2 Setup The auxiliary model we fine-tuned on generates an empirical digits distribution of $\{‘7’: 72\%, ‘9’: 24\%, \text{others: } 4\%\}$ over 10,000 samples. Out of the four pre-trained auxiliaries, this one is chosen because it was trained from the (70%, 30%) frequency and therefore is the closest in frequency to the target distribution of $\{‘7’: 60\%, ‘9’: 40\%\}$; intuitively, it should be the easiest to finetune among the four. Using full parameter fine-tuning, we initialize the score weights with the chosen auxiliary, and update all weights every step. We use the ADAM optimizer with `lr = 2e-5`. `Max_epoch` is set at 200, and rest of the training/test hyperparameters remain the same as stated as above (e.g. early-stopping criterion, exponential moving average).

F.3 Additional Statistics & Samples

Tables 3 and 4 provide additional information about the outcome of the MNIST experiments. Figure 6 shows the typical finetuning / training loss dynamics for Baselines 1 and 2. Lastly, Figures 7 and 8 show i.i.d. digits samples generated by the ScoreFusion-trained model versus those of the two baseline methods.

Table 3: Full digits proportions of 1024 images sampled from each of the four unadapted auxiliary models, classified by a SpinalNet (Kabir et al., 2022). For simplicity, in the main text we have combined the proportions of digits other than 7 and 9 into one meta class named **Others**.

Auxiliary	0	1	2	3	4	5	6	7	8	9
1	0.1%	0.1%	0.6%	0.6%	1.1%	0.3%	0.0%	18.7%	0.2%	78.2%
2	0.1%	0.1%	0.3%	0.8%	1.1%	0.5%	0.0%	41.1%	0.2%	55.8%
3	0.0%	0.2%	0.7%	0.7%	1.2%	0.8%	0.0%	72.1%	0.6%	23.7%
4	0.1%	0.5%	0.7%	0.5%	0.9%	0.4%	0.1%	87.9%	0.3%	8.6%
Target Distribution								60%	40%	

Table 4: Optimal weights λ^* corresponding to the ScoreFusion models whose NLL test losses we reported in Table 1. Each column is a weight vector that parameterizes the ScoreFusion model trained with 2^j data.

λ_i	2^6	2^7	2^8	2^9	2^{10}	2^{12}	2^{14}
$i = 1$	0.199	0.187	0.182	0.181	0.167	0.183	0.176
$i = 2$	0.305	0.326	0.328	0.319	0.345	0.311	0.310
$i = 3$	0.279	0.267	0.284	0.285	0.319	0.294	0.295
$i = 4$	0.217	0.220	0.206	0.216	0.170	0.213	0.220

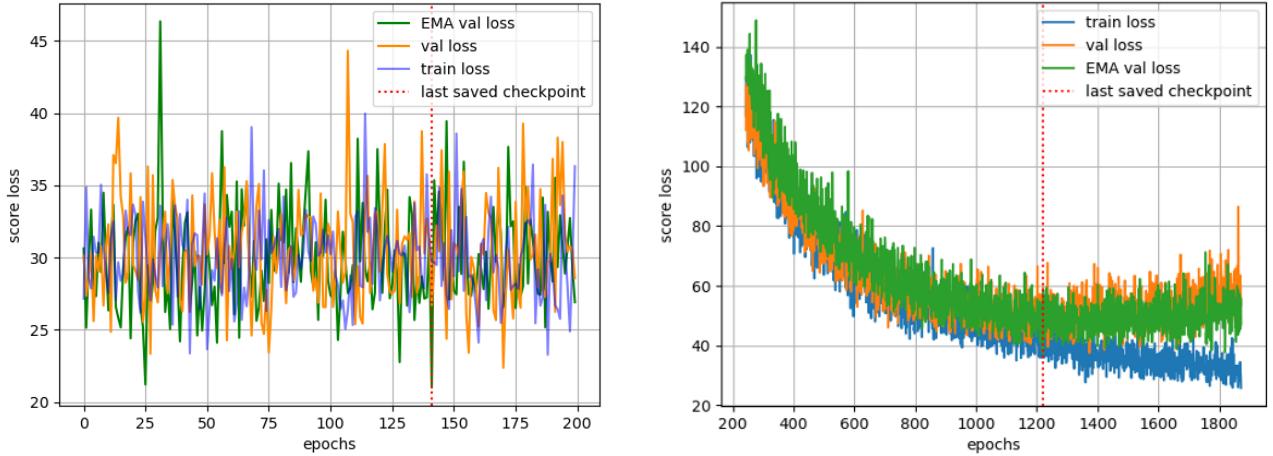


Figure 6: **Left:** Typical fine-tuning loss curves for Baseline 2. Train data size + Val data size = $64 + 16 = 80$. **Right:** Baseline 1 training loss curves. Train data size + Val data size = $512 + 128 = 640$. EMA = Exponential Moving Average. Dotted line is the epoch where EMA checkpoint had the lowest loss on the validation data.

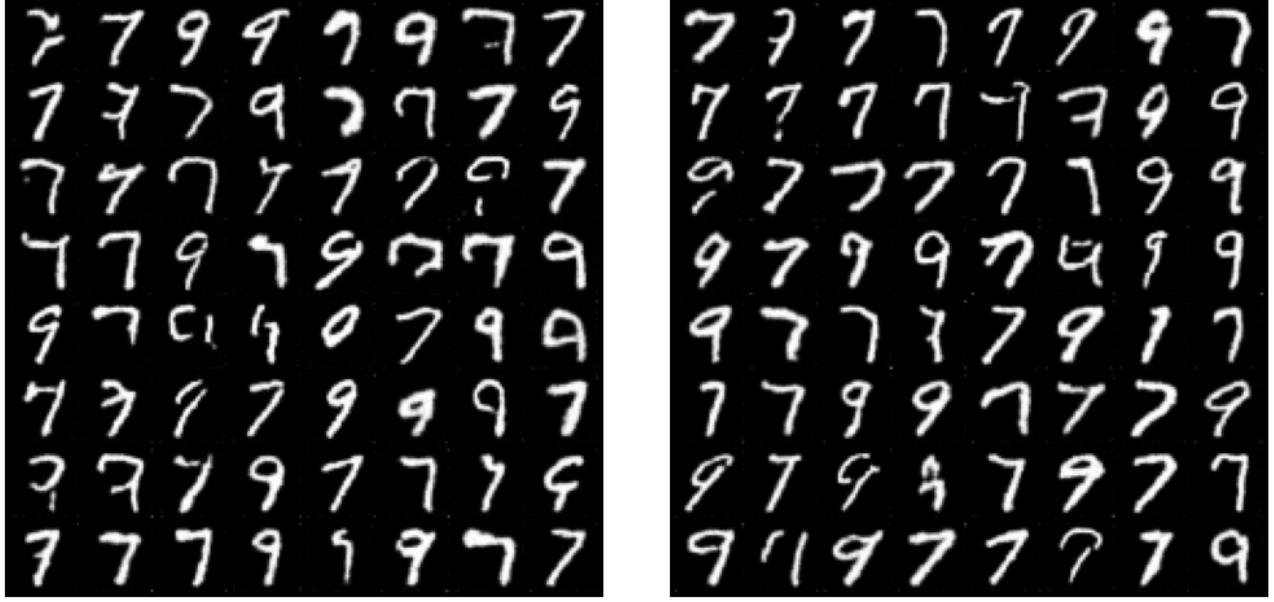


Figure 7: **Left:** uncurated samples generated by Baseline 2, obtained from directly fine-tuning a 70-30 auxiliary model. **Right:** uncurated samples generated by calibrating a ScoreFusion model. Both **Left** and **Right** models were fine-tuned/calibrated on the same 64 images from the target population.

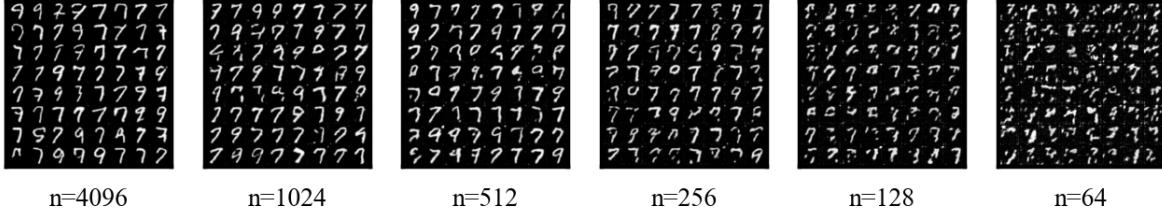


Figure 8: Samples generated by Baseline 1 under different quantities of training data.

G SDXL PROFESSIONAL PORTRAITS EXPERIMENT

G.1 Model Architecture

All generative models in our stable diffusion experiment are derived from the open-source SDXL 1.0 base (Stability-AI, 2024). As a type of latent diffusion model (Rombach et al., 2022), SDXL is composed of a variational autoencoder (VAE), a text encoder and a 2D conditioned U-Net. The U-Net consists of a combination of convolutional and cross-attention blocks. It operates across three resolution scales, with cross-attention integrated into both the downsampling and upsampling paths. The model conditions on text and time-step information via cross-attention, using positional embeddings for time encoding. According to Hugging Face’s documentation (HuggingFace, 2023a), fine-tuned models in the stable diffusion community are almost always produced with DreamBooth (Ruiz et al., 2022), which outputs a LoRA adapter (Low-Rank Adaptation; Hu et al. (2021)) of only the U-Net model, leaving the text encoder and the VAE unchanged. Intuitively, the LoRA adapter adds low-rank perturbations to the dense layers in a base U-Net so as to minimize the empirical score-matching loss on the fine-tuning dataset.

The two auxiliary diffusion models used in our experiment are also products of LoRA fine-tuning; their U-Nets were each fine-tuned from a gender- and race-homogeneous dataset by AlDahoul et al. (2024a,b), such that the unconditional generation (unspecified conditions being a person’s gender and race) of professional portraits creates

image samples that predominantly mirror the monolithic phenotype of the corresponding finetuning population. This design choice is intended to stylize the issue of distributional social bias in popular text-to-image (TTI) models, a problem widely recognized in the AI alignment literature (Cho et al., 2023; Luccioni et al., 2024). The sub-problem addressed by our SDXL experiment is: given TTI model checkpoints that are individually fine-tuned on a distinct subpopulation, sampling from the KL barycenter provides a theoretically grounded approach to organically blend heterogeneous features in the subpopulations into one generative model.

G.2 Implementation Details

Since Appendix F and Section 5.1 have thoroughly tested the sample efficiency of the training phase of ScoreFusion and reflected our theoretical development, we focus on exploring the inference-time / sampling benefit of our proposed KL barycenter approach in the SDXL experiment. Hence we assumed that the barycenter weights λ are fixed at $(0.5, 0.5)$ in this experiment. Our goal is to probe the ability of KL barycenter sampling to sample from low-probability regions (with respect to the auxiliary models’ sampling distributions) in the pixel space; we provide more theoretical motivation in section G.5.

To implement this sampling, we revised the source code of the *Diffusers* library that implements the denoising loop of SDXL 1.0, allowing us to ensemble score evaluations of the two auxiliary models at inference time.

G.2.1 Sampling from KL-Divergence Barycenter

Because the `StableDiffusionXLPipeline` class in Hugging Face’s `Diffusers` library does not natively support the ensembling of two U-Nets denoisers during inference, we adapted their source code so that it can ensemble the log-probability gradient tensors for a specified set of barycenter weights. The new pipeline class can be inspected in `src/SDXL_inference.py` in our Github repository. As mentioned in the previous subsection, the two auxiliary models share the exact same VAE and text encoder. Therefore, the only component of the generation pipeline that we combine at inference time is the U-Net’s outputs.

Our hyperparameter setting follows the default values of a general `Diffusers` inference pipeline; checkpoints are loaded at `fp16` precision, CFG guidance scale is 5.0, guidance rescale is zeroed, latent noise has shape $4 \times 128 \times 128$, and the final tensor output has shape $4 \times 1024 \times 1024$. The number of denoising steps is set to 100, and the timesteps are given by the default discrete Euler scheduler. We used the same text prompts across all generations: {‘Positive Prompt’: “*a photo of a mathematics scientist, looking at the camera, ultra quality, sharp focus*”, ‘Negative Prompt’: “*cartoon, anime, 3d, painting, b&w, low quality*.”}

G.2.2 Checkpoint Merging Model

For comparison, we also generated samples from a diffusion model that results from equally merging LoRA checkpoints of the two auxiliary models. Note that because the base SDXL model is the same for both models, merging the LoRA checkpoints and appending the merged checkpoint to the base model is the same as merging the entirety of the two auxiliary model checkpoints. As stated in the main text, checkpoint merging is a common model adaptation method in the online stable diffusion community. Figure 9 shows the checkpoint merging UI of the popular image generation tool `stable-diffusion-webui` (Automatic1111, 2022). The inference-time hyperparameter setting is the same as the one stated in Section G.2.1.

G.2.3 Individual Auxiliary Models

Sampling from an individual auxiliary model is the same as sampling from a regular SDXL model, also equivalent to setting $\lambda = (1, 0)$ or $(0, 1)$ for the KL barycenter approach or checkpoint merging. The inference-time hyperparameter setting is the same as the one stated in Section G.2.1.

G.2.4 CLIP Distance Computations

To quantify the semantic shifts and nuances we observe visually in the generated samples, we calculate CLIP distances (Radford et al., 2021) between each sampled image and four semantic concepts: [Female, Male, East Asian, White] using OpenAI’s `clip-vit-base-patch32` image encoder model. Per convention, the CLIP distance is calculated as $100 \times \text{cosine_similarity}(E_I, E_T)$, where E_I and E_T are respectively the image and text embedding vectors encoded by the CLIP model. Intuitively, it measures the semantic similarity between a sampled image

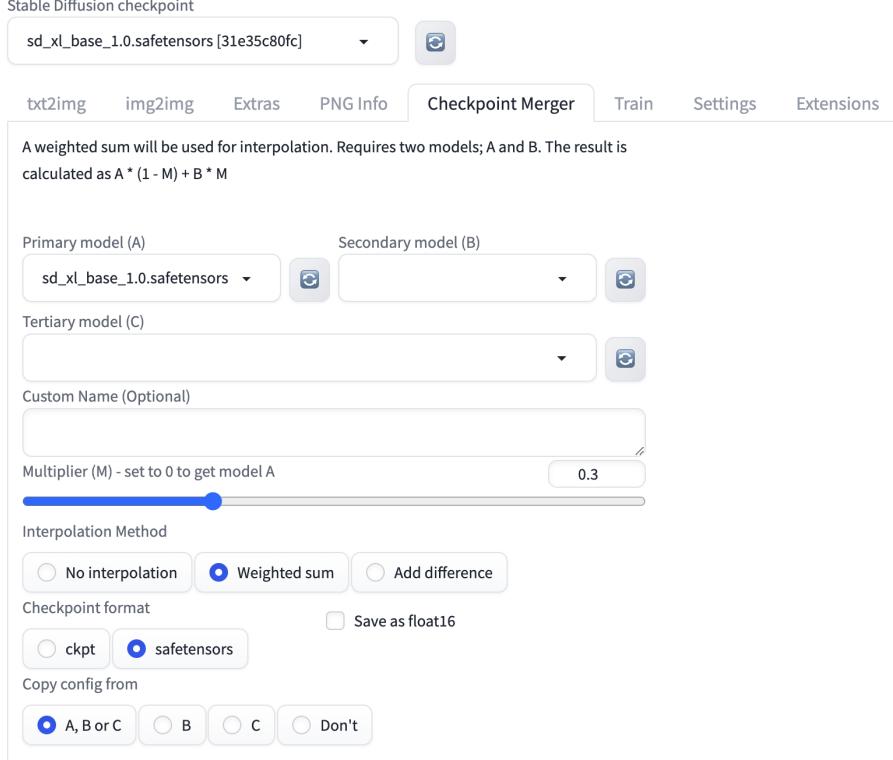


Figure 9: UI for checkpoint merging in `stable-diffusion-webui` (Automatic1111, 2022). Screenshot taken by us. The exposition text for computing the weighted sum was from the UI developer. The public repository has over 140k GitHub stars and 26.7k forks as of October 2024. To put its popularity in perspective, OpenAI’s API demo repository `openai-cookbook` has 59.2k stars and 9.4k forks.

and a gender / ethnic concept. This setup was used to produce Figures 4 and 5 shown earlier in the text. The kernel density estimation (KDE) contour plot is simply a smoothed version of the scatter plot, made using the `Seaborn` package for better readability. A couple additional comments are in place regarding Figure 4:

- We use a 2D joint plot instead of picking one gender, because gender expression can be non-binary and an individual representation may be rationalizable to multiple classes simultaneously; a “woman” can also exhibit visual traits that are considered “masculine” by social norms; the capacity for storing and representing ambiguity, we believe, should be a trait of a robust, information-theoretically efficient world simulator.
- One can see the two unimodal clusters (red & orange) formed by the empirical distribution of each biased, stereotypical model, confirming the qualitative judgement.
- The KL barycenter’s empirical distribution (blue) **interpolates** the valley region between the two modes, which was underexplored by the two biased models. By contrast, the baseline’s (checkpoint-merged model) empirical distribution appears to be more of a bimodal mixture of the two biased models’ empirical distributions, i.e., it is less effective than the barycenter at interpolating the underrepresented regions.

Figure 10 repeats the statistical analysis of Figure 4 for two ethnic concepts “East Asian” and “White”. Notice that the empirical distributions of the two biased models still form two distinct unimodal clusters, but the distance between their centroids is shorter than that of the gender clusters’. The relative ambiguity is expected, as there are many more identifiable ethnic categories a person can fall under besides East Asian or White. A sharper clustering would require the inclusion of more ethnic covariates to “lift” the scatter points to a higher-dimensional space. One can still observe the semantic dispersion of the KL barycenter in the valley region between the two biased modes, though the distinction with the checkpoint-merged model is less pronounced than that in Figure 4. Both of the adapted models exhibit a slight bias towards sampling the “White” hyperplane, which, as we hypothesized earlier, is likely an artifact of marginalization over other latent ethnic concepts.

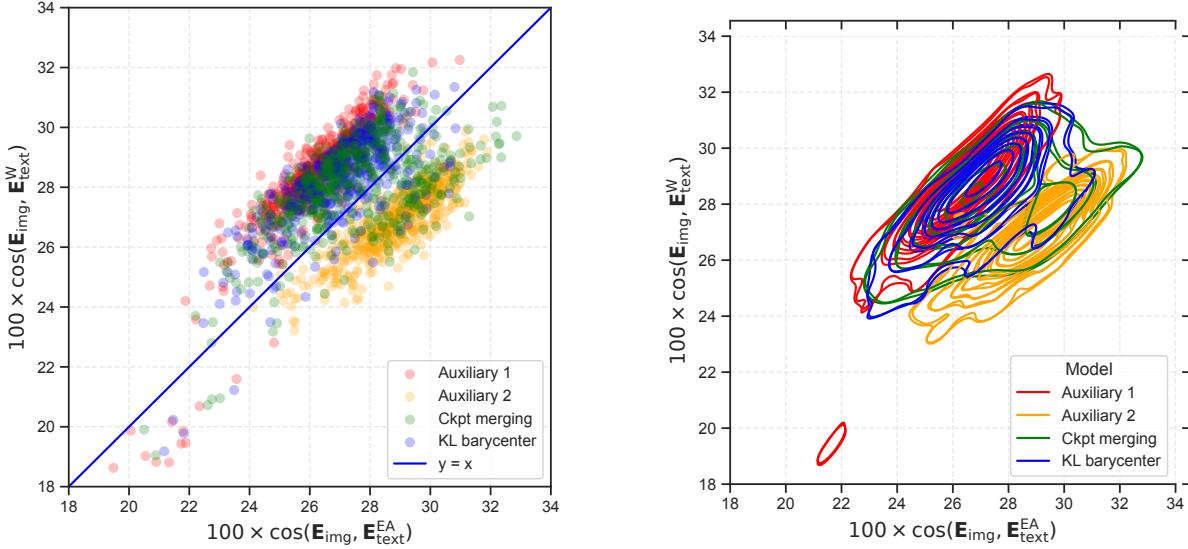


Figure 10: **Left:** Scatter plot of the empirical distribution (512 images) of each generative model, projected onto a 2D ethnicity semantic space. **Right:** KDE contour plot of the kernel density estimation (KDE) of the left scatter plot, with a bandwidth of 0.8.

\mathbf{E}_{img} are CLIP embeddings of image samples. $\mathbf{E}_{\text{text}}^{\text{EA}}, \mathbf{E}_{\text{text}}^W$ are (fixed) text embeddings of “*a photo of an East Asian scientist*” and “*a photo of a White scientist*”. The line $y = x$ is drawn to indicate the idea of “ethnic ambiguity”. The same PyTorch seed was used for all four pipelines.

G.3 Ablation Study

We provide additional quantitative evaluations on the visual effect of varying the KL barycenter weight vector $\lambda \in \Delta^1$. Figure 11 showcases shifts in image semantics (left subfigure is in the gender space, right subfigure is in the ethnicity space). Both subfigures show a **directional** drift of the semantic distribution as λ_2 increases, indicating the smoothness of the KL-divergence barycenter trajectory.

Another noteworthy observation is the asynchronous-ness of the two semantic transitions (gender & ethnicity): one might think that the transition would proceed at roughly the same rate, but instead, Figure 11 shows that between 0.2 (purple contours) to 0.4 (blue contours), the gender distribution drifts dramatically while the ethnicity distribution stays mostly unchanged. Figure 12 swaps out samples from $\lambda_2 = 0.8$ for the two original auxiliary models to allow for a complementary comparison.

In addition to Figure 3, we provide another visual comparison of KL barycenter and checkpoint merging on different λ values in Figure 13.

G.4 Additional Portrait Samples

For consistency, we use the same text prompt as the one stated in section G.2.1 for all image generations. Importantly, the text prompt does not specify the gender or race of the person. However, as Figure 14 shows, two grids of samples generated by the two auxiliary models display a homogeneous gender & ethnic expression. This setup is to contrast how our KL barycenter approach can effectively spotlight an underrepresented subpopulation in either model. Figures 15, and 16 provide additional samples generated by the model combination methods (checkpoint merging & KL barycenter) beyond those presented in the main text.

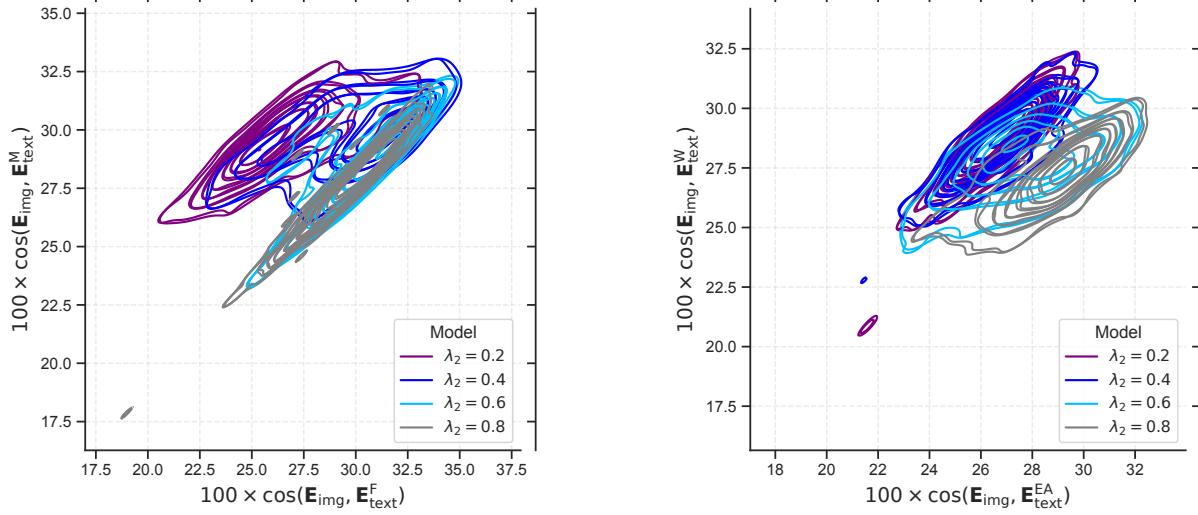


Figure 11: **Left:** KDE contour plot of the KL barycenter distribution under different weights λ_2 , projected onto a 2D gender semantic space. **Right:** Same analysis, but onto a 2D ethnicity semantic space.
 $\lambda_2 = 0$ and $\lambda_2 = 1$ (not shown) respectively correspond to “Auxiliary 1” and “Auxiliary 2” in Figure 12.

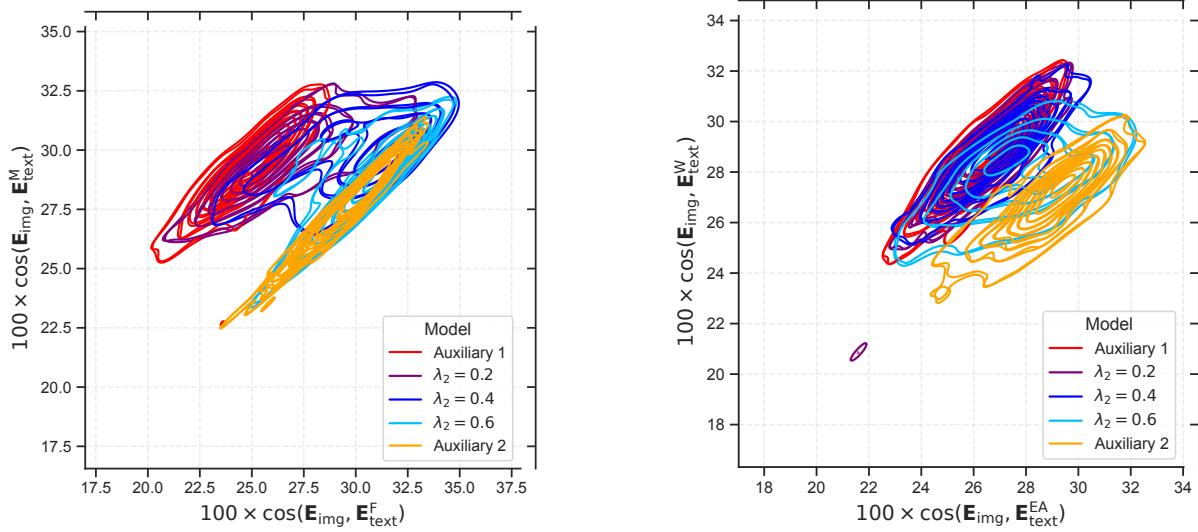


Figure 12: **Left:** KDE contour plot of the KL barycenter distribution under different weights λ_2 , projected onto a 2D gender semantic space. **Right:** Same analysis, but for ethnicity CLIP distances.
In this figure, we focus on comparing three non-trivial interpolation values with the original distributions.

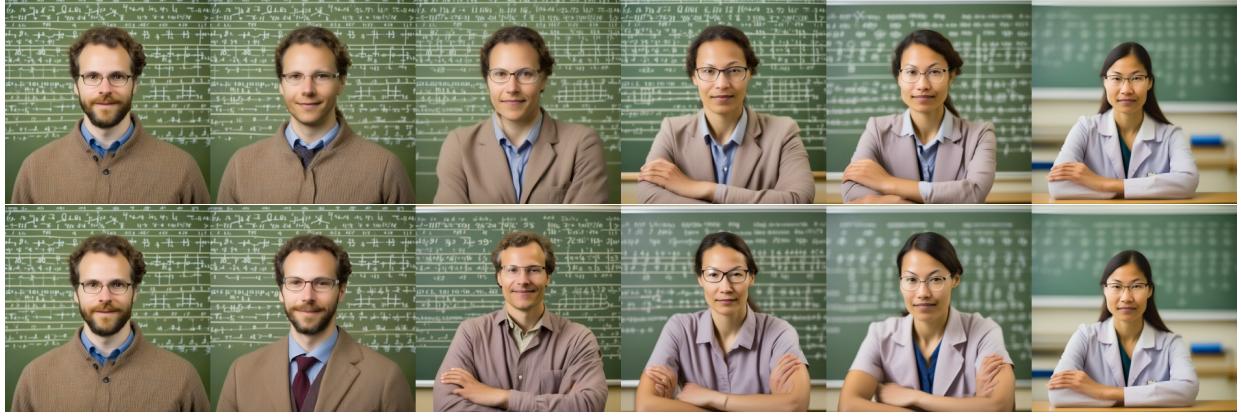


Figure 13: **Top row:** KL barycenter. **Bottom row:** Checkpoint merging.

The same Gaussian noise (but different from that of Figure 3) was used to seed all twelve images. From left to right, $\lambda_2 \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ and $\lambda_1 = 1 - \lambda_2$. $\lambda_2 = 0$ and $\lambda_2 = 1$ each reduce to an original (biased) auxiliary SDXL model. Observe that the bottom row samples show an abrupt identity shift from $\lambda_2 = 0.2$ to 0.4 and 0.4 to 0.6, whereas the top row shows a smoother transition from one identity concept to another.

G.5 Theory-Inspired Heuristic for the Phenomenon

As we saw in Figures 15 and 16, the KL barycenter of the two auxiliary models generate professional portraits that seem unlikely under the marginal distribution of either model, depicting an underrepresented and much more gender-neutral gender expression. We provide a heuristic, theory-based explanation for this phenomenon, using an example of combining two one-dimensional Gaussian models. Recall from Theorem 1 the density function of the distribution-level KL barycenter with barycenter weights vector λ :

$$p_{\lambda}(x) = \frac{\prod_{i=1}^k p_i(x)^{\lambda_i}}{\int_{\mathbb{R}^d} \prod_{i=1}^k p_i(x)^{\lambda_i} dx},$$

where p_1, \dots, p_k are the probability density functions of the marginal distributions of the k auxiliary models, and d is the dimension of the Euclidean space that x lives in.

Set $k = 2$, $d = 1$. Let p_1 and p_2 be probability density functions of Gaussian distributions with means μ_1, μ_2 and variance σ_1^2, σ_2^2 . Then the numerator becomes

$$\prod_{i=1}^2 p_i(x)^{\lambda_i} = \left(\prod_{i=1}^2 \left(\frac{1}{\sqrt{2\pi\sigma_i^2}} \right)^{\lambda_i} \right) \exp \left(- \sum_{i=1}^2 \lambda_i \frac{(x - \mu_i)^2}{2\sigma_i^2} \right).$$

Putting aside constant terms, we see that p_{λ} is also Gaussian. With some routine algebraic manipulations, it can be shown that the distribution has mean $\mu_{\lambda} = \frac{B}{A}$ and variance $\sigma_{\lambda}^2 = \frac{1}{2A}$, where

$$A = \lambda_1 \frac{1}{2\sigma_1^2} + \lambda_2 \frac{1}{2\sigma_2^2}, \quad B = \lambda_1 \frac{\mu_1}{2\sigma_1^2} + \lambda_2 \frac{\mu_2}{2\sigma_2^2}$$

and we impose the constraint that $\lambda_i \geq 0$ and $\lambda_1 + \lambda_2 = 1$. When $\lambda_i = 1$, $\mu_{\lambda} = \mu_i$ and $\sigma_{\lambda}^2 = \sigma_i^2$. Figure 17 shows the barycenter distribution under different λ_1 values.

One way to interpret this stylized setup is that the real line represents a spectrum of gender expression (or another semantic concept), with the origin ($x = 0$) denoting the peak of gender ambiguity. This corresponds to the two auxiliary models in Figure 14, each generating distinct, yet narrowly focused gender representations. As Figure 17 illustrates, at $\lambda_1 = 0.5$, the barycenter distribution ultimately emphasizes sampling from regions where neither generative model assigns high probability. Like celestial bodies pulling one another into balance,



Figure 14: **Left:** uncurated samples from the first (biased) auxiliary model. **Right:** uncurated samples from the second (biased) auxiliary model. Each 6x2 grid of portraits (here and in Figure 15) is generated from the same set of initial latent noise tensors, ensuring comparability across models.



Figure 15: **Left:** uncurated samples from the KL barycenter distribution with $\lambda = (0.5, 0.5)$. **Right:** uncurated samples from a 50-50 checkpoint merged model. The left grid embodies a more neutral, ambiguous gender expressions, such as the absence of mustaches or beards present in the right and Figure 14.



Figure 16: More image samples, continuing Figure 15. **Left:** uncurated samples from the KL barycenter distribution with $\lambda = (0.5, 0.5)$. **Right:** uncurated samples from a 50-50 checkpoint merged model.

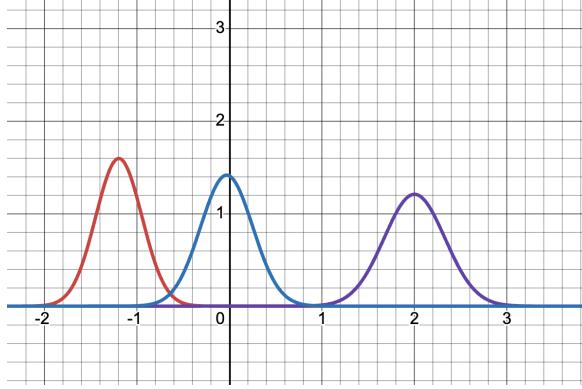


Figure 17: **Blue:** KL-divergence barycenter PDF with $\lambda_1 = 0.5$. **Red** and **Purple**: two auxiliary (unimodal Gaussian) PDFs. Red corresponds to $\lambda_1 = 1$, purple corresponds to $\lambda_1 = 0$.

the diffusion process finds a shared orbit between the biases of both models. The barycenter does not merely split the difference—it crafts a new gravitational midpoint, where the intersection of opposing forces yields something not yet observed, harmonizing their influence on the vast canvas of the latent space.

H ADDITIONAL EXPERIMENT WITH BIMODAL GAUSSIAN MIXTURES

We also test ScoreFusion’s ability to approximate a one-dimensional *bimodal* Gaussian mixture distribution using two auxiliary models that were trained on different Gaussian mixtures. Since the data is synthetic, we can access the true density function of the target distribution and auxiliary distributions, shown in the right of Figure 18; the ground truth distribution is in grey. Table 5 gives the 1-Wasserstein distance \mathcal{W}_1 between the distribution learned by ScoreFusion and the ground truth distribution, calculated using SciPy.

Table 5: 1-Wasserstein distance from the ground truth distribution. Standard error is calculated from the \mathcal{W}_1 distances of 10 random draws of 8096 samples from each generator.

Model	2^5	2^6	2^7
Baseline	106.93 ± 1.43	13.46 ± 0.28	16.74 ± 0.27
ScoreFusion	0.39 ± 0.02	0.51 ± 0.03	0.36 ± 0.02
λ^* of ScoreFusion	[0.62, 0.38]	[0.65, 0.35]	[0.46, 0.54]

Model	2^8	2^9	2^{10}
Baseline	2.13 ± 0.12	0.55 ± 0.04	0.15 ± 0.02
ScoreFusion	0.58 ± 0.03	0.38 ± 0.02	0.30 ± 0.02
λ^* of ScoreFusion	[0.68, 0.32]	[0.61, 0.39]	[0.58, 0.42]

Using only 64 training data, ScoreFusion can already learn a good representation of the ground truth distribution. In contrast, the standard diffusion model is overly widespread and fails to capture the modes of the Gaussian mixture. Moreover, ScoreFusion consistently outperforms the baseline in \mathcal{W}_1 distance when the number of training data is fewer than 2^{10} .

Additional histograms of the distributions learned by ScoreFusion versus the baseline are attached:

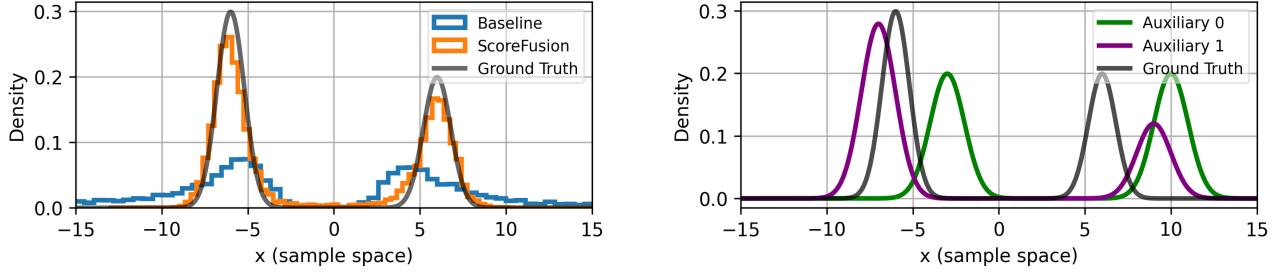


Figure 18: **Left:** Histograms of 8096 ScoreFusion samples and 8096 baseline samples; both models are calibrated / trained on 64 samples. **Right:** Density functions of ground truth vs. the auxiliary distributions.

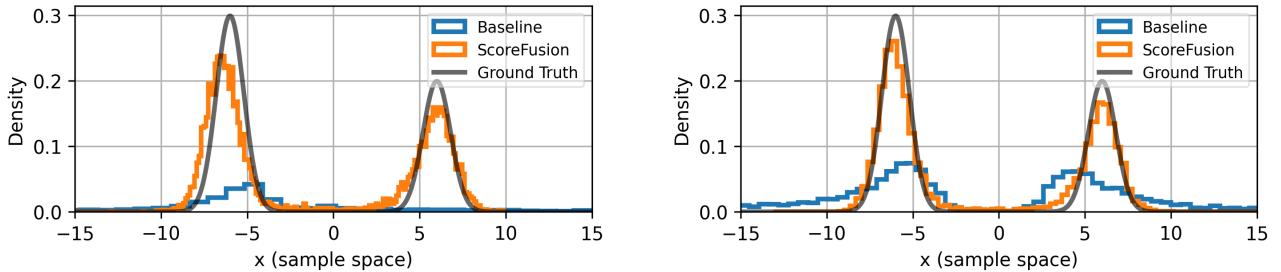


Figure 19: **Left:** Models trained on 32 samples. **Right:** Models trained on 64 samples.

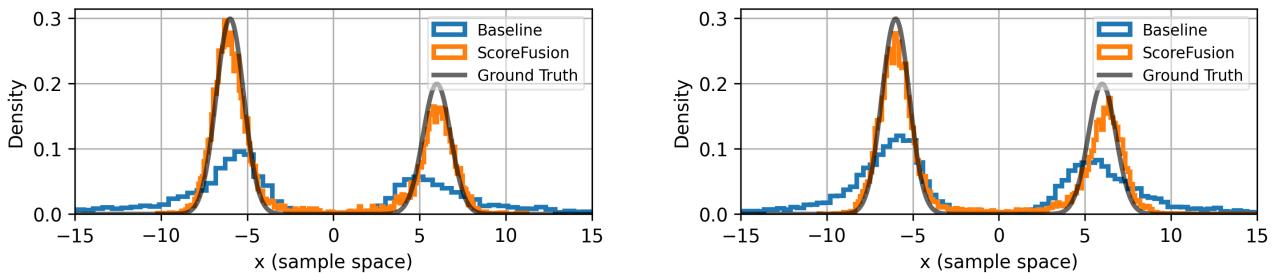


Figure 20: **Left:** Models trained on 128 samples. **Right:** Models trained on 256 samples.

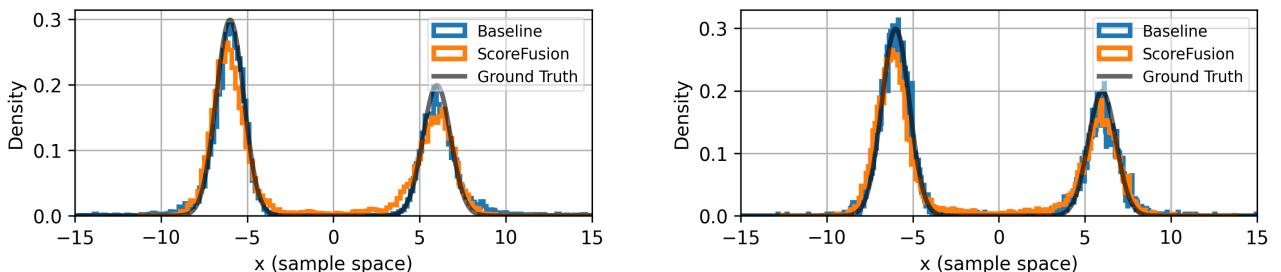


Figure 21: **Left:** Models trained on 512 samples. **Right:** Models trained on 1024 samples.