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

Nian Si

IEDA, HKUST

AISTATS 2025

Joint work with Jose Blanchet, Hao Liu, and Junze Tony Ye (Stanford)

Motivation

Generative AIs are data intense, what if we do not have a lot of data?

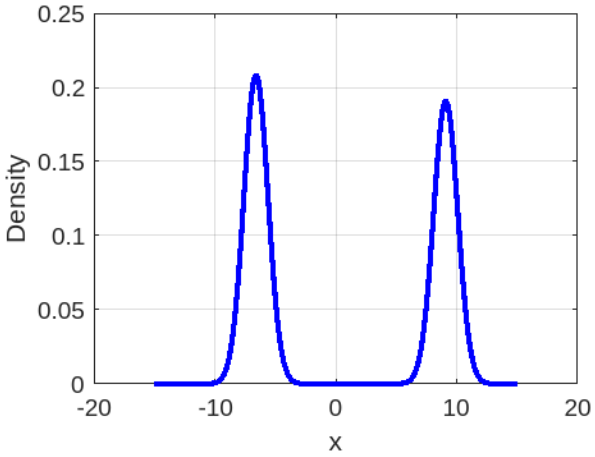
- (1): Find auxilary data sources (transfer learning).
- (2): Fuse auxilary data sources that align with the target data.

Challenges

- (1): How to optimally combine/fuse models?
- (2): How to train to obtain optimal combination weights?

Motivation

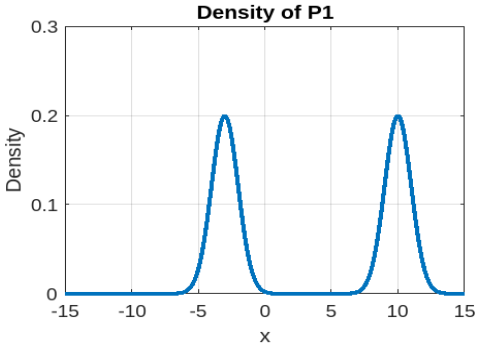
Generate new unknown distribution P^* with density function

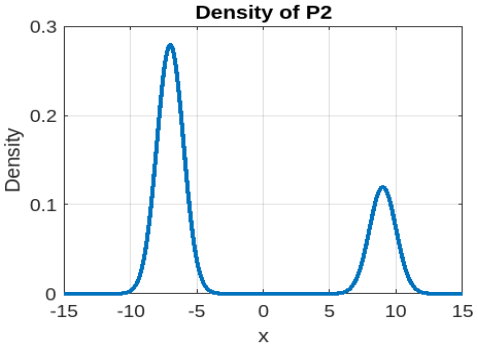


where few samples are available.

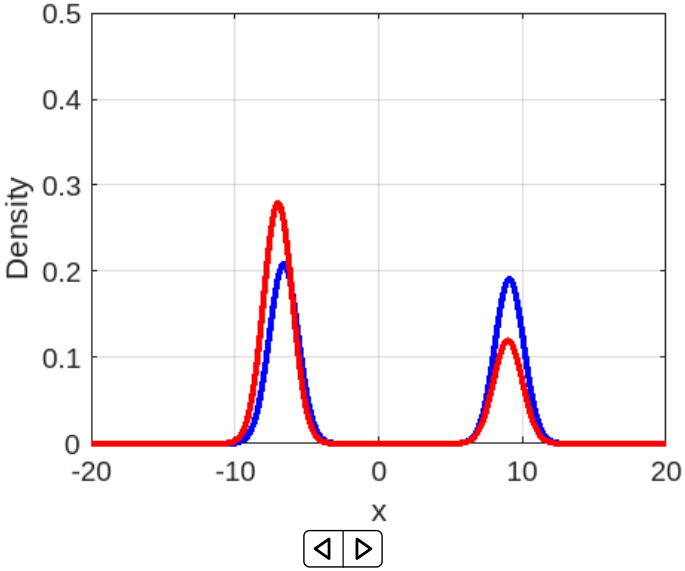
Some Knowledge

Suppose we have good generators for distributions P_1 and P_2





Combining Knowledge



Optimal Fusion

Given densities

$$\mu_1, \mu_2, \ldots, \mu_m,$$

the D-barycenter problem given weights $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m) \in \Delta_m$, where $\Delta_m = \{\boldsymbol{\lambda} \in [0,1]^m : \sum_{i=1}^m \lambda_i = 1\}$. is to find

$$\mu_{\boldsymbol{\lambda}} = \arg\min_{\mu} \sum_{i=1}^{m} \lambda_i D\left(\mu, P_i\right).$$

Optimal Nonparametric Fusion

D can be either a metric or a divergence between two probability measures μ and ν.

- Wasserstein distance: $D(\mu, \nu) = W_p(\mu, \nu)$: theoretically desirable, but hard to compute.
- KL divergence: $D(\mu, \nu) = D_{\mathsf{KL}}(\mu \parallel \nu)$: computationally more tractable.

For each $i = 1, 2, \ldots, m$,

$$D_{\mathsf{KL}}(\mu \parallel \mu_i) = \begin{cases} \int \log\left(\frac{d\mu}{d\mu_i}\right) d\mu, & \text{if } \mu \ll \mu_i \\ \infty, & \text{otherwise.} \end{cases}$$

Direct computation shows that

$$\mu_{\boldsymbol{\lambda}}(x) \propto \prod_{i=1}^{m} \mu_i(x)^{\lambda_i}.$$

Approach: Training Optimal Fusion

Let $\mu_n^\sigma(x) = \sum_{i=1}^n \frac{1}{n}\phi_\sigma(x - x_i)$ be the smoothed empirical density of data under target distribution ν, constructed from limited amount of data. We want to solve the optimization problem

$$\min_{\boldsymbol{\lambda}\in\Delta_m} F(\boldsymbol{\lambda}) = \min_{\boldsymbol{\lambda}\in\Delta_m} D_{\mathsf{KL}}\left(\mu_n^\sigma \parallel \mu_{\boldsymbol{\lambda}}\right) \tag{1}$$

to find optimal weights directly.

Lemma

Suppose the target and reference distributions are all compactly supported with absolutely continuous densities, then Problem 1 is convex in $\boldsymbol{\lambda}$.

Training and Inference Challenges

Given Lemma, Problem 1 is a convex problem on a compact set, so we can use gradient-based iterative method to numerically solve it. The gradient has the closed form

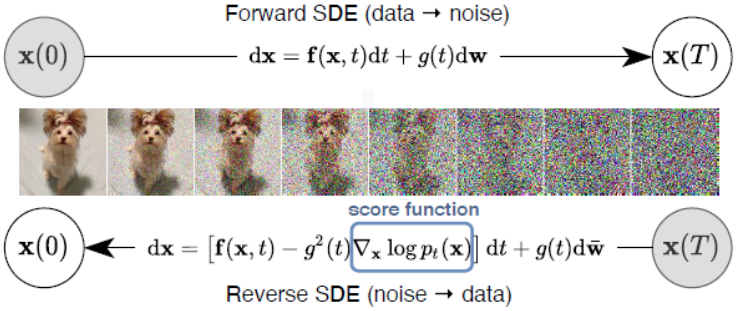
$$\frac{\partial F}{\partial \lambda_i}(\boldsymbol{\lambda}) = -\mathbb{E}_\nu \left[h_i(X)\right] + \frac{\int \exp\left(\sum_{k=1}^m \lambda_k h_k(y)\right) h_i(y) dy}{\int \exp\left(\sum_{k=1}^m \lambda_k h_k(y)\right) dy},$$

where $h_i = \log \mu_i$ for each $i = 1, 2, \ldots, m$ and μ_i is assumed to be estimated.

However, 1) the numerical integration to compute the gradients is difficult in high dimensions; 2) it is also hard to generate samples from the barycenter.

Diffusion Model to Rescue

To deal with the computational challenge, we make use the structure of diffusion model, a generative model establishing a stochastic transport map between an empirically observed, yet unknown, target distribution and a known prior.



Diffusion Model to Rescue

- Forward process:

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

- Backward process:

$$d\tilde{X}(t) = \left(a\tilde{X}(t) + \sigma^2 \nabla \log p_{T-t}\left(\tilde{X}(t)\right)\right) dt + \sigma dW(t), \tilde{X}(0) \sim p_T$$

- Score estimation via score matching:

$$\min_{\theta} \mathbb{E}_{t\sim\mathcal{U}[0,T]}\left[\mathbb{E}_{X(t)\sim p_t}\left[\left\|s_{t,\theta}\left(X(t)\right) - \nabla \log p_t(X(t))\right\|_2^2\right]\right]$$

- Above loss function motivates the loss function in the fusion method.

Diffusion Model to Rescue

Suppose $\mu_1, \ldots, \mu_m$ are auxiliary distributions trained using diffusion score matching. Thus we have the backward SDEs (with pretrained neural networks), for $i = 1, 2, \ldots, m$,

$$d\tilde{X}_i(t) = \left(a\tilde{X}_i(t) + \sigma^2 s^i_{T-t,\theta^*}\left(\tilde{X}_i(t)\right)\right)dt + \sigma dW_i(t), \tilde{X}_i(0) \sim p^i_T.$$

For simplicity, choose $\sigma = 1$.

Diffusion Model to Rescue

Suppose $\mu_1, \ldots, \mu_m$ are auxiliary distributions trained using diffusion score matching. Thus we have the backward SDEs (with pretrained neural networks), for $i = 1, 2, \ldots, m$,

$$d\tilde{X}_i(t) = \left(a\tilde{X}_i(t) + \sigma^2 s^i_{T-t,\theta^*}\left(\tilde{X}_i(t)\right)\right)dt + \sigma dW_i(t), \tilde{X}_i(0) \sim p^i_T.$$

For simplicity, choose $\sigma = 1$.

How to use this to help fusion?

Diffusion KL Barycenter

Theorem (informal)

Suppose for each $i = 1, 2, \ldots, m$, the i-th SDE has the form

$$dX_i(t) = a_i\left(t, X(t)\right) dt + dW_i(t), X_i(0) \sim \mu_i.$$

We further assume, for each $i = 1, 2, \ldots, m$, μ_i has continuous density, then process-level KL barycenter can be represented as the SDE

$$dX(t) = a\left(t, X(t)\right) dt + dW(t), X(0) \sim \mu,$$

where $a(t, x) = \sum_{i=1}^{k} \lambda_i a_i(t, x)$, μ is the distribution-level KL barycenter of reference measures $\mu_1, \ldots, \mu_m$, and W is a standard Brownian motion.

Diffusion KL Barycenter

For a fixed $\boldsymbol{\lambda}$, simulating the (backward) barycenter process

$$d\tilde{X}(t) = \left(a\tilde{X}(t) + \sigma^2 \sum_{i=1}^{m} \lambda_i s^i_{T-t,\theta^*} \left(\tilde{X}(t) \right) \right) dt + \sigma dW(t)$$

from Gaussian gives the $\boldsymbol{\lambda}$-barycenter $\mu_{\boldsymbol{\lambda}} \sim \tilde{X}(T)$.

Diffusion KL Barycenter

Let p_t^ν be the probability measure at time t for the forward process starting from the *target distribution*. Given $\hat{\mu}_1, \ldots, \hat{\mu}_m$ coupled with the processes $\tilde{X}_1, \ldots, \tilde{X}_m$, we minimize over $\boldsymbol{\lambda} \in \Delta_m$

$$\mathbb{E}_{t\sim\mathcal{U}[0,\tilde{T}]}\left[\left(\mathbb{E}_{X(t)\sim p_t^\nu}\left[\left\|\sum_{i=1}^m \left(\lambda_i s^i_{t,\theta^*}\left(X(t)\right)\right) - \nabla \log p_t^\nu(X(t))\right\|_2^2\right]\right)\right]$$

with $\tilde{T} \ll T$ and all pretrained neural networks frozen.

Remarks

Essentially, the features (structures) of auxiliaries are borrowed, making the training linear from the KL barycenter perspective.

- When $\tilde{T}$ is too small, there is numerical instability. This corresponds to difficulty of numerical integration without diffusion model.
- When $\tilde{T}$ is too large, the error is large.

Performance Guarantees

Theorem (Informal)

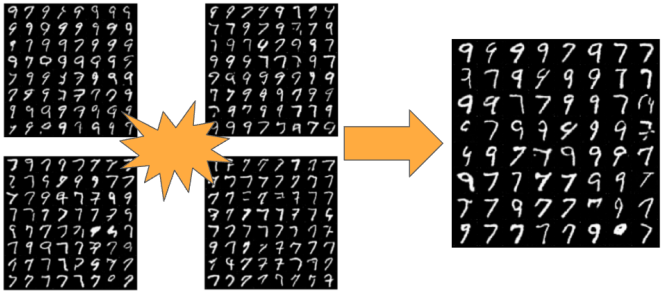
Let n be the number of samples in the target distribution, and we denote the output of the fusion algorithm as $\hat{\nu}_P$ and ν is the target distribution, then with high probability

$$TV(\nu, \hat{\nu}_P) \lesssim \underbrace{TV(\nu, \mu_{\mathbf{\Lambda}^*})}_{\text{quality of combined auxiliaries}} + \underbrace{\mathcal{O}\left(n^{-1/4}\right)}_{\text{sampling errors}} + \underbrace{\sqrt{m}\mathcal{O}\left(\tilde{T}^{1/4}\right)}_{\text{approximation of time 0}}$$

$$+ \underbrace{\mathcal{O}\left(\sqrt{\sum_{i=1}^{m} \mathbb{E}_{p^i_{\cdot}}\left[\left\| s^i_{\cdot,\theta^*} - \nabla \log p^i_{\cdot} \right\|_2^2\right]}\right)}_{\text{quality of auxiliary score estimations}}.$$

Example 1: MNIST with changing frequencies

Setting: The support of all our datasets are hand-drawn 1x28x28 images of 7's and 9's, but their frequencies vary. We previously trained auxiliary U-Nets on four frequencies: $(10\%, 90\%)$, $(30\%, 70\%)$, $(70\%, 30\%)$, $(90\%, 10\%)$. However, the target marginal distribution is $(\mathbf{60}\%, \mathbf{40}\%)$.



Objective: minimize negative log-likelihood & match target diversity

Results: MNIST with changing frequencies

B1 is the baseline where a U-Net is trained from scratch using only the target data.
B2 is the baseline where we directly fine-tune an auxiliary U-Net with frequencies {'7': 70%, '9': 30%}.

Table 1: Digit frequencies estimated by a SOTA classifier.

Digit	Target	2^6			2^8			2^{10}			2^{12}		
		B1	B2	Ours	B1	B2	Ours	B1	B2	Ours	B1	B2	Ours
7	60%	47.9%	72.4%	**55.6%**	66.8%	65.5%	**57.5%**	65.5%	65.1%	**56.6%**	66.7%	65.5%	**59.8%**
9	40%	10.3%	23.2%	**39.4%**	23.8%	29.9%	**38.0%**	26.7%	30.6%	**39.8%**	27.9%	30.4%	**36.7%**
Others	0	41.8%	4.4%	**5.0%**	9.4%	4.6%	**4.5%**	7.8%	4.3%	**3.6%**	5.4%	4.1%	**3.5%**

Example 2: Generating a New Facial Distribution

Task: Given two auxiliary generative models, each with a monotonic human face representation, how to fuse them to generate face images that organically blend features from both subgroups?

Baseline: A popular empirical method called *Checkpoint Merging* [Biggs et al. 2024] creates a new generative model by weighted averaging weights of the two neural networks parametrizing the two models.

Auxilary data sources

Figure 1: **Top**: portraits sampled from the first auxiliary model. It was finetuned on images of people identifying as White male. **Bottom**: portraits sampled from the second auxiliary model. It was finetuned on images of people identifying as Asian female.

Sampling from a Low Probability Region

(a) KL barycenter: gender-neural, smooth transitions

(b) Checkpoint merging: two clusters

Figure 2: Comparision of Portraits sampled from the KL barycenter and checking merging distributions of the two auxiliary models. $\boldsymbol{\lambda} = (0.5, 0.5)$.

Conclusions

1. $\boldsymbol{\lambda}$-KL-barycenter is useful if
   - Target in "KL convex hull" of auxiliaries.
   - Auxiliaries are well-trained.
2. ScoreFusion mitigates curse of dimensionality.
3. Connection to checkpoint merging.
   - ScoreFusion can generate samples from low probability region.
4. Compatibility with existing AI workflow
   - A user can easily adapt the U-Net training pipeline in the popular *Diffusers* library to ScoreFusion training with a few lines of code.

Conclusions