Improving Distribution Matching via Score-Based Priors and Structural Regularization

Anonymous authors
Paper under double-blind review

Abstract

Distribution matching (DM) can be applied to multiple tasks including fair classification, domain adaptation and domain translation. However, traditional variational DM methods such as VAE-based methods unnecessarily bias the latent distributions towards simple priors or fail to preserve semantic structure leading to suboptimal latent representations. To address these limitations, we propose novel VAE-based DM approach which incorporates a flexible score-based prior and a semantic structure preserving regularization. For score-based priors, the key challenge is that computing the likelihood is expensive. Yet, our key insight is that computing the likelihood is unnecessary for updating the encoder and thus we prove that the necessary gradients can be computed using only one score function evaluation. Additionally, we adapted the structure preserving regularization inspired by the Gromov-Wasserstein distance, which explicitly encourages the retention of geometric structure in the latent space, even when the latent space has fewer dimensions than the observed space. Our framework further allows the integration of semantically meaningful structure from pretrained or foundation models into the latent space, ensuring that the representations preserve semantic structure that is informative and relevant to downstream tasks. We empirically demonstrate that our DM approach leads to better latent representations compared to similar methods for fair classification, domain adaptation, and domain translation tasks.

1 Introduction

As machine learning (ML) continues to advance, trustworthy ML systems not only require impressive performance but also properties such as fairness, robustness, causality, and explainability. Unfortunately, collecting more data or building bigger models, as scaling laws (Kaplan et al., 2020) propose, improve performance with larger models and datasets but don't necessary address to solve these problems. For example, historical bias or imbalanced data can cause even well-trained models to produce unfair outcomes, requiring additional constraints to mitigate such biases. Distribution matching (DM), also known as distribution alignment or domain-invariant representation learning, has emerged as a promising approach to address these challenges. By minimizing the divergence between latent representations, distribution matching can introduce additional objectives to ML systems, enabling them to learn representations that are fair, robust, and causal. This approach has been successfully applied to a wide range of problems, including domain adaptation (Ganin et al., 2016; Zhao et al., 2018), domain generalization (Muandet et al., 2013) causal discovery (Spirtes & Zhang, 2016), and fairness-aware learning (Zemel et al., 2013).

Despite the potential of distribution matching (DM) methods, they face significant challenges due to the vast number of possible mappings in the latent space. Without sufficient constraints, these methods often fail to maintain meaningful structural relationships in the learned representations from the data distribution, resulting in suboptimal latent representations for downstream tasks. A popular method for learning representations is the use of variational approaches like Variational Autoencoders (VAEs), which have been widely adopted for their stability during training and their ability to learn meaningful representations (Chen et al., 2019; Burgess et al., 2018).

However, VAEs typically rely on a simple prior—commonly an isotropic Gaussian distribution—over the latent space. While this assumption simplifies optimization and ensures

computational tractability in generative tasks, it biases the latent space, often leading to a significant loss of structural information inherent in the data during transformation. This loss disrupts the preservation of the data's geometric properties, which is particularly critical in unsupervised settings for learning meaningful and robust representations Chen et al. (2020);Uscidda et al. (2024). Recent advancements in manifold learning have highlighted the importance of preserving intrinsic geometry of the data (Uscidda et al., 2024; Nakagawa et al., 2023; Hahm et al., 2024; Lee et al., 2022; Horan et al., 2021; Gropp et al., 2020; Chen et al., 2020). Notably, Uscidda et al. (2024) and Nakagawa et al. (2023) demonstrate that incorporating geometry-preserving constraints can induce disentanglement in the latent space. They propose VAE frameworks that directly regularize the objective using Gromov-Monge optimal transport, leveraging its ability to align latent representations with the data's inherent geometric structure. However, these approaches face significant practical challenges: the simultaneous goals of preserving data geometry and matching a simple prior often result in distortions within the latent space. To address this issue, Nakagawa et al. (2023) advocates for the use of more expressive priors, such as meta-priors, Gaussian mixtures, and neural priors, offering greater flexibility in capturing complex data distributions while preserving geometric consistency. In contrast, Uscidda et al. (2024) retains the use of a simple prior but focuses on learning latent representations that minimize feature distortion as effectively as possible.

The prospect of utilizing powerful and flexible priors is particularly compelling, as they can relax the trade-off between prior matching and data geometry preservation, reducing distortion and achieving better geometric consistency in the latent space. However, we argue that approaches such as Gaussian mixture priors, meta-priors, or expressive neural priors (Vahdat et al., 2021; Makhzani et al., 2016; Tomczak & Welling, 2018) may suffer from practical limitations, including poor scalability to high-dimensional spaces, significant computational expense, or instability during training. To overcome these limitations, we introduce the Score Function Substitution (SFS) trick, a novel approach that leverages a score model to indirectly parameterize the prior distribution. By doing so, our method achieves a balance between memory efficiency, stability during training, and geometric consistency in the latent space, providing a robust solution to the challenges faced by traditional distribution matching frameworks.

We summarize our contributions in the field of DM as follows:

- **Introduction of Score-Based Priors for Flexible Representation:** We propose the Score Function Substitution (SFS) method to learn score-based priors, preserving complex data structures while enhancing the efficiency and stability compared to prior methods.
- **Semantic Structural Preserving Constraints Inspired by Gromov-Wasserstein Distance:** To preserve geometry, we adopt the Gromov-Wasserstein-based constraint from Gromov Wasserstein Autoencoders (GWAE) Nakagawa et al. (2023). Specifically, we advocate for computing the cost function within the semantic space, if available, rather than the raw pixel space, as this approach is more suitable for capturing meaningful relationships in image datasets.
- **Empirical Validation:** Our experiments demonstrate improved downstream task performance in fairness learning, domain adaptation, and domain translation using score-based priors and structural preservation.

2 Preliminaries

Variational Alignment Upper Bound (VAUB) The paper by Gong et al. (2024) presents a novel approach to distribution matching for learning invariant representations. The author proposea a non-adversarial method based on Variational Autoencoders (VAEs), called the VAE Alignment Upper Bound (VAUB). Specifically, they introduce alignment upper bounds for distribution matching that generalize the Jensen-Shannon Divergence (JSD) with VAE-like objectives. The author formalizea the distribution matching problem with the following VAUB objective:

$$\text{VAUB}(q(z|x,d)) = \min_{p(z)} \mathbb{E}_{q(x,z,d)} \left[-\log \frac{p(x|z,d)}{q(z|x,d)} p(z) \right] + C, \tag{1}$$

where $q(z|x,d)$ is the probabilistic encoder, $p(x|z,d)$ is the decoder, $p(z)$ is the shared prior, and C is a constant independent of model parameters. The method ensures that

the distribution matching loss is an upper bound of the Jensen-Shannon divergence (JSD), up to a constant. This non-adversarial approach overcomes the instability of adversarial training, offering a robust, stable alternative for distribution matching in fairness, domain adaptation, and robustness applications. Empirical results show that VAUB and its variants outperform traditional adversarial methods, particularly in cases where model invertibility and dimensionality reduction are required.

Score-based Models *Score-based Models* (Song et al., 2021c) are a class of diffusion models that learn to generate data by denoising noisy samples through iterative refinement. Rather than directly modeling the data distribution $p(x)$, as done in many traditional generative models, score-based models focus on learning the gradient of the log-probability density of the target distribution, known as the score function. To learn the score function, Vincent (2011) and Song & Ermon (2019) propose training on the Denoising Score Matching (DSM) objective. Essentially, data points x are perturbed with various levels of Gaussian noise, resulting in noisy observations $\tilde{x}$. The score model is then trained to match the score of the perturbed distribution. The DSM objective is defined as follows:

$$\text{DSM} = \frac{1}{2L}\mathbb{E}_{q_{\sigma_i}(\tilde{x}|x)p_{\text{data}}(x)}[\|s_\phi(\tilde{x},\sigma_i) - \nabla_{\tilde{x}} \log q_{\sigma_i}(\tilde{x}|x)\|_2^2], \tag{2}$$

where $q_{\sigma_i}(\tilde{x}|x)$ represents the perturbed data distribution of $p_{\text{data}}(x)$, and where L is the number of noise scales $\{\sigma_i\}_{i=1}^L$. When the optimal score network s_ϕ^* is found, $s_\phi^*(x) = \nabla_x \log q_\sigma(x)$ almost surely (Vincent (2011),Song & Ermon (2019)) and approximates $\nabla_x \log p_{\text{data}}(x)$ when the noise is small ($\sigma \approx 0$). Since score-based models learn the gradient of the distribution rather than the distribution itself, generating samples involves multiple iterative refinement steps. These steps typically leverage techniques such as Langevin dynamics, which iteratively updates the sample using the learned score function Song & Ermon (2019).

Gromov-Wasserstein Distance The Optimal Transport (OT) problem seeks the most efficient way to transform one probability distribution into another, minimizing transport cost. Given two probability distributions μ and ν over metric spaces (X, d_X) and (Z, d_z), the OT problem is:

$$\inf_{\pi\in\Pi(\mu,\nu)} \mathbb{E}_{(x,z)\sim\pi}[d(x,z)] \tag{3}$$

where $\Pi(\mu,\nu)$ is the set of couplings with marginals μ and ν, and $d(x,z)$ is a cost function, often the Euclidean distance. The Gromov-Wasserstein (GW) distance extends OT to compare distributions on different metric spaces by preserving their relative structures, not absolute distances. For distributions μ and ν over spaces (X, d_X) and (Z, d_z), the GW distance is:

$$\text{GW}(\mu,\nu) = \inf_{\pi\in\Pi(\mu,\nu)} \mathbb{E}_{(x,z)\sim\pi,(x',z')\sim\pi}[|d_X(x,x') - d_Z(z,z')|^2] \tag{4}$$

$$= \inf_{\pi\in\Pi(\mu,\nu)} \text{GWCost}(\pi(x,z)) \tag{5}$$

3 Methodology

3.1 Training Objective for Distribution Matching with a Score-based Prior

By employing VAUB(Gong et al., 2024) as our distribution matching(DM) objective $\mathcal{L}_{\text{DM}}$,

$$\mathcal{L}_{\text{DM}} = \mathcal{L}_{\text{VAUB}} = \sum_d \frac{1}{\beta}\mathbb{E}_{q_\theta}\left[-\log \frac{p_\varphi(x|z,d)}{q_\theta(z|x,d)^\beta} Q_\psi(z)^\beta\right], \tag{6}$$

where d represents the domain $\forall d \in [1,\cdots,D]$ (e.g., different class datasets or modalities), and $\beta \in [0,1]$ acts as a regularizer controlling the mutual information between the latent variable z and the data x. $q_\theta(z|x,d)$ and $p_\varphi(x|z,d)$ are the d-th domain probabilistic encoder and decoder, respectively, and $Q_\psi(z)$ is a prior distribution that is invariant to domains (Gong et al., 2024). For notational simplicity, we ignore the SP loss and we assume $\beta = 1$. We can split the VAUB objective into three components: reconstruction loss, entropy loss, and cross entropy loss.

$$\mathcal{L}_{\text{VAUB}} \triangleq \sum_d \Big\{ \underbrace{\mathbb{E}_{q_\theta}[-\log p_\varphi(x|z,d)]}_{\text{reconstruction term}} - \underbrace{\mathbb{E}_{q_\theta}[-\log q_\theta(z|x,d)]}_{\text{entropy term}} + \underbrace{\mathbb{E}_{q_\theta}[-\log Q_\psi(z)]}_{\text{cross entropy term}} \Big\} \tag{7}$$

The prior distribution in the cross-entropy term aligns with the encoder's posterior but is often restricted to simple forms like Gaussians or Gaussian mixtures, which can distort the encoder's transformation function Uscidda et al. (2024). To address this, we propose an expressive, learnable prior that adaptively mitigates such distortions, better capturing the underlying data structure.

Learning an arbitrary probabilistic density function (PDF) is often times intractable or computationally expensive as the normalization constant must be computed. Therefore, instead of modeling a neural network directly on the density $Q(z)$, we propose to indirectly parameterize the prior via its score function $\nabla_z \log Q(z)$. But, the problem is that given only the score function, it is difficult to compute the log likelihood of a sample. It is well-known that weighted combinations of score matching losses do not generalize well and only provide an approximation to maximum-likelihood estimation (MLE). Moreover, directly optimizing MLE through the flow interpretation, while theoretically feasible, becomes computationally expensive in practice as it requires solving an ODE at each optimization step Song et al. (2021a). Modeling an arbitrary probabilistic density function (PDF) is computationally expensive due to the intractability of the normalization constant. Therefore, instead of directly modeling the density $Q(z)$, we propose to indirectly parameterize the prior via its score function $\nabla_z \log Q(z)$. While this avoids direct density estimation, the score function alone makes log-likelihood computations difficult. Weighted score matching losses only approximate maximum-likelihood estimation (MLE), and directly optimizing MLE using the flow interpretation becomes computationally prohibitive as it requires solving an ODE at each step Song et al. (2021a). Unlike VAEs, where efficient sampling from the prior is critical, we demonstrate that the distribution matching objective with a score-based prior can be optimized without costly sampling or computing log-likelihood. By reformulating the cross-entropy term as a gradient with respect to the encoder parameters θ, we derive an equivalent expression that retains the same gradient value. This allows us to decouple score function training from the encoder and compute gradients with a single evaluation of the score function. We call this the **Score Function Substitution (SFS)** trick.

Proposition 1 (Score Function Substitution (SFS) Trick)**.** *If $q_\theta(z|x)$ is the posterior distribution parameterized by θ, and $Q_\psi(z)$ is the prior distribution parameterized by ψ, then the* gradient *of the cross entropy term can be written as:*

$$\nabla_\theta \mathbb{E}_{z_\theta \sim q_\theta(z|x)} \left[-\log Q_\psi(z_\theta) \right] = \nabla_\theta \mathbb{E}_{z_\theta \sim q_\theta(z|x)} \Big[-\Big(\underbrace{\nabla_{\bar{z}} \log Q_\psi(\bar{z})\big|_{\bar{z}=z_\theta}}_{\textit{constant w.r.t. } \theta} \Big)^\top z_\theta \Big], \tag{8}$$

where the notation of z_θ emphasizes its dependence on θ and $\cdot|_{\bar{z}=z_\theta}$ denotes that while $\bar{z}$ is equal to z_θ, it is treated as a constant with respect to θ.

The full proof can be seen in Appendix A. In practice, Eqn. 8 detaches posterior samples from the computational graph, enabling efficient gradient computation without additional back-propagation dependencies. Details are provided in the next section. Following Proposition 1, we propose the score-based prior AUB (SAUB) objective defined as follows:

$$\mathcal{L}_{\text{SAUB}} \triangleq \sum_d \Big\{ \mathbb{E}_{z \sim q_\theta(z|x,d)} \Big[-\log p_\varphi(x|z,d) + \log q_\theta(z|x,d) - \Big(\nabla_{\bar{z}} \log Q_\psi(\bar{z})\big|_{\bar{z}=z} \Big)^\top z \Big] \Big\} \tag{9}$$

Since our new loss does not affect terms related to φ, and by Proposition 1, we have $\nabla_{\theta,\varphi} \mathcal{L}_{\text{VAUB}} = \nabla_{\theta,\varphi} \mathcal{L}_{\text{SAUB}}$. However, $\nabla_\psi \mathcal{L}_{\text{VAUB}}$ and $\nabla_\psi \mathcal{L}_{\text{SAUB}}$ are not guaranteed to be equal and are likely different.

3.1.1 Deriving an Alternating Algorithm with Learnable Score-Based Priors

Optimizing the parameters θ, φ, ψ for the VAUB objective differs from the SAUB objective, as $\nabla_\psi \mathcal{L}_{\text{VAUB}} \neq \nabla_\psi \mathcal{L}_{\text{SAUB}}$, making direct optimization intractable. Furthermore, the SAUB objective is complicated by the lack of direct access to the score function. To address this, we train the prior parameters ψ separately from the encoder θ and decoder φ. Prior work Cho et al. (2022); Gong et al. (2024) shows that aligning the prior closely with the encoder's posterior improves the variational bound. Thus, we approximate the prior's score function

using a score model $S_\psi(\cdot)$, trained on the denoising score matching objective with latent samples. This results in two training objectives:

$$\min_{\theta,\varphi} \sum_d \left\{ \mathbb{E}_{z\sim q_\theta(z|x,d)} \left[-\log p_\varphi(x|z,d) + \log q_\theta(z|x,d) - \left(S_\psi(z^*, \sigma_0 \approx 0)\Big|_{z^*=(z+\sigma_0\epsilon)} \right)^\top z \right] \right\}, \tag{10}$$

$$\min_{\psi} \sum_d \left\{ \mathbb{E}_{q_{\sigma_i}(\tilde{z}|z) q_\theta(z|x,d) p_{\text{data}}(x,d)} \left[\| S_\psi(\tilde{z}, \sigma_i) - \nabla_{\tilde{z}} \log q_{\sigma_i}(\tilde{z}|z) \|_2^2 \right] \right\}. \tag{11}$$

Eqn. 11 is the DSM objective, where $q_{\sigma_i}(\tilde{z}|z)$ is the perturbed latent representation, and $p_{\text{data}}(x,d)$ denotes the data distribution for domain d. Eqn. 10 is our SAUB loss with a fixed score model where $\epsilon \sim \mathcal{N}(0, I)$.

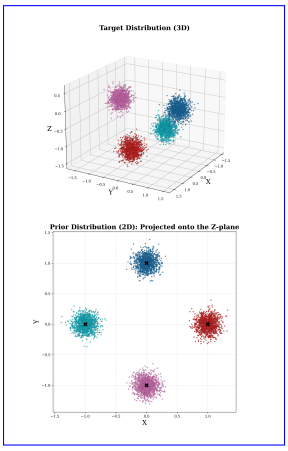
During VAE training, the score model is conditioned on the smallest noise level, $\sigma_0 = \sigma_{\min}$, to approximate the true score function. As previously mentioned, the output of the score model is detached to prevent gradient flow, ensuring memory-efficient optimization by focusing solely on the encoder and decoder parameters without tracking the score model's computational graph. After optimizing the encoder and decoder, these networks are fixed while the score model is updated using Eqn. 11. Theoretically, if the score model is sufficiently trained enough to fully capture latent distribution, it could be optimized using only small noise levels. However, extensive score model updates after each VAE step are computationally expensive. To mitigate this, we reduce score model updates and train with a larger maximum noise level, enhancing stability when the latent representation becomes out-of-distribution (OOD). The complete training process is outlined in Appendix B. We also listed the stabilization and optimization techniques in Appendix C.

3.2 Comparison with Latent Score-Based Generative Models and Connection to Score Distillation

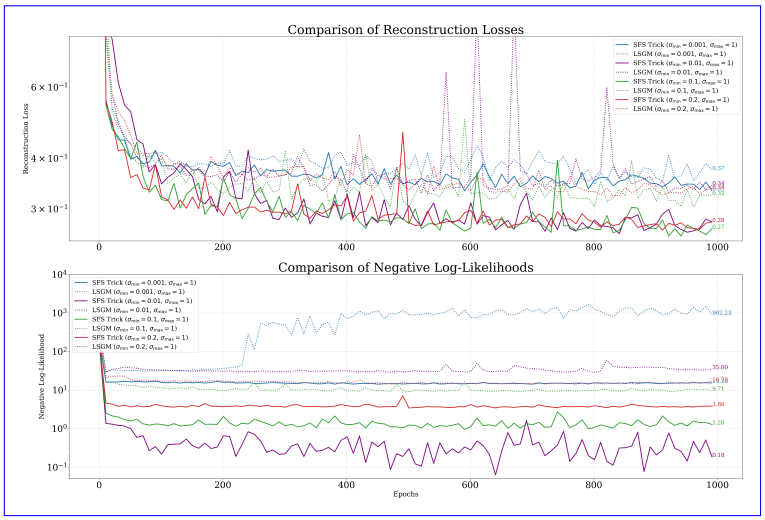
Latent Score-Based Generative Models (LSGM) Vahdat et al. (2021) provide a robust framework that combines latent variable models with score-based generative modeling, leveraging diffusion processes to improve data generation quality. A key innovation in LSGM is the incorporation of a learnable neural network prior. Similar to our approach, LSGM replaces the traditional cross-entropy term in the Evidence Lower Bound (ELBO) with terms involving the score function, approximated using a diffusion model. To elucidate the relationship between LSGM and our Score Function Susbstition (SFS) trick, we turn to the concept of Score Distillation Sampling (SDS) loss. SDS loss was introduced to stabilize the training of Implicit Neural Representation (INR) model parameters by circumventing the computation of the Jacobian term of the diffusion model's U-Net during optimization. Computing this Jacobian term is analogous to approximating the Hessian of the data distribution, which has been empirically shown to be unstable, particularly at low noise levels. Our approach appears to mirror the application of SDS loss within the LSGM framework. Both methods utilize a score model to guide optimization toward higher-density regions while avoiding the computation of the U-Net's Jacobian. Remarkably, this intuition is correct (Appendix E for detailed derivation and explanation). By applying the Sticking-the-Landing principle Roeder et al. (2017) directly to LSGM, we derive that the SFS trick is proportional to a distilled LSGM loss. This technique allows us to update the encoder parameters without backpropagating through the diffusion model, thereby avoiding potential instabilities associated with approximating higher-order derivatives at low noise levels. The full proof of this derivation is provided in subsection E.1.

3.2.1 Comparative Stability: SFS vs. LSGM

We evaluate stability by computing the negative log-likelihood (NLL) of the posterior against a predefined mixture Gaussian prior. Unlike standard training, which updates encoder, decoder, and prior parameters, our approach freezes the prior and uses a score model pre-trained on the defined prior, updating only the encoder and decoder. The same pre-trained score model is used for both SAUB and LSGM to ensure a fair comparison. Performance is evaluated under four minimum noise levels, $\sigma_{\min} \in 0.001, 0.01, 0.1, 0.2$, with $\sigma_{\max} = 1$ fixed. While lower noise levels should improve likelihood estimation, as the score model more



(a) Target/Prior Distribution



(b) Reconstruction Loss/Negative Log Likelihood

Figure 1: (a) The prior distribution is the target distribution projected onto the Z-space. (b) The reconstruction loss and negative log-likelihood are presented on a logarithmic scale for improved visualization. The experiment uses consistent hyperparameters ($\beta = 0.1$), an identical VAE architecture, and the same pretrained score model.

precisely approximates the true score function, LSGM requires backpropagation through the score model's U-Net, which causes instability at low noise levels due to inaccurate gradients. As shown in Fig. 1, when $\sigma_{\min} = 0.001$, LSGM exhibits catastrophic instability, with diverging NLL and spikes in reconstruction loss. At $\sigma_{\min} = 0.1$ and $\sigma_{\min} = 0.2$, LSGM performs worse than at $\sigma_{\min} = 0.01$, indicating that unstable gradients at lower noise levels negatively impacts prior matching. This is concerning since low noise levels, like $\sigma_{\min} = 0.01$, are commonly used in practice. In contrast, the SFS trick shows greater stability across noise levels. At $\sigma_{\min} = 0.01$, the NLL is better than at $\sigma_{\min} = 0.1$, which outperforms $\sigma_{\min} = 0.2$, suggesting that SFS ensures more reliable gradients at lower noise levels. While both LSGM and SAUB degrade at $\sigma_{\min} = 0.001$, SFS stabilizes and achieves a better NLL than LSGM at $\sigma_{\min} = 0.01$, demonstrating its robustness in handling small noise configurations.

3.3 Semantic Preservation (SP) in Latent Representations via GW Inspired Constraint

The Gromov-Wasserstein (GW) distance Section 2 is a powerful tool for preserving structural relationships between distributions in different metric spaces. Nakagawa et al. (2023) introduces the GW metric $\mathcal{L}_{\text{GW}}$ in an autoencoding framework, and we adopt this regularization in a similar manner.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{DM}} + \lambda_{\text{GW}} \mathcal{L}_{\text{GW}}(q_\theta(z|x)) \tag{12}$$

$$\mathcal{L}_{\text{GW}}(q_\theta(z|x)) \triangleq \text{GWCost}(\pi = q_{\text{data}}(x) q_\theta(z|x)) = \mathbb{E}\left[|d_X(x, x') - d_Z(z, z')|^2 \right] \tag{13}$$

where q_{data} represents the data distribution, d_X and d_Z are the predefined metric spaces for the observed and latent spaces, respectively, and λ_{GW} controls the importance of the structural preservation loss. $\mathcal{L}_{\text{DM}}(q_\theta(z|x))$ represents the distribution matching objective with $q_\theta(z|x)$ as the encoder, and $\mathcal{L}_{\text{GW}}(q_\theta(z|x))$ is the structural preservation loss where q_{data} is the data distribution, d_X and d_Z are the metric spaces for the observed and latent spaces, respectively, and λ_{GW} controls the GW loss $\mathcal{L}_{\text{GW}}(q_\theta(z|x))$. $\mathcal{L}_{\text{DM}}(q_\theta(z|x))$ is the distribution matching objective with encoder $q_\theta(z|x)$.

Selection of Metric Space and Distance Functions The GW framework's key strength lies in its ability to compare distributions across diverse metric spaces, where the choice of metric significantly impacts comparison quality. In low-dimensional datasets like Shape3D (Kim & Mnih, 2018) and dSprites (Matthey et al., 2017), Euclidean pixel-level distances align well with semantic differences, leading prior works (Nakagawa et al., 2023; Uscidda

et al., 2024) to use L2 or cosine distances for isometric mappings. However, this breaks down in high-dimensional data, like real-world images, which lie on lower-dimensional manifolds. The curse of dimensionality causes traditional metrics, such as pixel-wise distances, to lose effectiveness as dimensionality increases. Recent advancements in vision-language models like CLIP (Radford et al., 2021) have shown their ability to learn robust and expressive image representations by training on diverse data distributions Fang et al. (2022). Studies Yun et al. (2023) demonstrate that CLIP captures meaningful semantic relationships, even learning primitive concepts. Therefore, we propose using the semantic embedding space of pre-trained CLIP models as a more effective metric for computing distances between datasets, which we define as the Semantic Preservation (SP) loss. For a detailed evaluation of the improvements from using CLIP embeddings, please refer to the Appendix F, which includes demonstrations and additional results. In the following section, we will denote the Gromov-Wasserstein constraint as GW-EP, and GW-SP to differentiate the metric space we used for Gromov-Wasserstein constraint as Euclidean metric space Preservation (EP) and Semantic Structural Preservation (SP) respectively.

4 Related Works

Learnable Priors Most variational autoencoders (VAEs) typically use simple Gaussian priors due to the computational challenges of optimizing more expressive priors and the lack of closed-form solutions for their objectives. Early efforts to address this, such as Adversarial Autoencoders (AAEs) Makhzani et al. (2016), employed adversarial networks to learn flexible priors, resulting in smoother and more complete latent manifolds. Subsequent research Hoffman & Johnson (2016); Johnson et al. (2017) highlighted that simple priors can lead to over-regularized and less informative latent spaces, while Tomczak & Welling (2018) empirically showed that more expressive priors improve generative quality, with significant gains in log-likelihood. More recently, Latent Score-based Generative Models (LSGM) Vahdat et al. (2021) introduced score-based priors, leveraging a denoising score-matching objective to learn arbitrary posterior distributions. This approach enables high-quality image generation while capturing the majority of the data distribution.

Gromov-Wasserstein Based Learning Gromov-Wasserstein (GW) distance has found numerous applications in learning problems involving geometric and structural configuration of objects or distributions. Moreover, the GW metric has been adopted for mapping functions in deep neural networks. One of the key benefits of GW distance is its capacity to compare distributions with heterogeneous data and/or dimensional discrepancies. Prior works, such as Truong et al. (2022); Carrasco et al. (2024), although uses GW distance as part of the loss in the the objective but is focusing on calculating and minimizing the GW objective in the embedding space between domains $\mathcal{L}_{OT/GW} = OT/GW(z_{src}, z_{tgt})$. On the other hand, Uscidda et al. (2024) defineds the GW objective as being calculated between the data dimension and the embedding dimension.

5 Experiments

In this section, we evaluate the effectiveness of our proposed VAUB with a score-based prior on several tasks. We conduct experiments on synthetic data, domain adaptation, multi-domain matching, fairness evaluation, and domain translation. For each experiment, we compare our methods to VAUB and other baselines and evaluate performance using various metrics.

5.1 Improving Latent Space Separation by Using Score-based Prior

The primary objective of this experiment is to demonstrate the performance of different prior models within the VAUB framework. Additionally, we examine the effect of varying the number of samples used during training, specifically considering scenarios with limited dataset availability. To achieve this, we create a synthetic nested D-shaped dataset consists of two domains and two labels, as illustrated in Fig. 2. The aim is to learn a shared latent representation across two domains and evaluate the degree of separation between class labels within this shared latent space. Since downstream tasks rely on these shared latent representations, better separation of class labels in the latent space naturally leads to improved classification performance. This setup draws an analogy to domain adaptation tasks, where the quality of separation in the latent representation relative to the label space plays a critical role in determining downstream classification outcomes.

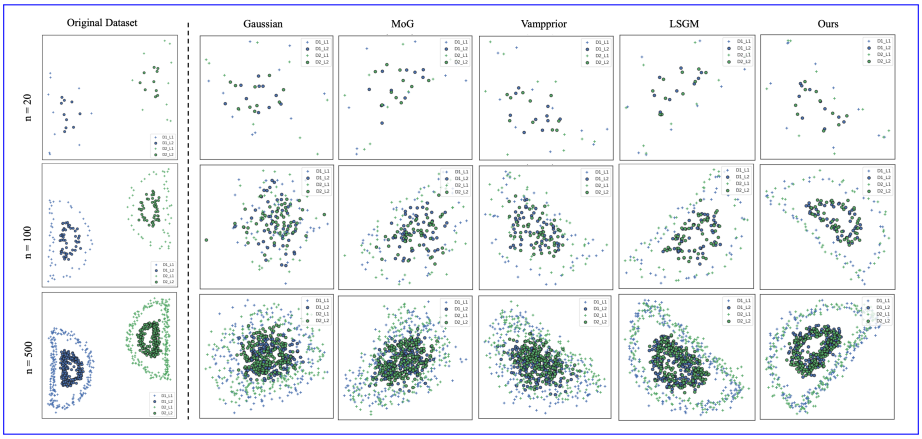


Figure 2: The dataset consists of two domains: Domain 1 (left nested 'D-shaped') and Domain 2 (right flipped 'D-shaped'). In each domain, the outer 'D' corresponds to Label 1, and the inner 'D' to Label 2. The shared latent spaces are visualized for models trained with varying data sizes ($n = 20, 100, 500$ samples) using Gaussian(Kingma et al., 2019), Mixture of Gaussians(Gong et al., 2024), Vampprior(Tomczak & Welling, 2018), LSGM,(Song et al., 2021c) and our score-based model (columns). Legends follow the format `D{domain_index}_L{label_index}`

In this experiment, we control the total number of data samples generated for the dataset, and compare the model's performance using five types of priors: Gaussian prior, Mixture of Gasussian Prior(MoG), Vampprior, and a score-based prior trained with LSGM, and ours (SFS method). Considering the strong relations between pointwise distance and the label information of the dataset, we use GW-EP to compute the constraint loss in both in the data domain and the latent domain. This helps to better visually reflect the underlying structure and separations in the latent space. As shown in Fig. 3, this performance improvement is evident in the latent space: the nested D structure is well-preserved under transformation with the two score-based prior method (LSGM and ours), resulting in well-separated latent representations across different classes. This holds consistently true for varying numbers of data points, from as low as 20 samples to higher counts. On the other hand, the Gaussian prior, MoG and Vamprior only achieves 90% of separation in the latent space when the number of data samples is sufficiently large ($n = 100$ for MoG and Vampprior prior and $n = 20$ for Gaussian prior), allowing the inner and outer classes to have a classifier bound supported by enough data points as shown in Fig. 3. This finding is especially relevant for real-world datasets, where the original data dimensionality can easily reach upto tens of thousands; while in this experiment, we worked with only a two-dimensional dataset, yet the Gaussian, MoG and Vampprior required more than hundreds of samples to achieve effective latent separation, whereas the score-based prior (LGSM and SFS) succeeded with as few as 20 samples.

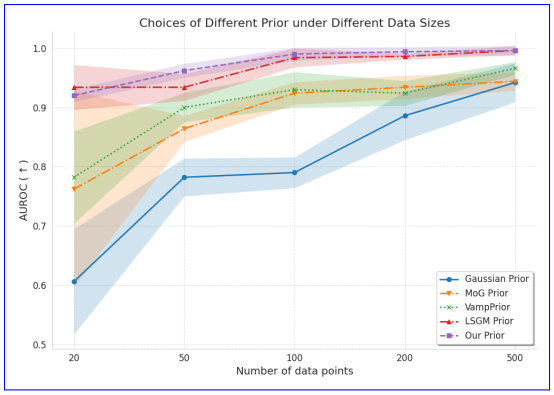


Figure 3: This figure shows label separation in the latent space under varying sample sizes and prior configurations, quantified by AUROC scores from the prediction of support vector classifier. Higher scores indicate better separation. Details of the metric are descriped in the appendix.

5.2 Improving the Tradeoff between Accuracy and Parity on Fairness Representation Learning

For this experiment, we apply our model to the well-known Adult dataset, derived from the 1994 census, which contains 30K training samples and 15K test samples. The target task is to predict whether an individual's income exceeds \$50K, with gender (a binary attribute in this case) considered as the protected attribute. We adopt the same preprocessing steps in Zhao et al. (2020), and the encoder and classifier architectures are consistent with those in Gupta et al. (2021). We adapt GW-EP as our constraint loss considering the lack of semantic models in tabular dataset such as Adult dataset. Please refer to Appendix H for more detailed architecture setup. For comparison, we benchmark our model against three non-adversarial models FCRL(Gupta et al., 2021), CVIB(Moyer et al., 2018), VAUB(Gong et al., 2024) and one adversarial model LAFTR-DP(Madras et al., 2018) and one extra baseline 'Unfair Classifier' which is obtained to serve as a baseline, computed by training the classifier directly on the original dataset.

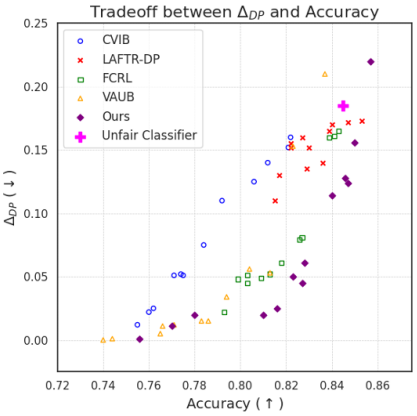


Figure 4: Demographic Parity gap (Δ_{DP}) vs. Accuracy trade-off for UCI Adult dataset. Lower Δ_{DP} is better, and higher **Accuracy** is better.

As illustrated in Fig. 4, our method not only retains the advantages of the SAUB method, achieving near-zero demographic parity (DP) gap while maintaining accuracy, but it also improves accuracy across the board under the same DP gap comparing to other methods. We attribute this improvement largely to the introduction of the score-based prior, which potentially allows for better semantic preservation in the latent space, enhancing both accuracy and fairness.

5.3 Domain Adaptation

We evaluate our method on the MNIST-USPS domain adaptation task, transferring knowledge from the labeled MNIST (70,000 images) to the unlabeled USPS (9,298 images) without using target labels. We compare our SAUB method (with and without structure-preserving constraints) against baseline DA methods: ADDA (Zhao et al., 2018), DANN (Ganin et al., 2016), and VAUB (Gong et al., 2024). All methods use the same encoder and classifier architecture for fairness, with structure-preserving constraints applied using $L2$ distance in Euclidean space(GW-EP) and CLIP embedding(GW-SP).

Model	MNIST to USPS (%)	USPS to MNIST (%)
ADDA	89.4	90.1
DANN	77.1	73
VAUB	40.7	45.3
Ours w/o GW	88.1	85.54
Ours w/ GW-EP	91.4	92.7
Ours w/ GW-SP	**96.1**	**97.4**

Table 1: Domain adaptation accuracy (%) for MNIST to USPS and USPS to MNIST tasks.

As shown in Table 1, our method outperforms the baselines in both directions. Unlike ADDA and DANN, which require joint classifier and encoder training, our approach allows for classifier training after the encoder is learned, simplifying domain adaptation. Additionally, the inclusion of a decoder enables our model to naturally adapt to domain translation tasks, as demonstrated in Fig. 14. We additionally conduct novel experiments to assess the generalizability and robustness of our model with limited source-labeled data, detailed in Appendix D. Additionally, image translation results between MNIST and USPS are presented in Appendix J.

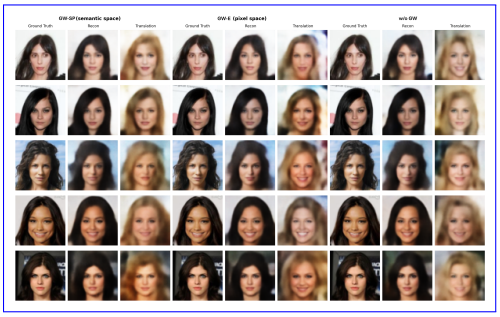
5.4 Domain Translation

We conduct domain translation experiments on the CelebA dataset, translating images of females with blonde hair to black hair and vice versa. We compare three settings: GW loss in semantic space, GW loss in Euclidean space, and no GW loss. This comparison shows that GW loss in the semantic space better preserves semantic features, while Euclidean space GW loss is less effective in high-dimensional settings. We want to note that achieving state-of-the-art image translation performance is not the primary objective of our work; instead, this experiment demonstrates our model's versatility across tasks.

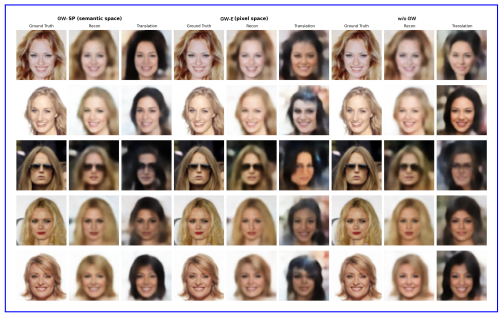
Task/Model	Top-1 (%)	Top-5 (%)	Top-10 (%)	Top-20 (%)
Black-to-Blonde Hair				
No GW	5.0 ± 1.4	14.6 ± 2.4	24.4 ± 4.0	40.0 ± 3.5
GW-EP	4.0 ± 1.0	11.6 ± 2.2	22.0 ± 2.9	35.0 ± 2.6
GW-SP	$\mathbf{9.0 \pm 1.6}$	$\mathbf{27.8 \pm 3.1}$	$\mathbf{39.2 \pm 4.2}$	$\mathbf{59.0 \pm 2.9}$
Blonde-to-Black Hair				
No GW	3.4 ± 1.7	10.8 ± 3.3	19.0 ± 2.9	33.4 ± 3.9
GW-EP	2.0 ± 0.7	9.2 ± 1.8	15.8 ± 2.6	30.4 ± 3.1
GW-SP	$\mathbf{4.8 \pm 2.3}$	$\mathbf{18.8 \pm 3.4}$	$\mathbf{28.6 \pm 4.1}$	$\mathbf{46.2 \pm 2.5}$

Table 2: Top-k retrieval accuracy (%) for semantic preservation experiments. Bold values indicate the best performance for each metric.

For quantitative evaluation of semantic preservation, we utilize image retrieval accuracy as our metric. The models, trained for 1,500 epochs, translate images from a domain of 100 females with black hair to a domain of 100 females with blonde hair and vice versa. For each translated image, we compute the cosine similarity with all translated images in the target domain using CLIP embeddings To ensure fairness, we use a different pretrained CLIP model for evaluation and for training GW-SP for more information see Appendix H. This process is repeated five times with randomly selected datasets to account for variability in the data. The experiment aims to measure how well the translated images preserve their semantic content. We compute the top-k accuracy, where the task is to retrieve the correct translated image from the set of all translated images. This bidirectional evaluation black-to-blonde and blonde-to-black ensures robustness and highlights the model's ability to maintain semantic consistency during translation. The results show that applying GW-EP harms performance in high-dimensional datasets due to poor distance scaling. In contrast, GW-SP in semantic space consistently improves accuracy. Notably, GW-EP performs worse than no GW loss. The domain translation images in Appendix L confirm that models with semantic space GW loss better preserve semantic features like hairstyle, smile, and facial structure, demonstrating its advantage. For additional experiments, we provide image translations between male and female subjects on the FairFace dataset in Appendix K for interested readers.



(a) Black to Blonde Hair Female Translation



(b) Blonde to Black Hair Female Translation

Figure 5: All models use the same architecture. Refer to Appendix H for details on the neural network and CLIP model. Applying GW loss in the CLIP semantic space shows superior semantic preservation in both (a) and (b). The samples are selectively chosen to represent diverse variations; random samples are in Appendix L.

6 Discussion and Conclusion

In conclusion, we introduce score-based priors and structure-preserving constraints to address the limitations of traditional distribution matching methods. Our approach uses score models to capture complex data distributions while maintaining geometric consistency. By applying Gromov-Wasserstein constraints in the semantic CLIP embedding space, we preserve meaningful relationships without the computational cost of expressive priors. Our experiments demonstrate improved performance in tasks like fairness learning, domain adaptation, and domain translation.

References

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae, 2018. URL https://arxiv.org/abs/1804.03599.

Xavier Aramayo Carrasco, Maksim Nekrashevich, Petr Mokrov, Evgeny Burnaev, and Alexander Korotin. Uncovering challenges of solving the continuous gromov-wasserstein problem, 2024. URL https://arxiv.org/abs/2303.05978.

Nutan Chen, Alexej Klushyn, Francesco Ferroni, Justin Bayer, and Patrick Van Der Smagt. Learning flat latent manifolds with vaes. *arXiv preprint arXiv:2002.04881*, 2020.

Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders, 2019. URL https://arxiv.org/abs/1802.04942.

Wonwoong Cho, Ziyu Gong, and David I. Inouye. Cooperative distribution alignment via jsd upper bound. In *Neural Information Processing Systems (NeurIPS)*, dec 2022.

Alex Fang, Gabriel Ilharco, Mitchell Wortsman, Yuhao Wan, Vaishaal Shankar, Achal Dave, and Ludwig Schmidt. Data determines distributional robustness in contrastive language image pre-training (clip), 2022. URL https://arxiv.org/abs/2205.01397.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Ziyu Gong, Ben Usman, Han Zhao, and David I. Inouye. Towards practical non-adversarial distribution matching. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, May 2024.

Amos Gropp, Matan Atzmon, and Yaron Lipman. Isometric autoencoders. *arXiv preprint arXiv:2006.09289*, 2020.

Umang Gupta, Aaron M Ferber, Bistra Dilkina, and Greg Ver Steeg. Controllable guarantees for fair outcomes via contrastive information estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7610–7619, 2021.

Jaehoon Hahm, Junho Lee, Sunghyun Kim, and Joonseok Lee. Isometric representation learning for disentangled latent space of diffusion models. *arXiv preprint arXiv:2407.11451*, 2024.

Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, 2016.

Daniella Horan, Eitan Richardson, and Yair Weiss. When is unsupervised disentanglement possible? *Advances in Neural Information Processing Systems*, 34:5150–5161, 2021.

Matthew J. Johnson, David Duvenaud, Alexander B. Wiltschko, Sandeep R. Datta, and Ryan P. Adams. Composing graphical models with neural networks for structured representations and fast inference, 2017.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2649–2658. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kim18b.html.

Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

Yonghyeon Lee, Sangwoong Yoon, MinJun Son, and Frank C Park. Regularized autoencoders for isometric representation learning. In *International Conference on Learning Representations*, 2022.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pp. 3384–3393. PMLR, 2018.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders, 2016.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Daniel Moyer, Shuyang Gao, Rob Brekelmans, Aram Galstyan, and Greg Ver Steeg. Invariant representations without adversarial training. *Advances in Neural Information Processing Systems*, 31, 2018.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 10–18, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/muandet13.html.

Nao Nakagawa, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Gromov-wasserstein autoencoders, 2023. URL https://arxiv.org/abs/2209.07007.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

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.

Geoffrey Roeder, Yuhuai Wu, and David Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference, 2017. URL https://arxiv.org/abs/1703.09194.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34: 1415–1428, 2021a.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models, 2021b. URL https://arxiv.org/abs/2101.09258.

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

Peter Spirtes and Kun Zhang. Causal discovery and inference: concepts and recent methodological advances. In *Applied informatics*, volume 3, pp. 1–28. Springer, 2016.

Jakub Tomczak and Max Welling. Vae with a vampprior. In *International conference on artificial intelligence and statistics*, pp. 1214–1223. PMLR, 2018.

Thanh-Dat Truong, Naga Venkata Sai Raviteja Chappa, Xuan Bac Nguyen, Ngan Le, Ashley Dowling, and Khoa Luu. Otadapt: Optimal transport-based approach for unsupervised domain adaptation, 2022. URL https://arxiv.org/abs/2205.10738.

Théo Uscidda, Luca Eyring, Karsten Roth, Fabian Theis, Zeynep Akata, and Marco Cuturi. Disentangled representation learning with the gromov-monge gap, 2024. URL https://arxiv.org/abs/2407.07829.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Tian Yun, Usha Bhalla, Ellie Pavlick, and Chen Sun. Do vision-language pretrained models learn composable primitive concepts?, 2023. URL https://arxiv.org/abs/2203.17271.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 325–333, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/zemel13.html.

Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Han Zhao, Amanda Coston, Tameem Adel, and Geoffrey J. Gordon. Conditional learning of fair representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkekl0NFPr.

Contents

1 Introduction **1**

2 Preliminaries **2**

3 Methodology **3**

3.1 Training Objective for Distribution Matching with a Score-based Prior 3

3.1.1 Deriving an Alternating Algorithm with Learnable Score-Based Priors 4

3.2 Comparison with Latent Score-Based Generative Models and Connection to Score Distillation 5

3.2.1 Comparative Stability: SFS vs. LSGM 5

3.3 Semantic Preservation (SP) in Latent Representations via GW Inspired Constraint 6

4 Related Works **7**

5 Experiments **7**

5.1 Improving Latent Space Separation by Using Score-based Prior 7

5.2 Improving the Tradeoff between Accuracy and Parity on Fairness Representation Learning 8

5.3 Domain Adaptation 9

5.4 Domain Translation 9

6 Discussion and Conclusion **10**

Table of Contents **14**

A Proof of Proposition 1 **16**

B Pseudo-code for learning VAUB with Score-Based Prior **16**

C Stabilization and Optimization Techniques **16**

D Limited Source Label for Domain Adaptation **17**

D.1 Results 17

E More Detailed Discussion of Gradient Comparison Between LSGM and SFS Trick **18**

E.1 Proof: SFS trick is proportional to a distilled LSGM loss 19

F Choices of different metric spaces in different dataset **21**

G Multi-Domain Distribution Matching Setting **22**

H Detailed Architecture of the model **22**

H.1 Fairness Representation Learning 22
H.2 Separation Metric for Synthetic Dataset 22
H.3 Domain Adaptation VAE Model 22
H.4 Domain Translation VAE Model 23
H.5 Domain Adaptation Classifier 23
H.6 Pretrained CLIP 23

I More Synthetic Dataset Results **23**

J Image translation between MNIST and USPS **30**

K FairFace Image Translation **30**
K.1 Handpicked samples 30
K.2 Random Samples 30

L Additional Random Image Translations on CelebA **31**

A Proof of Proposition 1

Proposition 1 (Score Function Substitution (SFS) Trick)

If $q_\theta(z|x)$ is the posterior distribution parameterized by θ, and $Q_\psi(z)$ is the prior distribution parameterized by ψ, then the *gradient* of the cross entropy term can be written as:

$$\nabla_\theta \mathbb{E}_{z_\theta \sim q_\theta(z|x)} \left[-\log Q_\psi(z_\theta) \right] = \nabla_\theta \mathbb{E}_{z_\theta \sim q_\theta(z|x)} \big[-z_\theta^T \underbrace{\nabla_{\bar{z}} \log Q_\psi(\bar{z})\big|_{\bar{z}=z_\theta}}_{\text{constant w.r.t. } \theta} \big], \tag{14}$$

where the notation of z_θ emphasizes its dependence on θ and $\cdot|_{\bar{z}=z_\theta}$ denotes that while $\bar{z}$ is equal to z_θ, it is treated as a constant with respect to θ.

Proof.

$$\nabla_\theta \mathbb{E}_{z_\theta \sim q_\theta(z|x)} \left[-\log Q_\psi(z_\theta) \right] \tag{15}$$

$$= \nabla_\theta \mathbb{E}_{\epsilon \sim p(\epsilon)} \left[-\log Q_\psi(g_\theta(\epsilon)) \right] \quad \text{(Reparameterization trick: } z_\theta = g_\theta(\epsilon)\text{)} \tag{16}$$

$$= \mathbb{E}_{\epsilon \sim p(\epsilon)} \left[\nabla_\theta \left(-\log Q_\psi(g_\theta(\epsilon)) \right) \right] \tag{17}$$

$$= \mathbb{E}_{\epsilon \sim p(\epsilon)} \left[\frac{\partial g_\theta(\epsilon)}{\partial \theta}^\top \frac{\partial \log Q_\psi(\bar{z})}{\partial \bar{z}}\Big|_{\bar{z}=g_\theta(\epsilon)} \right] \quad \text{(Chain rule: differentiating at } g_\theta(\epsilon)\text{)} \tag{18}$$

$$= \mathbb{E}_{\epsilon \sim p(\epsilon)} \left[\nabla_\theta g_\theta(\epsilon)^\top \frac{\partial \log Q_\psi(\bar{z})}{\partial \bar{z}}\Big|_{\bar{z}=g_\theta(\epsilon)} \right] \quad \text{(Simplify notation)} \tag{19}$$

$$= \mathbb{E}_{\epsilon \sim p(\epsilon)} \nabla_\theta \left[\left(\frac{\partial \log Q_\psi(\bar{z})}{\partial \bar{z}}\Big|_{\bar{z}=g_\theta(\epsilon)} \right)^\top g_\theta(\epsilon) \right] \quad \text{(Move } \nabla_\theta \text{ outside)} \tag{20}$$

$$= \nabla_\theta \mathbb{E}_{\epsilon \sim p(\epsilon)} \left[-\left(\nabla_{\bar{z}} \log Q_\psi(g_\theta(\epsilon))\Big|_{\bar{z}=g_\theta(\epsilon)} \right)^\top g_\theta(\epsilon) \right] \quad \text{(Gradient applied to parts dependent on } \theta\text{)} \tag{21}$$

$$= \nabla_\theta \mathbb{E}_{z_\theta \sim q_\theta(z|x)} \left[-\left(\nabla_{\bar{z}} \log Q_\psi(z_\theta)\Big|_{\bar{z}=z_\theta} \right)^\top z_\theta \right] \quad \text{(Change back to } z_\theta \text{ after pulling out gradient)} \tag{22}$$

□

B Pseudo-code for learning VAUB with Score-Based Prior

See Alg. 1.

C Stabilization and Optimization Techniques

Several factors, such as interactions between the encoder, decoder, and score model, as well as the iterative nature of the optimization process, can introduce instability. To mitigate these issues, we implemented stabilization and optimization techniques to ensure smooth and robust training.

Batch Normalization on Encoder Output (Without Affine Learning) Applying batch normalization to the encoder's mean output without affine transformations facilitates smooth transitions in the latent space, acting as a soft distribution matching mechanism. By centering the mean and mitigating large shifts, it prevents disjoint distributions, allowing the score model to keep up with the encoder's updates. This regularization ensures the

Algorithm 1 Training VAUB with Score-based Prior (Alternating Optimization)

Input: Data x, domain d, parameters $\{\theta_d, \varphi_d, \psi\}$, hyperparameters: noise levels $\{\sigma_{\min}, \sigma_{\max}\}$, number of loops L for score model update

Initialize: Parameters of Encoders θ, Decoders φ, and Score model ψ
while not converged **do**
    Step 1: Update Encoder and Decoder parameters $\{\theta, \varphi\}$
    Draw $x, d \sim p_{\text{data}}(x, d)$
    Draw $z \sim q_\theta(z|x, d)$
    Calculate score by computing $S_\psi(z^*, \sigma = \sigma_{\min})$ using z^*, which is detached from the computational graph
    Compute the following objective in Eqn. 10:
    Perform gradient descent to minimize the objective and update $\{\theta, \varphi\}$
    Step 2: Update Score Model parameters ψ
    for loop = 1 to L **do** ▷ Number of loops for score model update
        Draw $x, d \sim p_{\text{data}}(x, d)$
        Draw $z \sim q_\theta(z|x, d)$
        Draw perturbed latent variable $\tilde{z} \sim q_{\sigma_i}(\tilde{z}|z)$, where $\sigma_i \in [\sigma_{\min}, \sigma_{\max}]$
        Compute the DSM loss for the score model in Eqn. 11:
        Perform gradient descent to minimize the DSM objective and update ψ
    end for
    Repeat alternating optimization steps until convergence.
end while

latent space remains within regions where the score model is trained, enhancing stability and reducing the risk of divergence.

Gaussian Score Function for Undefined Regions: To further stabilize training, we incorporate a small Gaussian score function into the score model to handle regions beyond the defined domain of the score function (i.e., outside the maximum noise level, $\sigma_{\max}$). Inspired by the mixture neural score function in LSGMs Vahdat et al. (2021), this approach blends score functions to address out-of-distribution latent samples. The Gaussian score ensures smooth transitions and prevents instability in poorly defined areas of the latent space, maintaining robustness even in undertrained regions of the score model.

Weight Initialization and Hyperparameter Tuning: We observed that the initialization of weights significantly impacts the stability and convergence of our model. Poor initialization can lead to bad alignment. Therefore, gridsearch was used to find an optimal weight scale.

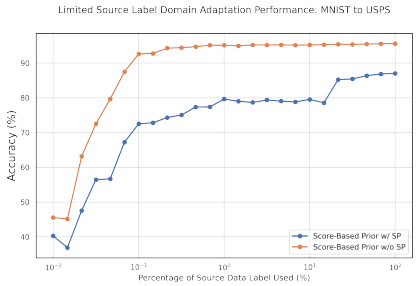
D Limited Source Label for Domain Adaptation

We introduce, to the best of our knowledge, a novel downstream task setup where there is limited labeled data in the source domain (i.e., $1\%, 5\%, 10\%$) and no supervision in the target domain. We apply this setup to the MNIST-to-USPS domain adaptation task. The objective is to determine how well our model with and without structural preservation can generalize with limited source supervision.

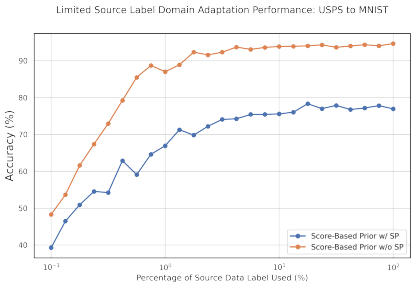
D.1 Results

As shown in Figure Fig. 6, our method without the SP constraint (which is entirely unsupervised in the source domain) demonstrates remarkable sample efficiency. With as little as 0.04% of the dataset (roughly two images per class), our method achieves an accuracy of around 40%. By increasing the labeled data to just 0.1% (about five images per class), the accuracy surpasses 73%. When we introduce the structural preservation constraint, which allows the model to transfer knowledge from a pretrained model, we observe a significant improvement in performance. With only 0.2% of the labeled data, the model's accuracy approaches the performance of models trained on the full dataset. This boost in performance shows the effectiveness of incorporating semantic information into the latent space, allowing the model to generalize better with minimal supervision.

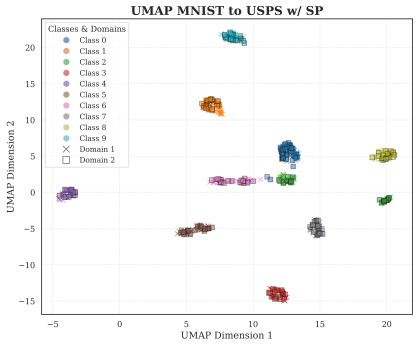
The performance gap between models with and without the structural preservation (SP) constraint becomes more evident through UMAP visualizations of the latent space (Figure Fig. 6). While both methods achieve distribution matching and show label separation, the model without SP struggles to distinguish structurally similar digits, such as "4" and "9". In contrast, with the SP constraint, the latent space exhibits clearer, distinct separations, even for similar digits. The semantic structure injected by the SP constraint leads to more robust and meaningful representations, helping the model better differentiate between challenging classes. This highlights the effectiveness of the SP constraint in refining latent space organization.



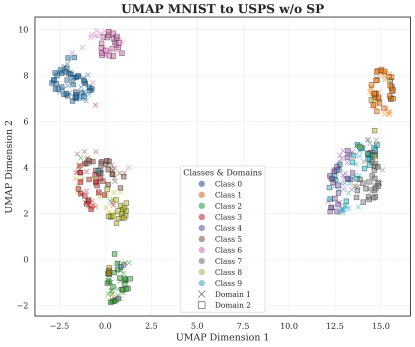
(a) MNIST to USPS



(b) USPS to MNIST (LSDA)



(c) UMAP with SP



(d) UMAP without SP

Figure 6: (a) MNIST to USPS (LSDA). (b) USPS to MNIST (LSDA). (c) UMAP with SP. (d) UMAP without SP. All labeled data is randomly selected from the source dataset and tested on the target dataset, with results averaged over 10 trials. Both (a) and (b) demonstrate that with SP loss, the model is more robust to limited data. This is further supported by the corresponding UMAP visualizations, where (c) shows larger separation between classes compared to (d), reflecting better class distinction.

E More Detailed Discussion of Gradient Comparison Between LSGM and SFS Trick

Below, we detail the encoder and decoder optimization objectives for LSGM:

$$\min_{\theta,\varphi} \mathbb{E}_{q_\theta(z_0|x)}\left[-\log p_\varphi(x|z_0)\right] + \mathbb{E}_{q_\theta(z_0|x)}\left[\log q_\theta(z_0|x)\right] + \mathbb{E}_{t,\epsilon,q(z_t|z_0),q_\theta(z_0|x)}\left[\frac{w(t)}{2}\|\epsilon - \epsilon_\psi(z_t,t)\|_2^2\right],$$

where $w(t)$ is a weighting function,$\epsilon_\psi(\cdot)$ represents a diffusion model, and $\epsilon \sim \mathcal{N}(0, I)$. Similar to our loss objective (refer to Eqn. 10), LSGM substitutes the traditional cross-entropy term with a learnable neural network prior. Specifically, the final term in the Evidence Lower Bound (ELBO) is replaced with a weighted denoising score matching objective.

We first adapt notations used in our objective for easy readability during comparison. Diffusion model can approximate the denoising score function by rewritting $\epsilon_\psi(z_t, t) =$

$\sigma_t S_\psi(z + \sigma_t\epsilon, \sigma_t)$ Song et al. (2022). To streamline discussion and avoid repetition, we will refer to the final term of this formulation as the **LSGM objective**, we can write the cross-entropy term of LSGM as below with the weighting function as $w(t) = g(t)^2/{\sigma_t}^2$ which maximizes the likelihood between the encoder posterior and the prior where $g(\cdot)$ is the diffusion coefficient typically proportional to the variance scheduling function Song et al. (2021b)Vahdat et al. (2021).

$$\mathcal{L}_{\text{LSGM}} = \mathbb{E}_{q_{\sigma_t}(\tilde{z}|z), q_\theta(z|x)} \left[\frac{w(t)}{2} \|\epsilon - \epsilon_\psi(z_t, t)\|_2^2 \right] \tag{23}$$

$$= \mathbb{E}_{q_{\sigma_t}(\tilde{z}|z), q_\theta(z|x)} \left[\frac{g(t)^2}{2} \left\| \frac{\epsilon}{\sigma_t} - S_\psi(\tilde{z} = z + \sigma_t\epsilon, \sigma_t) \right\|_2^2 \right] \tag{24}$$

During encoder updates, the gradient computation for the last term with respect to the encoder parameters is expressed as:

$$\nabla_\theta L_{\text{LSGM}} = \mathbb{E}_{q_{\sigma_t}(\tilde{z}|z), q_\theta(z|x)} \left[g(t)^2 \left(\frac{\epsilon}{\sigma_t} - S_\psi(\tilde{z}, \sigma_t) \right)^\top \frac{\partial S_\psi(\tilde{z}, \sigma_t)}{\partial \tilde{z}} \frac{\partial \tilde{z}}{\partial \theta} \right]. \tag{25}$$

This framework requires computing the Jacobian term $\frac{\partial S_\psi(\tilde{z}, \sigma_t)}{\partial z_t}$, which is both computationally expensive and memory-intensive. To mitigate this, the Score Function Substitution (SFS) trick eliminates the need for Jacobian computation by detaching the latent input z^* in the score function from the encoder parameters. The resulting gradient is expressed as:

$$\nabla_\theta L_{\text{SFS}} = -\mathbb{E}_{z \sim q_\theta(z|x,d)} \left[\frac{\partial z}{\partial \theta}^\top \left(S_\psi(z^*, \sigma \approx 0) \Big|_{z^*=z} \right) \right]. \tag{26}$$

This modification provides significant advantages, reducing memory usage by bypassing the computational graph of the diffusion model's U-NET and enhancing stability. Poole et al. (2022) highlighted that the Jacobian computation approximates the Hessian of the dataset distribution, which is particularly unstable at low noise levels. Our empirical results in Fig. 1 confirm these findings, demonstrating improved stability with our loss objective compared to LSGM.

E.1 Proof: SFS trick is proportional to a distilled LSGM loss

To demonstrate that applying the Sticking-the-Landing principle Roeder et al. (2017) to LSGM yields the SFS trick, we begin by expressing Eqn. 25 in its score function form:

$$\nabla_\theta L_{\text{LSGM}} = g(t)^2 \mathbb{E}_{q_{\sigma_t}(\tilde{z}|z), q_\theta(z|x)} \left[\left(\frac{\epsilon}{\sigma_t} - \nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z) \right)^\top \frac{\partial (\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z))}{\partial \tilde{z}} \frac{\partial \tilde{z}}{\partial \theta} \right]. \tag{27}$$

For clarity, we decompose Eqn. 27 into three components:

- $A = \left(\frac{\epsilon}{\sigma_t} - \nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z) \right)^\top$
- $B = \frac{\partial (\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z))}{\partial \tilde{z}}$
- $C = \frac{\partial \tilde{z}}{\partial \theta}$

We first compute the expectation $\mathbb{E}[A^\top BC]$:

$$\mathbb{E}[A^\top BC] = \mathbb{E}\left[\frac{\epsilon}{\sigma_t}^\top BC\right] - \mathbb{E}\left[\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z)^\top BC\right] \tag{28}$$

$$= \mathbb{E}\left[\frac{\epsilon}{\sigma_t}^\top\right] \mathbb{E}[BC] - \mathbb{E}\left[\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z)^\top \frac{\partial(\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z))}{\partial \tilde{z}} C\right] \tag{29}$$

$$= -\mathbb{E}_{q_\theta(z|x)}\left[\mathbb{E}_{q_{\sigma_t}(\tilde{z}|z)}\left[\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z)^\top \frac{\partial(\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z))}{\partial \tilde{z}}\right] C\right]. \tag{30}$$

Here, Eqn. 28 follows from substituting the definition of A, Eqn. 29 separates the expectation terms because $\frac{\epsilon}{\sigma_t}^\top$ is independent of BC, and Eqn. 30 eliminates the first term since $\mathbb{E}\left[\frac{\epsilon}{\sigma_t}^\top\right] = 0$.

Next, we evaluate the expectation $\mathbb{E}[A^\top C]$:

$$\mathbb{E}[A^\top C] = \mathbb{E}\left[\left(\frac{\epsilon}{\sigma_t} - \nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z)\right)^\top C\right] \tag{31}$$

$$= -\mathbb{E}_{q_\theta(z|x)}\left[\mathbb{E}_{q_{\sigma_t}(\tilde{z}|z)}\left[\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z)^\top\right] C\right]. \tag{32}$$

In Eqn. 32, the term $\frac{\epsilon}{\sigma_t}$ is removed due to its independence from C and its zero expectation. Now, we assume the score model perfectly predicts the noisy latent representation of the encoder, i.e., $q_{\sigma_t}(\tilde{z}|z) = p_\psi(\tilde{z}|z)$, to compute $\mathbb{E}[A^\top BC]$ and $\mathbb{E}[A^\top C]$.

For $\mathbb{E}[A^\top BC]$, considering only the inner expectation, we note that $p_\psi(\tilde{z}|z)$ is conditonal Gaussian:

$$\mathbb{E}_{q_{\sigma_t}(\tilde{z}|z)}\left[\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z)^\top \frac{\partial(\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z))}{\partial \tilde{z}}\right] \tag{33}$$

$$= \mathbb{E}_{q_{\sigma_t}(\tilde{z}|z)}\left[\left(\frac{\tilde{z} - z}{\sigma^2}\right)^\top \nabla_{\tilde{z}} \left(\frac{\tilde{z} - z}{\sigma^2}\right)\right] \tag{34}$$

$$= \mathbb{E}_{p_\psi(\tilde{z}|z)}\left[\left(\frac{\tilde{z} - z}{\sigma^2}\right)^\top \left(\frac{1}{\sigma^2}\right)\right] \tag{35}$$

$$= 0. \tag{36}$$

In Eqn. 35, the substitution $q_{\sigma_t}(\tilde{z}|z) = p_\psi(\tilde{z}|z)$ simplifies the expectation to zero. Similarly, for the inner expectation of $\mathbb{E}[A^\top C]$:

$$\mathbb{E}_{q_{\sigma_t}(\tilde{z}|z)}\left[\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z)\right] = \mathbb{E}_{p_\psi(\tilde{z}|z)}\left[\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z)\right] \tag{37}$$

$$= \int p_\psi(\tilde{z}|z) \frac{\nabla_{\tilde{z}} p_\psi(\tilde{z}|z)}{p_\psi(\tilde{z}|z)} d\tilde{z} \tag{38}$$

$$= \nabla_{\tilde{z}} \int p_\psi(\tilde{z}|z) d\tilde{z} \tag{39}$$

$$= 0. \tag{40}$$

Thus, when the score model accurately predicts the posterior score function, removing the B term from $\mathbb{E}[A^\top BC]$ introduces no gradient bias. Consequently, applying the Sticking-the-Landing methodology eliminates the Hessian term, reducing variance. The result is:

$$\nabla_\theta \mathcal{L}_{\text{Distilled-LSGM}} = g(t)^2 \mathbb{E}_{q_{\sigma_t}(\tilde{z}|z), q_\theta(z|x)} \left[\left(\frac{\epsilon}{\sigma_t} - \nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z) \right)^\top \frac{\partial \tilde{z}}{\partial \theta} \right]. \tag{41}$$

Sticking-the-Landing can be applied once more to Eqn. 41 by removing $\frac{\epsilon}{\sigma_t}$ and constraining on small noise levels ($\sigma_t \approx 0$), the gradient becomes proportional to that of the SFS trick:

$$\nabla_\theta L_{\text{SFS}} = -\mathbb{E}_{z \sim q_\theta(z|x,d)} \left[\frac{\partial z}{\partial \theta}^\top \left(S_\psi(z^*, \sigma \approx 0)\Big|_{z^*=z} \right) \right] \propto -g(t)^2 \mathbb{E}_{q_{\sigma_t}(\tilde{z}|z), q_\theta(z|x)} \left[(\nabla_{\tilde{z}} \log p_\psi(\tilde{z}|z))^\top \frac{\partial \tilde{z}}{\partial \theta} \right]. \tag{42}$$

□

F Choices of different metric spaces in different dataset

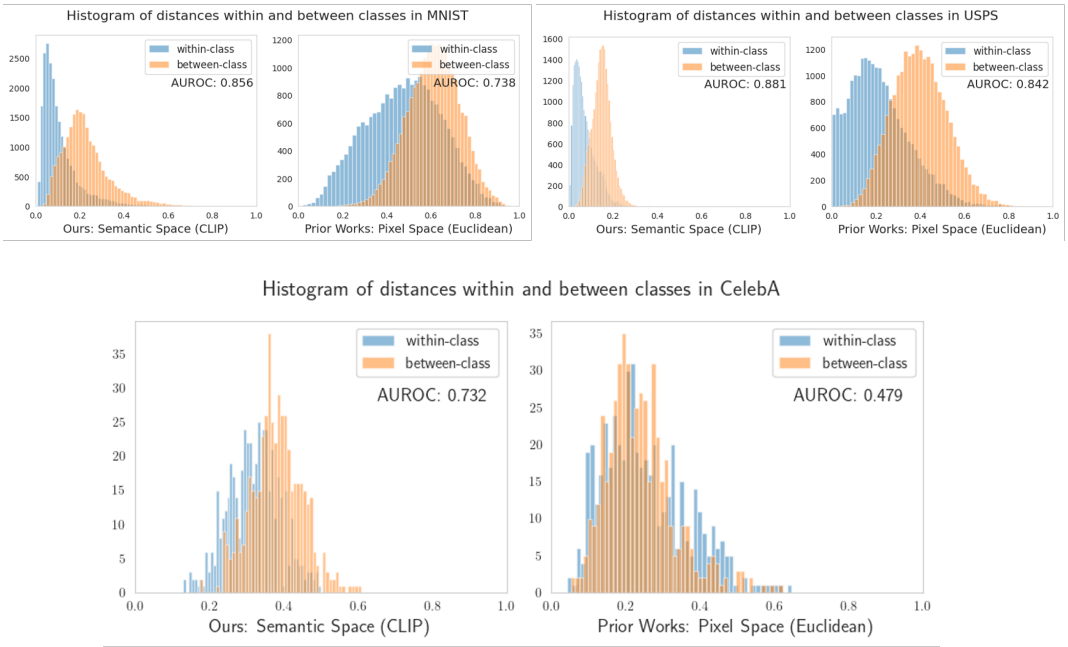


Figure 7: Histogram of the pairwise distance between data samples within a class and between different classes for three datasets: MNIST, USPS, and CelebA. The amount of separation of two histogram is computed by using the AUROC score which being measured by a binary classifier to distinguish between **with-in** class results and **between-class** results. The class considered in MNIST and USPS is the digits, and in CelebA is hair color.

From the graph, we observe that for the MNIST and USPS datasets, both the Euclidean pixel space metric and the semantic space metric can effectively separate data pairs into within-class or between-class categories. However, the semantic space metric demonstrates a higher AUROC separation score, indicating that it provides a more reliable metric for distinguishing between these pair types.

In contrast, for the CelebA dataset, relying solely on pixel-based Euclidean distances struggles to differentiate whether the paired distances belong to within-class or between-class data

pairs. By employing a semantic metric, such as the one derived from CLIP, a clear distinction emerges, underscoring its utility.

These observations highlight that while pixel space metrics like Euclidean distance may be useful for certain datasets, semantic distance metrics, when available, often offer superior performance and may even be essential for datasets with more complex structures or features.

G Multi-Domain Distribution Matching Setting

We train SAUB with SP on three different MNIST rotation angles: $0°, 30°, 60°$. The top row is the ground truth image, the second row is the reconstruction, the third row is translation to MNIST $30°$, and last row is translation to MNIST $60°$ in Fig. 8. Qualitatively most of the stylistic and semantic features are preserved with the correct rotation.

H Detailed Architecture of the model

H.1 Fairness Representation Learning

The encoder is a 3-layer MLP with hidden dimension 64, and latent dimension 8 with ReLU layers connecting in between. The classifier is a 3-layer MLP with hidden dimension 64 with ReLU layers connecting in between.

H.2 Separation Metric for Synthetic Dataset

The classifier is trained by a support vector where hyperparmeters are chosen from the list 'C': [0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001] with 5-fold cross validation. Error plot is generated from 5 runs.

H.3 Domain Adaptation VAE Model

Encoder Architecture

The encoder compresses the input image $\mathbf{x} \in \mathbb{R}^{1\times28\times28}$ into a latent representation. The architecture consists of the following layers:

- **Conv2D:** 4×4, stride 2, 16 channels (input size $28 \times 28 \to 14 \times 14$).
- **Residual Block:** 16 channels.
- **Conv2D:** 4×4, stride 2, 64 channels (input size $14 \times 14 \to 7 \times 7$).
- **Residual Block:** 64 channels.
- **Conv2D:** 3×3, stride 2, $2 \times$ latent size channels (input size $7 \times 7 \to 4 \times 4$).
- **Residual Block:** $2 \times$ latent size channels.
- **Conv2D:** 4×4, stride 1, $2 \times$ latent size channels (output size $4 \times 4 \to 1 \times 1$).
- Split into two branches for μ and $\log \sigma^2$, each with latent size channels.

Decoder Architecture

The decoder reconstructs the input image $\mathbf{x}' \in \mathbb{R}^{1\times28\times28}$ from the latent representation. The architecture consists of the following layers:

- **Reshape:** Latent vector reshaped to size $(\text{latent size}, 1, 1)$.
- **Residual Block:** latent size channels.
- **ConvTranspose2D:** 4×4, stride 1, 64 channels (output size $1 \times 1 \to 4 \times 4$).
- **Residual Block:** 64 channels.
- **ConvTranspose2D:** 4×4, stride 2, 16 channels (output size $4 \times 4 \to 8 \times 8$).
- **Residual Block:** 16 channels.
- **ConvTranspose2D:** 4×4, stride 4, 1 channel (output size $8 \times 8 \to 28 \times 28$).

H.4 Domain Translation VAE Model

Encoder Architecture

The encoder compresses the input image $\mathbf{x} \in \mathbb{R}^{3\times64\times64}$ into a latent representation. The architecture consists of the following layers:

- **Conv2D:** 3×3, stride 2, 64 channels (input size $64 \times 64 \to 32 \times 32$).
- **Residual Block:** 64 channels.
- **Conv2D:** 3×3, stride 2, 128 channels (input size $32 \times 32 \to 16 \times 16$).
- **Residual Block:** 128 channels.
- **Conv2D:** 3×3, stride 2, 256 channels (input size $16 \times 16 \to 8 \times 8$).
- **Residual Block:** 256 channels.
- **Conv2D:** 3×3, stride 2, $2 \times$ latent size channels (input size $8 \times 8 \to 4 \times 4$).
- **Residual Block:** $2 \times$ latent size channels.
- Split into two branches for μ and $\log \sigma^2$, each with latent size channels.

Decoder Architecture

The decoder reconstructs the input image $\mathbf{x}' \in \mathbb{R}^{3\times64\times64}$ from the latent representation. The architecture consists of the following layers:

- **Reshape:** Latent vector reshaped to size $(\text{latent size}, 4, 4)$.
- **Residual Block:** latent size channels.
- **ConvTranspose2D:** 3×3, stride 2, 256 channels (output size $4 \times 4 \to 8 \times 8$).
- **Residual Block:** 256 channels.
- **ConvTranspose2D:** 3×3, stride 2, 128 channels (output size $8 \times 8 \to 16 \times 16$).
- **Residual Block:** 128 channels.
- **ConvTranspose2D:** 3×3, stride 2, 64 channels (output size $16 \times 16 \to 32 \times 32$).
- **Residual Block:** 64 channels.
- **ConvTranspose2D:** 3×3, stride 2, 3 channels (output size $32 \times 32 \to 64 \times 64$).
- **Sigmoid Activation:** To map outputs to the range $[0, 1]$.

H.5 Domain Adaptation Classifier

Classifier consists of 2 linear layers and a ReLU activation function.

H.6 Pretrained CLIP

For this work, we utilized pretrained CLIP models from the OpenCLIP repository. Specifically:

- **ViT-H-14-378-quickgelu on dfn5b dataset** was employed for training the GW-SP regularizer.
- **ViT-L-14-quickgelu on dfn2b dataset** was used for evaluation on the Image Retrieval task.

I More Synthetic Dataset Results

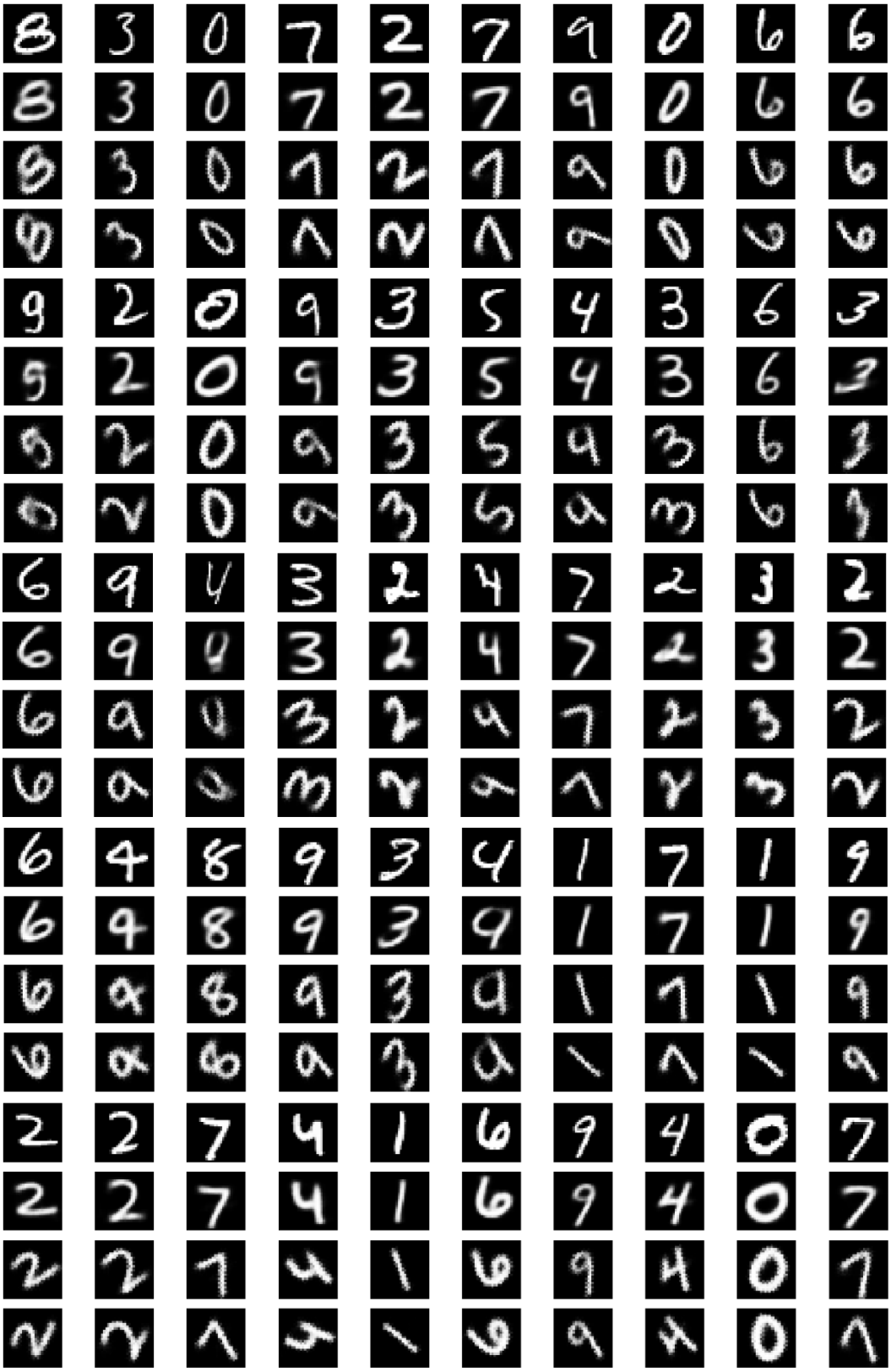


Figure 8: Multi-domain adaptation: MNIST images rotated at various angles.

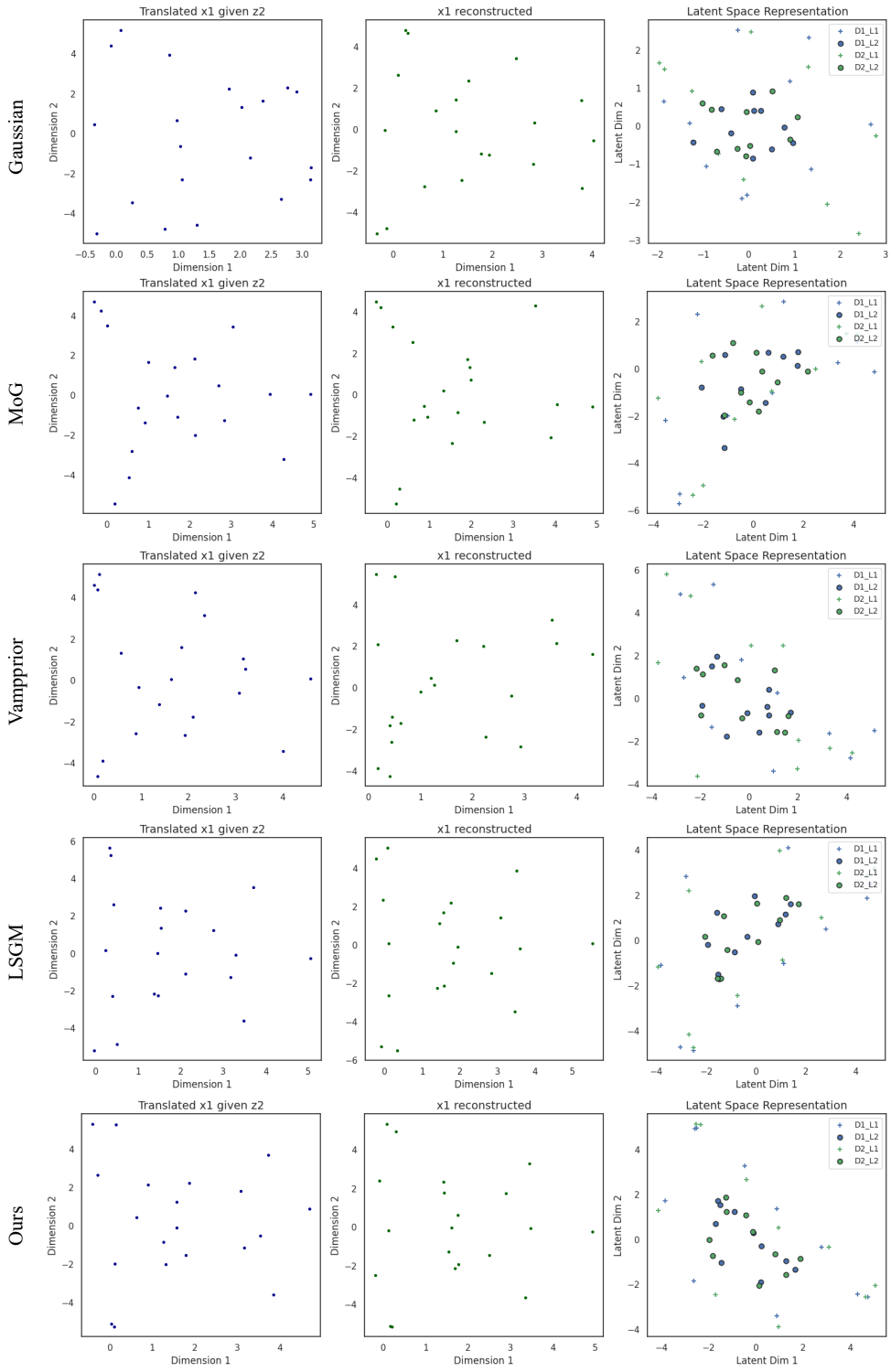


Figure 9: This figures show the translated dataset, reconstructed dataset, as well as the latent space under sample size 20.

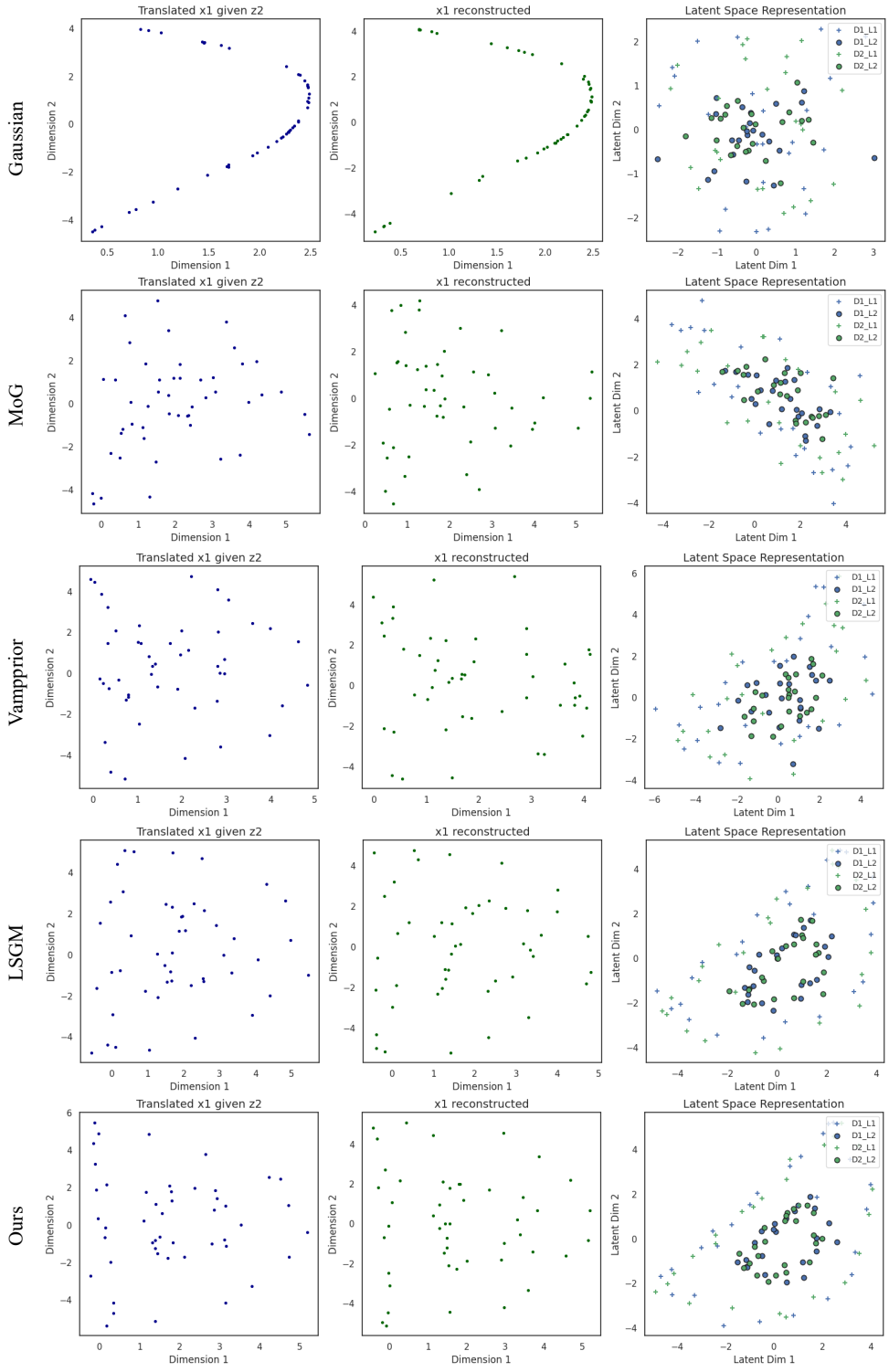


Figure 10: This figures show the translated dataset, reconstructed dataset, as well as the latent space under sample size 50.

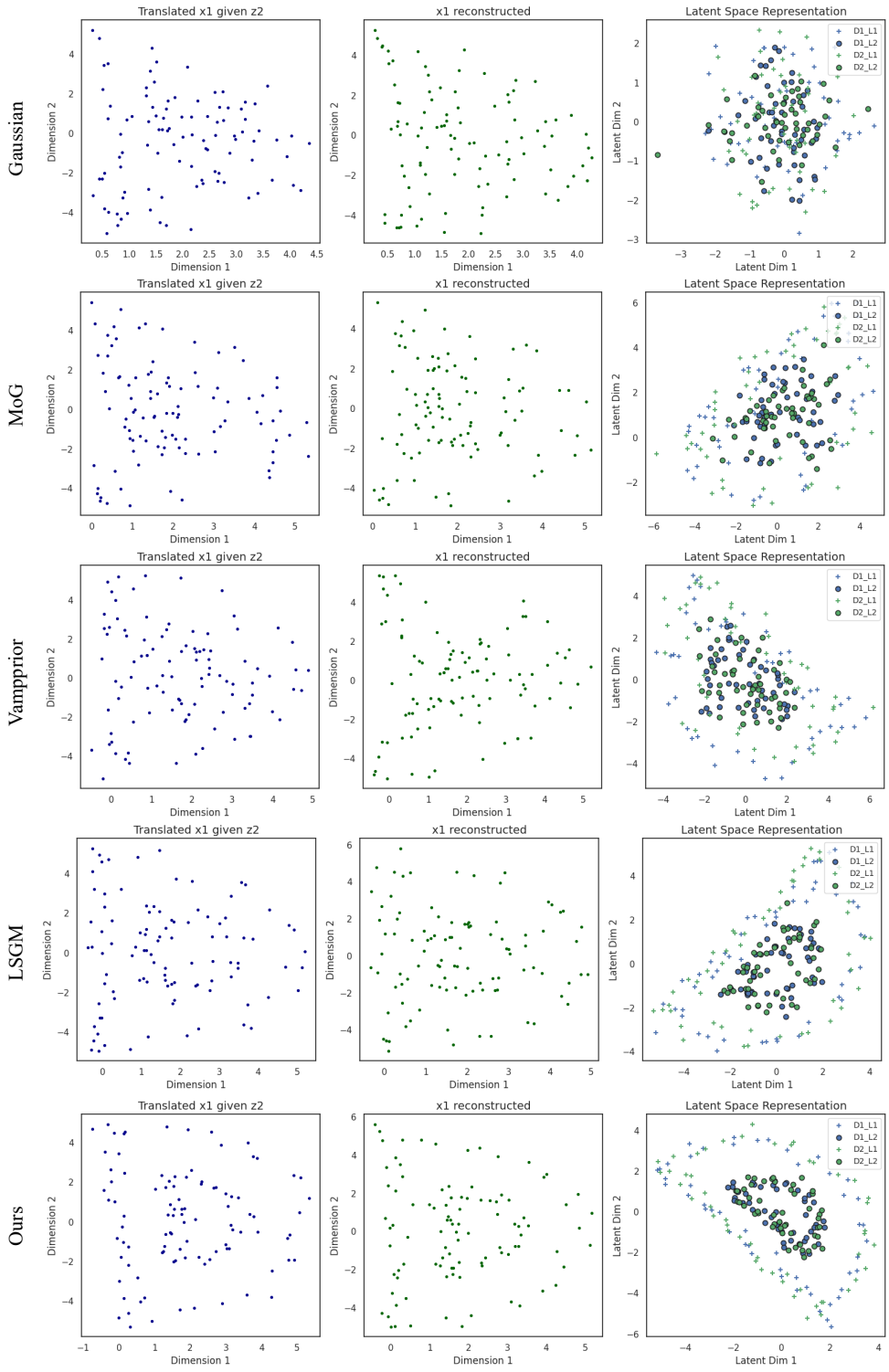


Figure 11: This figures show the translated dataset, reconstructed dataset, as well as the latent space under sample size 100.

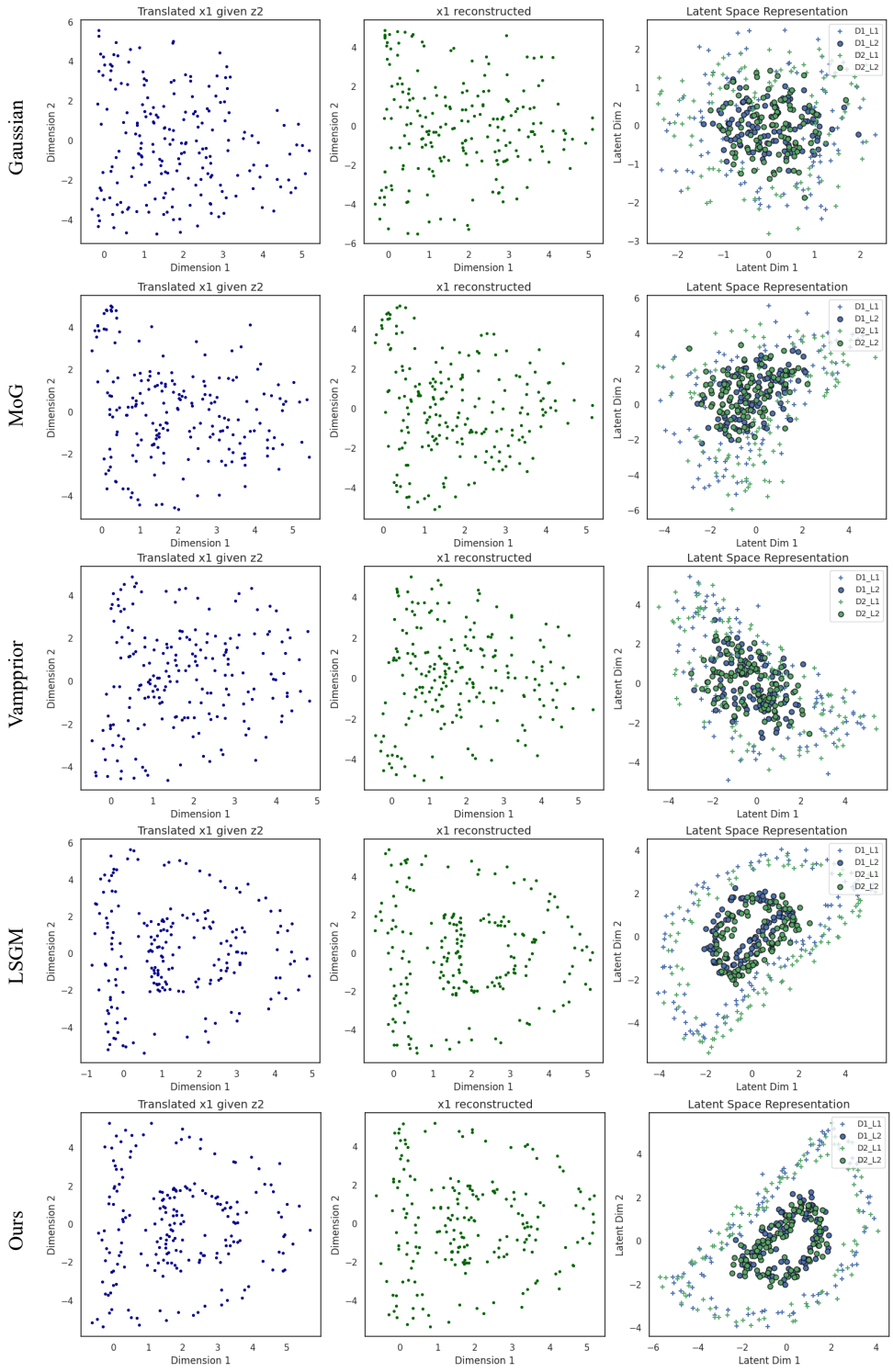


Figure 12: This figures show the translated dataset, reconstructed dataset, as well as the latent space under sample size 200.

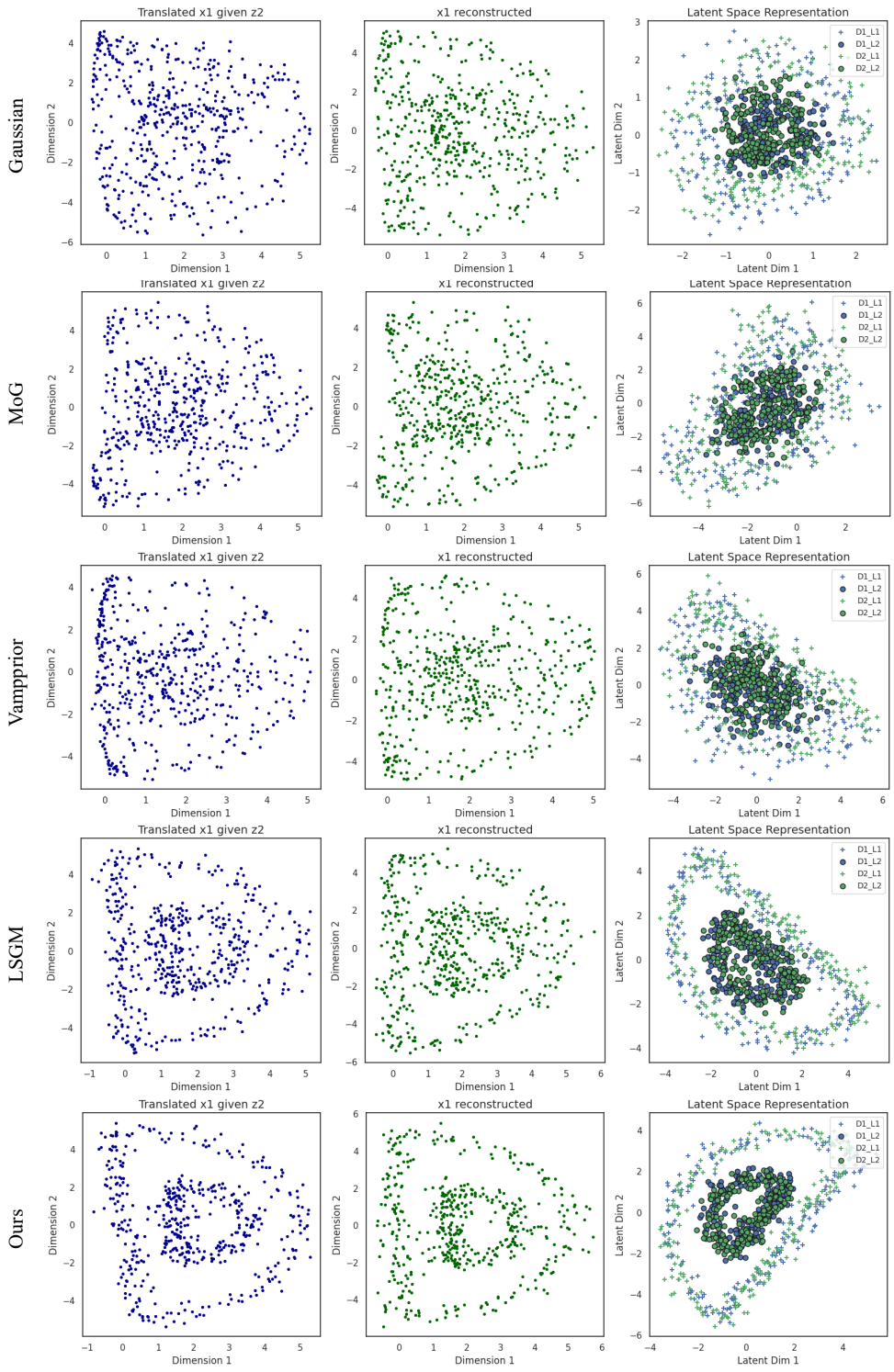


Figure 13: This figures show the translated dataset, reconstructed dataset, as well as the latent space under sample size 500.

J Image translation between MNIST and USPS

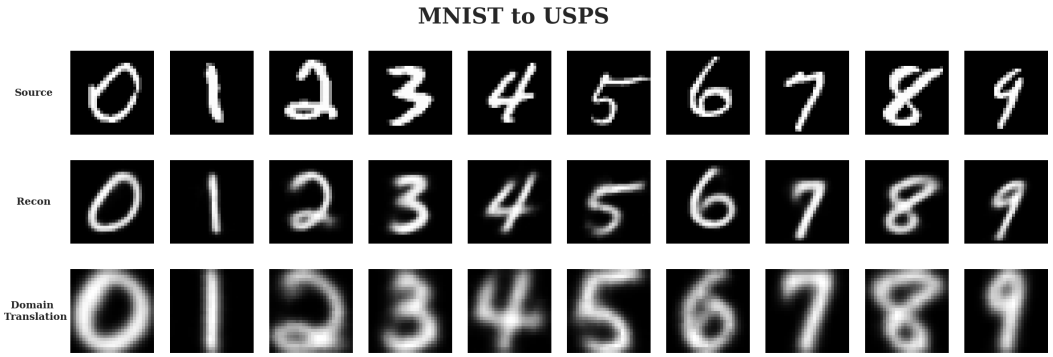
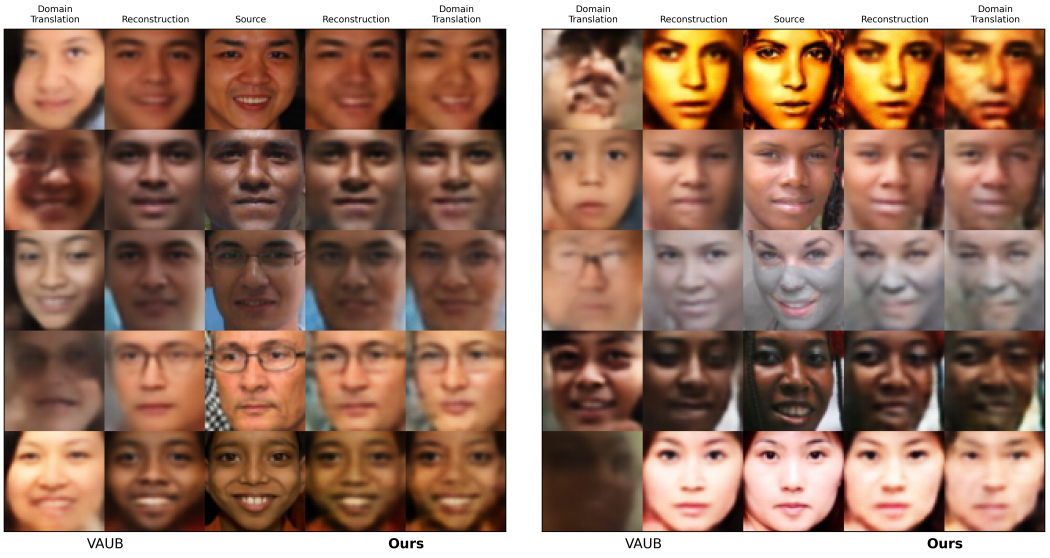


Figure 14: MNIST to USPS translated image trained with SP.

K FairFace Image Translation

This experimental setting is conducted in a fully unsupervised manner without SP loss. We compare our proposed score-based prior (SAUB) with a multi-Gaussian-based learning prior (VAUB) to evaluate their effectiveness.

K.1 Handpicked samples



(a) Male to Female translation Female to Male translation

Figure 15: In this experiment, both models are trained in an unsupervised manner (i.e., SAUB is trained without GW-SP loss). SAUB clearly exhibits superior semantic preservation in both (a) and (b), particularly with respect to features such as skin color, race, and age. Notably, SAUB makes minimal adjustments when altering gender, while VAUB struggles to retain the identity of the original data. (These samples are handpicked to illustrate the trend.)

K.2 Random Samples

In Fig. 16, we show completely random samples from the FairFace dataset.

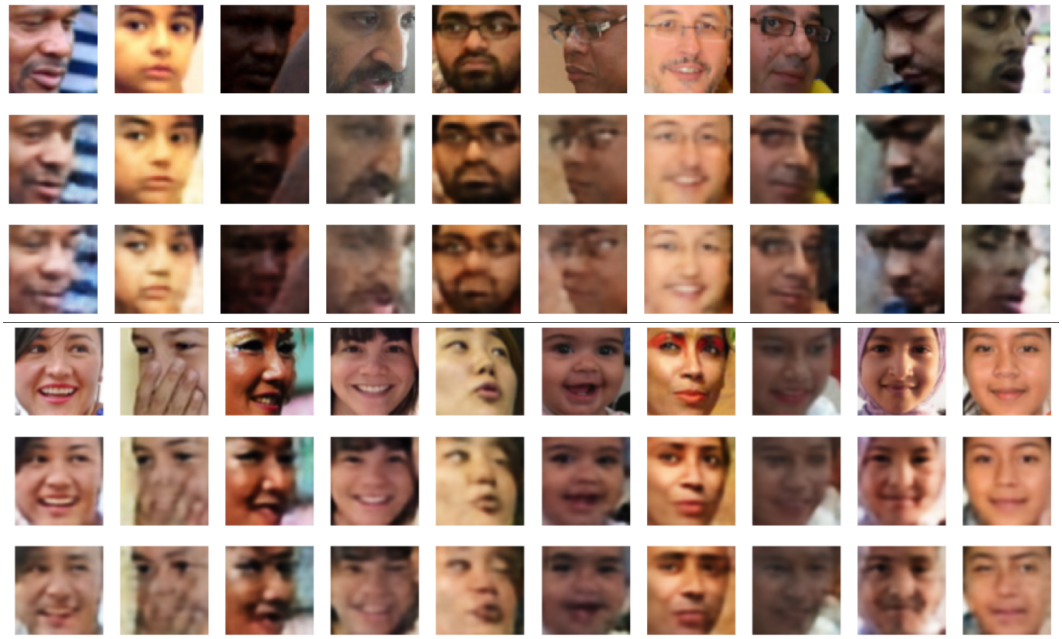


Figure 16: Random samples from the FairFace experiment using our method. Top three rows translate from male to female and the bottom three rows translate from female to male. First row is original, second is reconstructed, and third is translated.

L Additional Random Image Translations on CelebA

Examples of random image translations between black hair and blonde hair are presented in Fig. 17 and Fig. 18

Figure 17: Random Samples from Black to Blonde Hair Female

Figure 18: Random Samples from Blonde to Black Hair Female