Towards Practical Non-Adversarial Distribution Matching

Ziyu Gong
Purdue University
gong123@purdue.edu

Ben Usman
Google Research
usmn@google.com

Han Zhao
University of Illinois Urbana-Champaign
hanzhao@illinois.edu

David I. Inouye
Purdue University
dinouye@purdue.edu

Abstract

Distribution matching can be used to learn invariant representations with applications in fairness and robustness. Most prior works resort to adversarial matching methods but the resulting minimax problems are unstable and challenging to optimize. Non-adversarial likelihood-based approaches either require model invertibility, impose constraints on the latent prior, or lack a generic framework for distribution matching. To overcome these limitations, we propose a non-adversarial VAE-based matching method that can be applied to any model pipeline. We develop a set of alignment upper bounds for distribution matching (including a noisy bound) that have VAE-like objectives but with a different perspective. We carefully compare our method to prior VAE-based matching approaches both theoretically and empirically. Finally, we demonstrate that our novel matching losses can replace adversarial losses in standard invariant representation learning pipelines without modifying the original architectures—thereby significantly broadening the applicability of non-adversarial matching methods.

1 INTRODUCTION

Distribution matching can be used to learn invariant representations that have many applications in robustness, fairness, and causality. For example, in Domain Adversarial Neural Networks (DANN) (Ganin et al., 2016; Zhao et al., 2018), the key is enforcing an intermediate latent space to be invariant with respect to the domain. In fair representation learning (e.g., Creager et al. (2019); Louizos et al. (2015); Xu et al. (2018); Zhao et al. (2020)), a common approach is to enforce that a latent representation is invariant with respect to a sensitive attribute. In both of these cases, distribution matching is formulated as a (soft) constraint or regularization on the overall problem that is motivated by the context (either domain adaptation or fairness constraints). Thus, there is an ever-increasing need for reliable distribution matching methods.

Most prior works of distribution matching resort to adversarial training to implement the required matching constraints. While adversarial loss terms are easy to implement as they only require a discriminator network, the corresponding minimax optimization problems are unstable and difficult to optimize in practice (see e.g. Lucic et al. (2018); Kurach et al. (2019); Farnia and Ozdaglar (2020); Nie and Patel (2020); Wu et al. (2020); Han et al. (2023)) in part because of the competitive nature of the min-max optimization problem. To reduce the dependence on adversarial learning, Grover et al. (2020) proposed an invertible flow-based method to combine likelihood and adversarial losses under a common framework. Usman et al. (2020) proposed a completely non-adversarial matching method using invertible flow-based models where one distribution is assumed to be fixed. Cho et al. (2022) unified these previous non-adversarial flow-based approaches for distribution matching by proving that they are upper bounds of the Jensen-Shannon divergence called the alignment upper bound (AUB). However, these non-adversarial methods require invertible model pipelines, which significantly limit their applicability in key distribution matching applications. For example, because invertibility is required, the aligner cannot reduce the

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

dimensionality. As another consequence, it is impossible to use a shared invertible aligner because JSD is invariant under invertible transformations[1]. Most importantly, invertible architectures are difficult to design and optimize compared to general neural networks. Finally, several fairness-oriented works have proposed variational upper bounds (Louizos et al., 2015; Moyer et al., 2018; Gupta et al., 2021) for distribution matching via the perspective of mutual information. Some terms in these bounds can be seen as special cases of our proposed bounds (detailed in Section 5; also see comparisons to these methods in subsection 6.2). However, these prior works impose a fixed prior distribution and are focused on the fairness application, i.e., they do not explore the broader applicability of their matching bounds beyond fairness.

To address these issues, we propose a VAE-based matching method that can be applied to any model pipeline, i.e., it is model-agnostic like adversarial methods, but it is also non-adversarial, i.e., it forms a min-min cooperative problem instead of a min-max problem. Inspired by the flow-based alignment upper bound (AUB)(Cho et al., 2022), we relax the invertibility of AUB by replacing the flow with a VAE and add a mutual information term that simplifies to a β-VAE (Higgins et al., 2017) matching formulation. From another perspective, our development can be seen as revisiting VAE-based matching bounds where we show that prior works impose an unnecessary constraint caused by using a fixed prior and do not encourage information preservation as our proposed relaxation of AUB. We then prove novel noisy alignment upper bounds as summarized in Table 1, which may help avoid vanishing gradient and local minimum issues that may exist when using the standard JSD as a divergence measure (Arjovsky et al., 2017).[2] This JSD perspective to distribution matching complements and enhances the mutual information perspective in prior works (Moyer et al., 2018; Gupta et al., 2021) because the well-known equivalence between mutual information of the observed features and the domain label and the JSD between the domain distributions. Our contributions can be summarized as follows:

- We relax the invertibility constraint of AUB using VAEs and a mutual information term while ensuring the distribution matching loss is an upper bound of the JSD up to a constant.
- We propose noisy JSD and noise-smoothed JSD to help avoid vanishing gradients and local minima during optimization, and we develop the corresponding noisy alignment upper bounds.
- We demonstrate that our non-adversarial VAE-based matching losses can replace adversarial losses without any change to the original model's architecture. Thus, they can be used within any standard invariant representation learning pipeline such as domain-adversarial neural networks or fair representation learning without modifying the original architectures.

[1] refer to subsection E.1 for proof

[2] while the term *alignment* in deep learning is often being interpreted as "bringing human values and goals into models", for the purposes of maintaining consistency with the terminology employed in the AUB paper, we instead use the term *alignment* interchangeably with the idea of *distribution matching* throughout the rest of the paper.

Notation. Let $\boldsymbol{x}$ and $d \in \{1, 2, \ldots, k\}$ denote the observed variable and the domain label, respectively, where $k > 1$ is the number of domains. Let $\boldsymbol{z} = g(\boldsymbol{x}|d)$ denote a deterministic representation function, i.e., an *aligner*, that is invertible w.r.t. $\boldsymbol{x}$ conditioned on d being known. Let q denote the *encoder* distribution: $q(\boldsymbol{x}, d, \boldsymbol{z}) = q(\boldsymbol{x}, d)q(\boldsymbol{z}|\boldsymbol{x}, d)$, where $q(\boldsymbol{x}, d)$ is the true data distribution and $q(\boldsymbol{z}|\boldsymbol{x}, d)$ is the encoder (i.e., probabilistic aligner). Similarly, let p denote the *decoder* distribution $p(\boldsymbol{x}, d, \boldsymbol{z}) = p(\boldsymbol{z})p(d)p(\boldsymbol{x}|\boldsymbol{z}, d)$, where $p(\boldsymbol{z})$ is the shared prior, $p(\boldsymbol{x}|\boldsymbol{z}, d)$ is the decoder, and $p(d) = q(d)$ is the marginal distribution of the domain labels. Entropy and cross entropy will be denoted as $\mathrm{H}(\cdot)$, and $\mathrm{H_c}(\cdot, \cdot)$, respectively, where the KL divergence is denoted as $\mathrm{KL}(p, q) = \mathrm{H_c}(p, q) - \mathrm{H}(p)$. Jensen-Shannon Divergence (JSD) is denoted as $\mathrm{JSD}(p, q) = \mathrm{H}(\frac{1}{2}(p+q)) - \frac{1}{2}(\mathrm{H}(p)+\mathrm{H}(q))$. Furthermore, the Generalized JSD (GJSD) extends JSD to multiple distributions (Lin, 1991) and is equivalent to the mutual information between $\boldsymbol{z}$ and d: $\mathrm{GJSD}(\{q(\boldsymbol{z}|d)\}_{d=1}^k) \equiv I(\boldsymbol{z}, d) = \mathrm{H}(\mathbb{E}_{q(d)}[q(\boldsymbol{z}|d)]) - \mathbb{E}_{q(d)}[\mathrm{H}(q(\boldsymbol{z}|d))]$, where $q(d)$ are the weights for each domain distribution $q(\boldsymbol{z}|d)$. JSD is recovered if there are two domains and $q(d) = \frac{1}{2}, \forall d$.

2 BACKGROUND

Adversarial Methods. Adversarial methods based on GANs (Goodfellow et al., 2014) maximize a *lower bound* on the GJSD using a probabilistic classifier denoted by f:

$$\min_g \left(\max_f \mathbb{E}_{q(\boldsymbol{x},d)}[\log f_d(g(\boldsymbol{x}|d))] \right) = \min_g \mathrm{ADV}(g) \leq \min_g \mathrm{GJSD}(\{q(z|d)\}_{d=1}^k).$$

where $\mathrm{ADV}(g)$ is the adversarial loss that lower bounds the GJSD and can be made tight if f is optimized overall all possible classifiers. This optimization can be difficult to optimize in practice due to its adversarial formulation and vanishing gradients caused by JSD (see e.g. Lucic et al. (2018); Kurach et al. (2019); Farnia and Ozdaglar (2020); Nie and Patel (2020); Wu et al. (2020); Han et al. (2023); Arjovsky and Bottou (2017)).

Table 1: Summary of new alignment upper bounds where $C \triangleq -\mathbb{E}_{q(d)}[\mathrm{H}(q(\boldsymbol{x}|d))]$ and Flow+JSD is prior work from Cho et al. (2022). Blue represents changes needed for VAEs, i.e., replacing log Jacobian term and adding β term to encourage near-invertibility of encoder, and red highlights that we merely need to add noise to the latent representation before evaluating the shared latent distribution. JSD bounds can be made tight via optimization while noisy JSD bounds can only be made tight if the noise variance goes to 0.

Model	JSD	Noisy JSD
Flow $\boldsymbol{z} = g(\boldsymbol{x}\|d)$	$\min_{p(\boldsymbol{z})} \mathbb{E}_q[-\log(\lvert J_g(\boldsymbol{x}\|d)\rvert \cdot p(\boldsymbol{z}))] + C$	$\min_{p(\tilde{\boldsymbol{z}})} \mathbb{E}_q[-\log(\lvert J_g(\boldsymbol{x}\|d)\rvert \cdot p(\boldsymbol{z}+\epsilon))] + C$
β-VAE $(\beta \le 1)$ $\boldsymbol{z} \sim q(\boldsymbol{z}\|\boldsymbol{x}, d)$	$\min_{\substack{p(\boldsymbol{z}) \\ p(\boldsymbol{x}\|\boldsymbol{z},d)}} \mathbb{E}_q\left[-\log\left(\frac{p(\boldsymbol{x}\|\boldsymbol{z},d)}{q(\boldsymbol{z}\|\boldsymbol{x},d)^\beta} \cdot p(\boldsymbol{z})^\beta\right)\right] + C$	$\min_{\substack{p(\tilde{\boldsymbol{z}}) \\ p(\boldsymbol{x}\|\boldsymbol{z},d)}} \mathbb{E}_q\left[-\log\left(\frac{p(\boldsymbol{x}\|\boldsymbol{z},d)}{q(\boldsymbol{z}\|\boldsymbol{x},d)^\beta} \cdot p(\boldsymbol{z}+\epsilon)^\beta\right)\right] + C$

We aim to address both optimization issues by forming a min-min problem (Section 3) and considering additive noise to avoid vanishing gradients (Section 4).

Fair VAE Methods. A series of prior works in fairness implemented distribution matching methods based on VAEs (Kingma et al., 2019), where the prior is assumed to be the standard normal distribution $\mathcal{N}(0, I)$ and the probabilistic encoder represents a stochastic aligner. Concretely, the fair VAE objective (Louizos et al., 2015) can be viewed as an *upper bound* on the GJSD:

$$\min_{q(\boldsymbol{z}|\boldsymbol{x},d)} \Big(\min_{p(\boldsymbol{x}|\boldsymbol{z},d)} \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z},d)} \big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)} \cdot p_{\mathcal{N}(0,I)}(\boldsymbol{z}) \big] \Big) \\ \ge \min_{q(\boldsymbol{z}|\boldsymbol{x},d)} \mathrm{GJSD}(\{q(z|d)\}_{d=1}^k).$$

We revisit and compare to this and other VAE-based methods (Gupta et al., 2021; Moyer et al., 2018) in detail in Section 5.

Flow-based Methods. Leveraging the development of invertible normalizing flows (Papamakarios et al., 2021), Grover et al. (2020) proposed a combination of flow-based and adversarial distribution matching objectives for domain adaptation. Usman et al. (2020) proposed another upper bound for flow-based models where one distribution is fixed. Recently, Cho et al. (2022) generalized prior flow-based methods under a common framework by proving that the following flow-based alignment upper bound (AUB) is an *upper bound* on GJSD[3]:

$$\min_g \left(\min_{p(\boldsymbol{z}) \in \mathcal{P}} \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z},d)} \big[-\log \left(\lvert J_g(\boldsymbol{x}|d)\rvert \cdot p(\boldsymbol{z}) \right) \big] \right) \\ = \min_g \mathrm{AUB}(g) \ge \min_g \mathrm{GJSD}(\{q(z|d)\}_{d=1}^k).\,,$$

Like VAE-based methods, this forms a min-min problem but the optimization is over the prior distribution $p(\boldsymbol{z})$ rather than a decoder. But, unlike VAE-based methods, AUB does not impose any constraints on the shared latent distribution $p(\boldsymbol{z})$, i.e., it does not have to be a fixed latent distribution. While AUB (Cho et al., 2022) provides an elegant characterization of distribution matching theoretically, the implementation of AUB still suffers from two issues. First, AUB assumes that its aligner g is invertible, which requires specialized architectures and can be challenging to optimize in practice. Second, AUB inherits the vanishing gradient problem of theoretic JSD even if the bound is made tight—which was originally pointed out by Arjovsky and Bottou (2017). We aim to address these two issues in the subsequent sections and then revisit VAE-based matching to see how our resulting matching loss is both similar and different from prior VAE-based methods.

[3]refer to Appendix A for more background on AUB

3 RELAXING INVERTIBILITY CONSTRAINT OF AUB VIA VAES

One key limitation of the AUB matching measure is that g must be invertible, which can be challenging to enforce and optimize. Yet, the invertibility of g provides two distinct properties. First, invertibility enables exact log likelihood computation via the change-of-variables formula for invertible functions. Second, invertibility perfectly preserves mutual information between the observed and latent space conditioned on the domain label, i.e., $I(\boldsymbol{x}, \boldsymbol{z}|d)$ achieves its maximal value. Therefore, we aim to relax invertibility while seeking to retain the benefits of invertibility as much as possible. Specifically, we approximate the log likelihood via a VAE approach, which we show is a true relaxation of the Jacobian determinant computation, and we add a mutual information term $I(\boldsymbol{x}, \boldsymbol{z}|d)$, which attains its maximal value if the encoder is invertible. These two relaxation steps together yield an distribution matching objective that is mathematically similar to the domain-conditional version of β-VAE (Higgins et al., 2017) where $\beta \le 1$. Finally, we propose a plug-and-play version of our objective that can be used as a drop-

in replacement for adversarial loss terms so that the matching bounds can be used in any model pipeline.

3.1 VAE-based Alignment Upper Bound (VAUB)

We will first relax invertibility by replacing g with a stochastic autoencoder, where $q(\boldsymbol{z}|\boldsymbol{x}, d)$ denotes the encoder and $p(\boldsymbol{x}|\boldsymbol{z}, d)$ denotes the decoder. As one simple example, the encoder could be a deterministic encoder g plus some learned Gaussian noise, i.e., $q_g(\boldsymbol{z}|\boldsymbol{x}, d) = \mathcal{N}(g(\boldsymbol{x}|d), \sigma^2(\boldsymbol{x}, d)I)$. The marginal latent encoder distribution is $q(\boldsymbol{z}|d) = \int_{\mathcal{X}} q(\boldsymbol{x}|d)q(\boldsymbol{z}|\boldsymbol{x}, d)\mathrm{d}\boldsymbol{x}$. Given this, we can define an VAE-based objective and prove that it is an upper bound on the GJSD. All proofs are provided in Appendix E.

Definition 1 (VAE Alignment Upper Bound (VAUB))**.** The VAUB($q(\boldsymbol{z}|\boldsymbol{x}, d)$) of a probabilistic aligner (i.e., encoder) $q(\boldsymbol{z}|\boldsymbol{x}, d)$ is defined as:

$$\min_{\substack{p(\boldsymbol{z}) \\ p(\boldsymbol{x}|\boldsymbol{z},d)}} \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z},d)}\left[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z}, d)}{q(\boldsymbol{z}|\boldsymbol{x}, d)} \cdot p(\boldsymbol{z})\right] + C, \quad (1)$$

where $C \triangleq -\mathbb{E}_{q(d)}[\mathrm{H}(q(\boldsymbol{x}|d))]$ is constant w.r.t. $q(\boldsymbol{z}|\boldsymbol{x}, d)$ and thus can be ignored during optimization.

Theorem 1 (VAUB is an upper bound on GJSD)**.** *VAUB is an upper bound on GJSD between the latent distributions $\{q(\boldsymbol{z}|d)\}_{d=1}^k$ with a bound gap of $\mathrm{KL}(q(\boldsymbol{z}), p(\boldsymbol{z})) + \mathbb{E}_{q(d)q(\boldsymbol{z}|d)}[\mathrm{KL}(q(\boldsymbol{x}|\boldsymbol{z}, d), p(\boldsymbol{x}|\boldsymbol{z}, d))]$ that can be made tight if the $p(\boldsymbol{x}|\boldsymbol{z}, d)$ and $p(\boldsymbol{z})$ are optimized over all possible densities.*

While prior works proved a similar bound (Louizos et al., 2015; Gupta et al., 2021), an important difference is that this bound can be made tight if optimized over the shared latent distribution $p(\boldsymbol{z})$, whereas prior works assume $p(\boldsymbol{z})$ is a fixed normal distribution (more details in Section 5). Thus, VAUB is a more direct relaxation of AUB, which optimizes over $p(\boldsymbol{z})$. Another insightful connection to the flow-based AUB Cho et al. (2022) is that the term $\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},d)}[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}]$ can be seen as an upper bound generalization of the $-\log|J_g(\boldsymbol{x}|d)|$, similar to the correspondence noticed in Nielsen et al. (2020). The following proposition proves that this term is indeed a strict generalization of the Jacobian determinant term.

Proposition 2. *If the decoder is optimal, i.e., $p(\boldsymbol{x}|\boldsymbol{z}, d) = q(\boldsymbol{x}|\boldsymbol{z}, d)$, then the decoder-encoder ratio is the ratio of the marginal distributions: $\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},d)}\Big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}\Big] = \mathbb{E}_q\Big[-\log \frac{q(\boldsymbol{x}|d)}{q(\boldsymbol{z}|d)}\Big]$. If the encoder is also invertible, i.e., $q(\boldsymbol{z}|\boldsymbol{x}, d) = \delta(\boldsymbol{z} - g(\boldsymbol{x}))$, where δ is a Dirac delta, then the ratio is equal to the Jacobian determinant: $\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},d)}\Big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}\Big] = -\log|J_g(\boldsymbol{x}|d)|$.*

This proposition gives a stochastic version of the change of (probability) volume under transformation. In the invertible case, this is captured by the Jacobian determinant. While for the stochastic case, given an input $\boldsymbol{x}$, consider sampling multiple latent points from the posterior $q(\boldsymbol{z}|\boldsymbol{x}, d)$, which can be thought of as a posterior mean prediction plus some small noise. Now take the expected ratio between the marginal densities of $q(\boldsymbol{z}|d)$ for each sample point and $q(\boldsymbol{x}|d)$. If on average $q(\boldsymbol{x}|d)/q(\boldsymbol{z}|d) > 1$, then the transformation locally expands the space (akin to determinant greater than 1) and vice versa. This ratio estimator can consider volume changes due to non-invertibility and stochasticity. For example, $\boldsymbol{z} = b\boldsymbol{x} + \epsilon$ for some $b < 1$ and independent noise ϵ, locally "shrinks" because $b < 1$ but also locally expands because the noise flattens the distribution.

While the VAUB looks similar to standard VAE objectives, the key difference is noticing the role of d in the bound. Specifically, the encoder and decoder can be conditioned on the domain d but the trainable prior $p(\boldsymbol{z})$ is *not conditioned on the domain d*. This shared prior ties all the latent domain distributions together so that the optimal is only achieved when $q(\boldsymbol{z}|d) = q(\boldsymbol{z})$ for all d. Additionally, the perspective here is flipped from the VAE generative model perspective; rather than focusing on the generative model p the goal is finding the encoder q while p is seen as a variational distribution used to learn q. Finally, we note that VAUB could accommodate the case where the encoder is shared, i.e., it does not depend on d so that $q(\boldsymbol{z}|\boldsymbol{x}, d) = q(\boldsymbol{z}|\boldsymbol{x})$. However, the dependence of the decoder $p(\boldsymbol{x}|\boldsymbol{z}, d)$ on d should be preserved (otherwise the domain information would be totally ignored and distribution matching would not be enforced).

3.2 Preserving Mutual Information via Reconstruction Loss

While the previous section proved an alignment upper bound for probabilistic aligners based on VAEs, we would also like to preserve the property of flow-based methods that preserves the mutual information between the observed and latent spaces. Formally, for flow-based aligners g, we have that by construction $I(\boldsymbol{x}, \boldsymbol{z}|d) = I(\boldsymbol{x}, g^{-1}(\boldsymbol{z}|d)|d) = I(\boldsymbol{x}, \boldsymbol{x}|d) = \mathrm{H}(\boldsymbol{x}|d)$, i.e., no information is lost. Instead of requiring exact invertibility, we relax this property by maximizing the mutual information between $\boldsymbol{x}$ and $\boldsymbol{z}$ given the domain d. Mutual information can be lower bounded by the negative log likelihood of a decoder (i.e., the reconstruction loss term of VAEs), i.e., $I(\boldsymbol{x}, \boldsymbol{z}|d) \geq$

$\max_{\tilde{p}(\boldsymbol{x}|\boldsymbol{z},d)} \mathbb{E}_{q(\boldsymbol{x}|d)}[\log \tilde{p}(\boldsymbol{x}|\boldsymbol{z},d)] + C$, where $\tilde{p}$ is a variational decoder and C is independent of model parameters (though well-known, we include the proof in Appendix E for completeness). Similar to the previous section, this relaxation is a strict generalization of invertibility in the sense that mutual information is maximal in the limit of the encoder being exactly invertible. While technically this decoder $\tilde{p}$ could be different from the alignment-based p, it is natural to make them the same so that $\tilde{p}(\boldsymbol{x}|\boldsymbol{z},d) \triangleq p(\boldsymbol{x}|\boldsymbol{z},d)$. Therefore, this additional reconstruction loss can be directly combined with the overall objective:

$$
\begin{aligned}
&\mathrm{GJSD}(\{q(\boldsymbol{z}|d)\}_{d=1}^{k}) + \lambda \mathbb{E}_q[-I(\boldsymbol{x},\boldsymbol{z}|d)] + C \\
&\le \min_{\substack{p(\boldsymbol{z}) \\ p(\boldsymbol{x}|\boldsymbol{z},d)}} \underbrace{\mathbb{E}_q\Big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)} p(\boldsymbol{z})\Big]}_{\text{VAUB objective}} + \lambda \underbrace{\mathbb{E}_q[-\log p(\boldsymbol{x}|\boldsymbol{z},d)]}_{\text{Bound on } I(\boldsymbol{z},\boldsymbol{x}|d)} \\
&= \min_{\substack{p(\boldsymbol{z}) \\ p(\boldsymbol{x}|\boldsymbol{z},d)}} \tfrac{1}{\beta} \underbrace{\mathbb{E}_q\Big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)^{\beta}} p(\boldsymbol{z})^{\beta}\Big]}_{\beta\text{-VAUB obj with } \beta \triangleq \frac{1}{1+\lambda}}
\end{aligned}
$$

where $\lambda \ge 0$ is the mutual information regularization and $\beta \triangleq \frac{1}{1+\lambda} \le 1$ is a hyperparameter reparametrization that matches the form of β-VAE (Higgins et al., 2017). While the form is similar to a vanilla β-VAE (except for conditioning on the domain label d), the goal of β here is to encourage good reconstruction while also ensuring distribution matching rather than making features independent or more disentangled as in the original paper (Higgins et al., 2017). Therefore, we always use $\lambda \ge 0$ (or equivalently $\beta \le 1$). As will be discussed when comparing to other VAE-based matching methods, this modification is critical for the good performance of VAUB, particularly to avoid posterior collapse. Indeed, posterior collapse can satisfy latent distribution matching but would not preserve any information about the input, i.e., if $q(\boldsymbol{z}|\boldsymbol{x},d) = q_{\mathcal{N}(0,I)}(\boldsymbol{z})$, then the latent distributions will be trivially matched but no information will be preserved, i.e., $I(\boldsymbol{x},\boldsymbol{z}|d) = 0$.

3.3 Plug-and-Play Matching Loss

While it may seem that VAUB requires a VAE model, we show in this section that VAUB can be encapsulated into a self-contained loss function similar to the self-contained adversarial loss function. Specifically, using VAUB, we can create a plug and play matching loss that can replace any adversarial loss with a non-adversarial counterpart *without requiring any architecture changes to the original model*.

Definition 2 (Plug-and-play matching loss)**.** Given a deterministic feature extractor g, let $q_{g,\sigma^2}(\boldsymbol{z}|\boldsymbol{x},d) \triangleq \mathcal{N}(g(\boldsymbol{x}|d), \mathrm{diag}(\sigma^2(\boldsymbol{x},d)))$ be a simple probabilistic version of g where $\sigma^2(\boldsymbol{x},d)$ is a trainable diagonal convariance matrix. Then, the plug-and-play matching loss for any deterministic representation function g can be defined as:

$$
\begin{aligned}
&\text{VAUB\_PnP}(g) \triangleq \beta\text{-VAUB}(q_{g,\sigma^2}(\boldsymbol{z}|\boldsymbol{x},d)) \qquad (2) \\
&= \min_{\substack{\sigma^2(\boldsymbol{x},d) \\ p(\boldsymbol{x}|\boldsymbol{z},d), p(\boldsymbol{z})}} \mathbb{E}_q\left[-\log\left(\frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q_{g,\sigma^2}(\boldsymbol{z}|\boldsymbol{x},d)^{\beta}} p(\boldsymbol{z})^{\beta}\right)\right].
\end{aligned}
$$

Like the adversarial loss in Eqn. 1, this loss is a self-contained variational optimization problem where the auxiliary models (i.e., discriminator for adversarial and decoder distributions for VAUB) are only used for distribution matching optimization. This loss does not require the main pipeline to be stochastic and these auxiliary models could be simple functions. While weak auxiliary models for adversarial could lead to an arbitrarily poor approximation to GJSD, our upper bound is guaranteed to be an upper bound even if the auxiliary models are weak. Thus, we suggest that our β-VAUB_PnP loss function can be used to safely replace any adversarial loss function.

4 NOISY JENSEN-SHANNON DIVERGENCE

While in the previous section we addressed the invertibility limitation of AUB, we now consider a different issue related to optimizing a bound on the JSD. As has been noted previously Arjovsky et al. (2017), the standard JSD can saturate when the distributions are far from each other which will produce vanishing gradients. While Arjovsky et al. (2017) switch to using Wasserstein distance instead of JSD, we revisit the idea of adding noise to the JSD as in the predecessor work Arjovsky and Bottou (2017). Arjovsky and Bottou (2017) suggest smoothing the input distributions with Gaussian noise to make the distributions absolutely continuous and prove that this noisy JSD is an upper bound on the Wasserstein distance. However, in Arjovsky and Bottou (2017), the JSD is estimated via an adversarial loss, which is a *lower bound* on JSD. Thus, it is incompatible with their theoretic upper bound on Wasserstein distance. In contrast, because we have proven an upper bound on JSD, we can also consider a upper bound on noisy JSD that could avoid some of the problems with the standard JSD. We first define noisy JSD and prove that it is in fact a true divergence.

Definition 3 (Noisy JSD)**.** Noisy JSD is the JSD after adding Gaussian noise to the distributions, i.e., $\mathrm{NJSD}_{\sigma}(p,q) = \mathrm{JSD}(\tilde{p}_{\sigma}, \tilde{q}_{\sigma})$, where $\tilde{p}_{\sigma} \triangleq p * \mathcal{N}(0,\sigma^2 I)$ ($*$ denotes convolution) and similarly for $\tilde{q}_{\sigma}$.

Note that in terms of random variables, if $x \sim p_x$, $y \sim p_y$, and $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, then $\text{NJSD}(p_x, p_y) = \text{NJSD}(p_{x+\epsilon}, p_{y+\epsilon})$. We prove that Noisy JSD is indeed a statistical divergence using the properties of JSD and the fact that convolution with a Gaussian density is invertible.

Proposition 3. *Noisy JSD is a statistical divergence.*

In Appendix B, we present a toy example illustrating how adding noise to JSD can alleviate plateaus in the theoretic JSD that can cause vanishing gradient issues and can smooth over local minimum in the optimization landscape. We now prove noisy versions of both AUB and VAUB to be upper bounds of Noisy JSD. To the best of our knowledge, these bounds are novel though straightforward in hindsight.

Theorem 4 (Noisy alignment upper bounds)**.** *For the flow-based AUB, the following upper bound holds for :*

$$\begin{aligned}
&\text{NAUB}(q(\boldsymbol{z}|\boldsymbol{x}, d); \sigma^2) + C \\
&\triangleq \min_{p(\tilde{\boldsymbol{z}})} \mathbb{E}_{q(\boldsymbol{x},d)q(\epsilon;\sigma^2)}[-\log |J_g(\boldsymbol{x}|d)| p(g(\boldsymbol{x}|d) + \epsilon)] \\
&\geq \text{NJSD}(\{q(\boldsymbol{z}|d)\}_{d=1}^k; \sigma^2)\,,
\end{aligned}$$

where $C \triangleq \mathbb{E}_{q(d)}[\text{H}(q(\boldsymbol{x}|d))]$*. Similarly, for VAUB, the following upper bound holds:*

$$\begin{aligned}
&\text{NVAUB}(q(\boldsymbol{z}|\boldsymbol{x}, d); \sigma^2) + C \\
&\triangleq \min_{\substack{p(\tilde{\boldsymbol{z}}) \\ p(\boldsymbol{x}|\boldsymbol{z},d)}} \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z},d)q(\epsilon;\sigma^2)} \left[-\log \left(\frac{p(\boldsymbol{x}|\boldsymbol{z}, d)}{q(\boldsymbol{z}|\boldsymbol{x}, d)} \cdot p(\boldsymbol{z} + \epsilon) \right) \right] \\
&\geq \text{NJSD}(\{q(\boldsymbol{z}|d)\}_{d=1}^k; \sigma^2)\,.
\end{aligned}$$

By comparing the original objectives and these noisy objectives, we notice the correspondence between adding noise before passing to the shared distribution $p(\boldsymbol{z} + \epsilon)$ and the noisy JSD. This suggests that simple additive noise can add an implicit regularization that could make the optimization smoother. Similar to VAUB, a β-VAUB version of these can be used to preserve mutual information between $\boldsymbol{x}$ and $\boldsymbol{z}$.

5 REVISITING VAE-BASED MATCHING METHODS FROM FAIRNESS LITERATURE

The literature on fair classification has proposed several VAE-based methods for distribution matching. For fairness applications, the global objective includes both a classification loss and an matching loss but we will only analyze the matching losses in this paper. Louizos et al. (2015) first proposed the vanilla form of a VAE with an matched latent space where the prior distribution is fixed. Moyer et al. (2018) and Gupta et al. (2021) take a mutual information perspective and bound two different mutual information terms in different ways. They formulate the problem as minimizing the mutual information between $\boldsymbol{z}$ and d, where d corresponds to their sensitive attribute—our generalized JSD is in fact equivalent to this mutual information term, i.e., $\text{GJSD}(\{q(\boldsymbol{z}|d)\}_{d=1}^k) \equiv I(\boldsymbol{z}, d)$. They then use the fact that $I(\boldsymbol{z}, d) = I(\boldsymbol{z}, d|\boldsymbol{x}) + I(\boldsymbol{z}, \boldsymbol{x}) - I(\boldsymbol{z}, \boldsymbol{x}|d) = I(\boldsymbol{z}, \boldsymbol{x}) - I(\boldsymbol{z}, \boldsymbol{x}|d)$, where the second equals is by the fact that $\boldsymbol{z}$ is a deterministic function of $\boldsymbol{x}$ and independent noise. Finally, they bound $I(\boldsymbol{z}, \boldsymbol{x})$ and $-I(\boldsymbol{z}, \boldsymbol{x}|d)$ separately. We explain important differences here and point the reader to the Appendix C for a detailed comparison between methods.

We notice that most prior VAE-based methods use a fixed standard normal prior distribution $p_{\mathcal{N}}(\boldsymbol{z})$. This can be seen as a special case of our method in which the prior is not learnable. However, a fixed prior actually imposes constraints on the latent space beyond distribution matching, which we formalize in this proposition.

Proposition 5. *The fair VAE objective from Louizos et al. (2015) with a fixed latent distribution* $p_{\mathcal{N}(0,I)}(\boldsymbol{z})$ *can be decomposed into a VAUB term and a regularization term on the latent space:*

$$\begin{aligned}
&\min_{q(\boldsymbol{z}|\boldsymbol{x},d)} \min_{p(\boldsymbol{x}|\boldsymbol{z},d)} \mathbb{E}_q\big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z}, d)}{q(\boldsymbol{z}|\boldsymbol{x}, d)} p_{\mathcal{N}(0,I)}(\boldsymbol{z}) \big] \\
&= \min_{q(\boldsymbol{z}|\boldsymbol{x},d)} \underbrace{\Big(\min_{p(\boldsymbol{x}|\boldsymbol{z},d)} \min_{p(\boldsymbol{z})} \mathbb{E}_q\big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z}, d)}{q(\boldsymbol{z}|\boldsymbol{x}, d)} p(\boldsymbol{z}) \big] \Big)}_{\textit{VAUB Alignment Objective}} \\
&\qquad + \underbrace{\text{KL}(q(\boldsymbol{z}), p_{\mathcal{N}(0,I)}(\boldsymbol{z}))}_{\textit{Regularization}}\,.
\end{aligned}$$

This proposition highlights that prior VAE-based matching objectives are actually solving distribution matching *plus a regularization term* that pushes the learned latent distribution to the normal distribution—i.e., they are biased distribution matching methods. In contrast, GAN-based matching objectives do not have this bias as they do not assume anything about the latent space. Similarly, our VAUB methods can be seen as relaxing this by optimizing over a class of latent distributions for $p(\boldsymbol{z})$ to reduce the bias.

Furthermore, we notice that Moyer et al. (2018) used a similar term as ours for $-I(\boldsymbol{z}, \boldsymbol{x}|d)$ but used a non-parametric pairwise KL divergence term for $I(\boldsymbol{z}, \boldsymbol{x})$, which scales quadratically in the batch size. On the other hand, Gupta et al. (2021) uses a similar variational KL term as ours for $I(\boldsymbol{z}, \boldsymbol{x})$ but decided on a contrastive mutual information bound for $-I(\boldsymbol{z}, \boldsymbol{x}|d)$.

Gupta et al. (2021) did consider using a similar term for $-I(\boldsymbol{z}, \boldsymbol{x}|d)$ but ultimately rejected this alternative in favor of the contrastive approach. We summarize the differences as follows:

1. We allow the shared prior distribution $p_\theta(\boldsymbol{z})$ to be *learnable* so that we do not impose any distribution on the latent space. Additionally, optimizing $p_\theta(\boldsymbol{z})$ is significant as we show in Theorem 1 that this can make our bound tight.
2. The β-VAE change ensures better preservation of the mutula information of $\boldsymbol{x}$ and $\boldsymbol{z}$ inspired by the invertible models of AUB. This seemingly small change seems to overcome the limitation of the reconstruction-based approach originally rejected in Gupta et al. (2021).
3. We propose a novel noisy version of the bound that can smooth the optimization landscape while still being a proper divergence.
4. We propose a *plug-and-play* version of our bound that can be added to *any* model pipeline and replace *any* adversarial loss. Though not a large technical contribution, this perspective decouples the distribution matching loss from VAE models to create a self-contained distribution matching loss that can be broadly applied outside of VAE-based models.

6 EXPERIMENTS

6.1 Simulated Experiments

Non-Matching Dimensions between Latent Space and Input Space. In order to demonstrate that our model relaxes the invertibility constraint of AUB, we use a dataset consisting of rotated moons where the latent dimension does not match the input dimension (i.e. the transformation between the input space and latent space is not invertible). Please note that the AUB model proposed by Cho et al. (2022) is not applicable for such situations due to the requirement that the encoder and decoder need to be invertible, which restricts our ability to select the dimensionality of the latent space. As depicted in Fig. 2, our model is able to effectively reconstruct and flip the original two sample distributions despite lacking the invertible features between the encoders and decoders. Furthermore, we observe the mapped latent two sample distributions are matched with each other while sharing similar distributions to the shared distribution $p(z)$.

Noisy-AUB helps mitigate the Vanishing Gradient Problem. In this example, we demonstrate that the optimization can get stuck in a plateau region without noise injection. However, this issue can be resolved using the noisy-VAUB approach. Initially, we attempt to match two Gaussian distributions with widely separated means. As shown in Fig. 1, the shared distribution, $p(\boldsymbol{z})$ (which is a Gaussian mixture model), initially fits the bi-modal distribution, but this creates a plateau in the optimization landscape with small gradient even though the latent space is misaligned. While VAUB can eventually escape such plateaus with sufficient training time, these plateaus can unnecessarily prolong the training process. In contrast, NVAUB overcomes this issue by introducing noise in the latent domain and thereby reducing the small gradient issue.

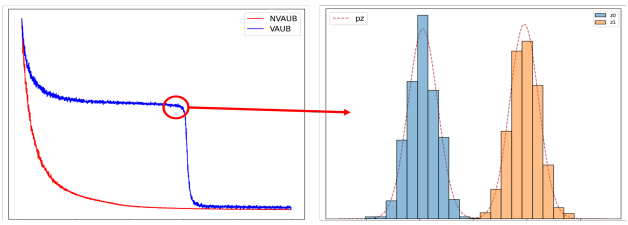
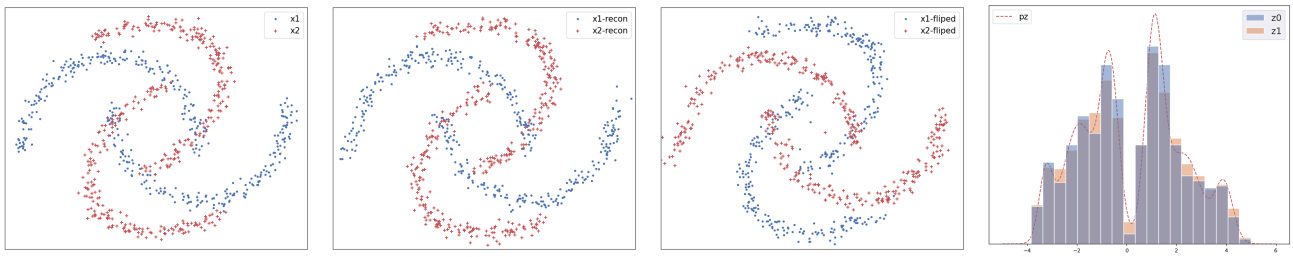


Figure 1: This figure shows that the loss reaches a plateau during learning if VAUB is used. (a) shows the loss convergence graph for VAUB and NVAUB, while (b) visualizes the latent distribution $p(\boldsymbol{z})$ with histogram density estimation of $z_i \sim q(z|x, d = i), i \in 0, 1$ at the **red circle** in figure (a). Notice that the latent distribution matches the mixture of the domains but the latent domain distributions are not yet aligned.

6.2 Other Non-adversarial Bounds

Due to the predominant focus on fairness learning in related works, we select the *Adult dataset*[4], comprising 48,000 instances of census data from 1994. To investigate distribution matching, we create domain samples by grouping the data based on gender attributes (male, female) and aim to achieve matching between these domains. We employ two alignemnt metrics: sliced Wasserstein distance (SWD) and a classification-based metric. For SWD, we obtain the latent sample distribution ($z_i \sim q(z|x_i)$) and whiten it to obtain z_i^{white}. Because Wasserstein distance is sensitive to scaling, the whitening step is required to remove the effects of scaling, which do not fundamentally change the distribution matching performance. We then measure the SWD between the whitened sample distributions $z^{white}male$ and z_{female}^{white} by projecting them onto randomly chosen directions and calculating the 1D Wasserstein distance. We use t-test to assess significant differences between models. For the classification-based metric, we train a Support Vector Machine (SVM) with a Gaussian kernel to classify the latent distribution, using gender as the label. A more effective distribution matching model is expected to exhibit lower classification accuracy, reflecting the increasing difficulty in differentiating between

[4]https://archive.ics.uci.edu/ml/datasets/adult



(a) Original dist. (b) Reconstructed dist. (c) Transformed dist. (d) Latent dist. with $p(\boldsymbol{z})$

Figure 2: This figure shows distribution matching on the 2D rotated moons dataset while the latent sample distribution is 1D. (a) shows the original distribution where x_1 is the original moons dataset and x_2 is the rotated and scaled moons dataset. (b) shows the reconstructed distribution generated by using the probabilistic encoder and decoder within the same domain, i.e. $x_i^{recon} \sim p(x|z, d_{=i})$ (c) shows the flipped distribution between two distributions by forwarding the probabilistic encoder and decoder in different domains, i.e. $x_i^{flipped} \sim p(x|z, d_{=j})$, where the dataset is flipped from i to j. (d) shows the shared distribution $p(\boldsymbol{z})$, along with the histogram density estimation of the latent distributions of $z_i \sim q(z|x, d = i), i \in 0, 1$.

matched distributions. We chose kernel SVM over training a deep model because the optimization is convex with a unique solution given the hyperparameters. In all cases, we used cross-validation to select the kernel SVM hyperparameters.

We choose two non-adversarial bounds Moyer et al. (2018); Gupta et al. (2021) as our baseline methods. Here we only focus on the distribution matching loss functions of all baseline methods. Note that these methods are closely related (see Section 5). To ensure a fair comparison, we employ the same encoders for all models and train them until convergence. The results in Table 2 suggest that our VAUB method can achieve better distribution matching compared to prior variational upper bounds.

Table 2: Distribution matching comparison between non-adversarial bounds (refer to Appendix F for t-test details)

	Moyer's	Gupta's	VAUB
SWD (↓)	9.71	6.70	**5.64**
SVM (↓)	0.997	0.833	**0.818**

6.3 Plug-and-Play Implementation

In this section, we see if our plug-and-play loss can be used to replace prior adversarial loss functions in generic model pipelines rather than those that are specifically VAE-based.

Replacing the Adversarial Objective in Fairness Representation Models. In this experiment, we explore the effectiveness of using our min-min VAUB plug-and-play loss as a replacement for the adversarial objective of LAFTR (Madras et al., 2018). We include the LAFTR classifying loss and adjust the trade-off between the classification loss and matching loss via λ_{aub} Definition 2. In this section, we conduct a comparison between our proposed model and the LAFTR model on the Adult dataset where the goal is to fairly predict whether a person's income is above 50K, using gender as the sensitive attribute. To demonstrate the plug-and-play feature of our VAUB model, we employ the same network architecture for the deterministic encoder g and classifier h as in LAFTR-DP. We also share the parameters of our encoders (i.e. $q(z|x, d) = q(z|x)$) to comply with the structure of the LAFTR. The results of the fair classification task are presented in Table 3, where we evaluate using three metrics: overall accuracy, demographic parity gap (Δ_{DP}), and test VAUB loss. We argue that our model(VAUB) still maintains the trade-off property between classification accuracy and fairness while producing similar results to LAFTR-DP. Moreover, we observe that LAFTR-DP cannot achieve perfect fairness in terms of demographic parity gap by simply increasing the fairness coefficient γ, whereas our model can achieve a gap of 0 with only a small compromise in accuracy. Finally, we note that the VAUB loss correlates well with the demographic parity gap, indicating that better distribution matching leads to improved fairness; perhaps more importantly, this gives some evidence that VAUB may be useful for measuring the relative distribution matching between two approaches. Please refer to the appendix for a more comprehensive explanation of all the experiment details.

Comparison of Training Stability between Adversarial Methods and Ours. To demonstrate the stability of our non-adversarial training objective,

Table 3: Accuracy, Demographic Parity Gap (Δ_{DP}), and VAUB Metric (in nats) on Adult Dataset with the models VAUB and LAFTR-DP. Numbers after VAUB indicate λ_{aub}, and numbers after LAFTR-DP indicate the fairness coefficient γ.

Method	**Accuracy**	Δ_{DP}	**VAUB**
VAUB(100)	0.74	0	308.36
VAUB(50)	0.76	0.0002	318.31
VAUB(10)	0.797	0.058	324.28
VAUB(1)	0.835	0.252	333.47
LAFTR-DP(1000)	0.765	0.015	-
LAFTR-DP(4)	0.77	0.022	-
LAFTR-DP(0.1)	0.838	0.21	-

we use plug-and-play VAUB model to replace the adversarial objective of DANN (Ganin et al., 2016). We re-define the min-max objective of DANN to a min-min optimization objective while keeping the model the same. Because of the plug-and-play feature, we can use the exact same network architecture of the encoder (feature extractor) and label classifier as proposed in DANN and only replace the adversarial loss. We conduct the experiment on the MNIST (LeCun et al., 2010) and MNIST-M (Ganin et al., 2016) datasets, using the former as the source domain and the latter as the target domain. We present the results of our model in Table 4[5], which reveals a comparable or better accuracy compared to DANN. Notably, the NVAUB approach exhibits further improvement over accuracy, as adding noise may smooth the optimization landscape. In contrast to the adversarial method, our model provides a reliable metric (VAUB loss) for assessing adaptation performance. The VAUB loss shows a strong correlation with test accuracy, whereas the adversarial method lacks a valid metric for evaluating adaptation quality (see Appendix D for figure). This result on DANN and the previous on LAFTR give evidence that our method could be used as drop-in replacements for adversarial loss functions while retaining the performance and matching of adversarial losses.

Table 4: The table shows the accuracy after the domain adaptation in DANN, VAUB and NVAUB models. For each model, results are averaged from 5 experiments with different random seeds.

Method	DANN	VAUB	NVAUB
Accuracy (↑)	75.42	75.53	**76.47**

[5]We optimized the listed results for the DANN experiment to the best of our ability.

7 LIMITATION

Empirical Scope. Since this paper is theoretically focused rather than empirically focused, our goal was to prove theoretic bounds for distribution matching, elucidate insightful connections to prior works (AUB, fair VAE, and GAN methods), and then empirically validate our methods compared to other related approaches on simple targeted experiments, which means the experiments conducted in our study are deliberately simple and may not fully capture the complexity of real-world scenarios. We primarily utilize toy datasets such as the 2D moons dataset and 1D noisy VAUB illustrations to illustrate key insights. Although we validate our methods on common benchmark datasets like Adult and MNIST, our choice to avoid state-of-the-art methods and complex datasets may limit the generalizability of our findings.

Comparisons with SOTA models. We intentionally select representative methods such as LAFTR and DANN, along with well-known benchmark datasets, to demonstrate the feasibility of our approach. However, this choice may not fully capture the diversity of existing methods and datasets used in practical applications. Also, our study does not aim for state-of-the-art performance on specific tasks, which may lead to overlooking certain performance metrics or nuances that are crucial in practical applications. While our approach shows promise as an alternative to adversarial losses, further exploration is needed to understand its performance across various metrics and tasks.

8 DISCUSSION AND CONCLUSION

In conclusion, we present a model-agnostic VAE-based distribution matching method that can be seen as a relaxation of flow-based matching or as a new variant of VAE-based methods. Unlike adversarial methods, our method is non-adversarial, forming a min-min cooperative problem that provides upper bounds on JSD divergences. We propose noisy JSD variants to avoid vanishing gradients and local minima and develop corresponding alignment upper bounds. We compare to other VAE-based bounds both conceptually and empirically showing how our bound differs. Finally, we demonstrate that our non-adversarial VAUB alignment losses can replace adversarial losses without modifying the original model's architecture, making them suitable for standard invariant representation pipelines such as DANN or fair representation learning. We hope this will enable distribution matching losses to be applied generically to different problems without the challenges of adversarial losses.

Acknowledgement

Z.G. and D.I. acknowledge support from NSF (IIS-2212097) and ARL (W911NF-2020-221). H.Z. was partly supported by the Defense Advanced Research Projects Agency (DARPA) under Cooperative Agreement Number: HR00112320012, an IBM-IL Discovery Accelerator Institute research award, and Amazon AWS Cloud Credit.

References

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

Cho, W., Gong, Z., and Inouye, D. I. (2022). Cooperative distribution alignment via jsd upper bound. In *Neural Information Processing Systems (NeurIPS)*.

Creager, E., Madras, D., Jacobsen, J.-H., Weis, M., Swersky, K., Pitassi, T., and Zemel, R. (2019). Flexibly fair representation learning by disentanglement. In *International conference on machine learning*, pages 1436–1445. PMLR.

Farnia, F. and Ozdaglar, A. (2020). Do GANs always have Nash equilibria? In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3029–3039. PMLR.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Grover, A., Chute, C., Shu, R., Cao, Z., and Ermon, S. (2020). Alignflow: Cycle consistent learning from multiple domains via normalizing flows. In *AAAI*.

Gupta, U., Ferber, A. M., Dilkina, B., and Ver Steeg, G. (2021). Controllable guarantees for fair outcomes via contrastive information estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7610–7619.

Han, X., Chi, J., Chen, Y., Wang, Q., Zhao, H., Zou, N., and Hu, X. (2023). FFB: A Fair Fairness Benchmark for In-Processing Group Fairness Methods.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

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

Kurach, K., Lucic, M., Zhai, X., Michalski, M., and Gelly, S. (2019). The GAN landscape: Losses, architectures, regularization, and normalization.

LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Louizos, C., Swersky, K., Li, Y., Welling, M., and Zemel, R. (2015). The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. (2018). Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, pages 700–709.

Madras, D., Creager, E., Pitassi, T., and Zemel, R. (2018). Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pages 3384–3393. PMLR.

Moyer, D., Gao, S., Brekelmans, R., Galstyan, A., and Ver Steeg, G. (2018). Invariant representations without adversarial training. *Advances in Neural Information Processing Systems*, 31.

Nie, W. and Patel, A. B. (2020). Towards a better understanding and regularization of gan training dynamics. In *Uncertainty in Artificial Intelligence*, pages 281–291. PMLR.

Nielsen, D., Jaini, P., Hoogeboom, E., Winther, O., and Welling, M. (2020). Survae flows: Surjections to bridge the gap between vaes and flows. *Advances in Neural Information Processing Systems*, 33:12685–12696.

Papamakarios, G., Nalisnick, E. T., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(57):1–64.

Tran, N.-T., Tran, V.-H., Nguyen, N.-B., Nguyen, T.-K., and Cheung, N.-M. (2021). On data augmentation for gan training. *IEEE Transactions on Image Processing*, 30:1882–1897.

Usman, B., Sud, A., Dufour, N., and Saenko, K. (2020). Log-likelihood ratio minimizing flows: Towards robust and quantifiable neural distribution alignment.

Advances in Neural Information Processing Systems, 33:21118–21129.

Wu, Y., Zhou, P., Wilson, A. G., Xing, E., and Hu, Z. (2020). Improving gan training with probability ratio clipping and sample reweighting. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5729–5740. Curran Associates, Inc.

Xu, D., Yuan, S., Zhang, L., and Wu, X. (2018). Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575. IEEE.

Zhao, H., Coston, A., Adel, T., and Gordon, G. J. (2020). Conditional learning of fair representations. In *International Conference on Learning Representations*.

Zhao, H., Zhang, S., Wu, G., Moura, J. M., Costeira, J. P., and Gordon, G. J. (2018). Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31.

Checklist

The checklist follows the references. For each question, choose your answer from the three possible options: Yes, No, Not Applicable. You are encouraged to include a justification to your answer, either by referencing the appropriate section of your paper or providing a brief inline description (1-2 sentences). Please do not modify the questions. Note that the Checklist section does not count towards the page limit. Not including the checklist in the first submission won't result in desk rejection, although in such case we will ask you to upload it during the author response period and include it in camera ready (if accepted).

In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

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

Appendix for Towards Practical Non-Adversarial Distribution Alignment

Table of Contents

A More Background and Theory from AUB (Cho et al., 2022)

Alignment Upper Bound (AUB), introduced from Cho et al. (2022), jointly learns invertible deterministic aligners g with a shared latent distribution $p(\boldsymbol{z})$, where $\boldsymbol{z} = g(\boldsymbol{x}|d)$, or equivalently where $q(\boldsymbol{z}|\boldsymbol{x}, d)$ is a Dirac delta function centered at $g(\boldsymbol{x}|d)$. In this section, we remember several important theorems and lemmas from Cho et al. (2022) based on invertible aligners g. These provide both background and formal definitions. Our proofs follow a similar structure to those in Cho et al. (2022).

Theorem 6 (GJSD Upper Bound from Cho et al. (2022))**.** *Given a density model class $\mathcal{P}$, we form a GJSD variational upper bound:*

$$\mathrm{GJSD}(\{q(\boldsymbol{z}|d)\}_{d=1}^{k}) \leq \min_{p(\boldsymbol{z})\in\mathcal{P}} \mathrm{H_c}(q(\boldsymbol{z}), p(\boldsymbol{z})) - \mathbb{E}_{q(d)}[\mathrm{H}(q(\boldsymbol{z}|d))]\,,$$

where $q(\boldsymbol{z}) = \sum_d \int q(\boldsymbol{x}, d)q(\boldsymbol{z}|\boldsymbol{x}, d)\mathrm{d}\boldsymbol{x} = \sum_d q(d)q(\boldsymbol{z}|d)$ is the marginal of the encoder distribution and the bound gap is exactly $\min_{p(\boldsymbol{z})\in\mathcal{P}} \mathrm{KL}(q(\boldsymbol{z}), p(\boldsymbol{z}))$.

Lemma 7 (Entropy Change of Variables from Cho et al. (2022))**.** *Let $\boldsymbol{x} \sim q(\boldsymbol{x})$ and $\boldsymbol{z} \triangleq g(\boldsymbol{x}) \sim q(\boldsymbol{z})$, where g is an invertible transformation. The entropy of $\boldsymbol{z}$ can be decomposed as follows:*

$$\mathrm{H}(q(\boldsymbol{z})) = \mathrm{H}(q(\boldsymbol{x})) + \mathbb{E}_{q(\boldsymbol{x})}[\log|J_g(\boldsymbol{x})|]\,, \tag{3}$$

where $|J_g(\boldsymbol{x})|$ is the absolute value of the determinant of the Jacobian of g.

Definition 4 (Alignment Upper Bound Loss from Cho et al. (2022))**.** The alignment upper bound loss for aligner $g(\boldsymbol{x}|d)$ that is invertible conditioned on d is defined as follows:

$$\mathrm{AUB}(g) \triangleq \min_{p(\boldsymbol{z})\in\mathcal{P}} \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z},d)}\big[-\log\big(|J_g(\boldsymbol{x}|d)|\cdot p(\boldsymbol{z})\big)\big], \tag{4}$$

where $\mathcal{P}$ is a class of shared prior distributions and $|J_g(\boldsymbol{x}|d)|$ is the absolute value of the Jacobian determinant.

Theorem 8 (Alignment at Global Minimum of AUB from Cho et al. (2022))**.** *If* AUB *is minimized over the class of all invertible functions, a global minimum of* AUB *implies that the latent distributions are aligned, i.e., for all d, d', $q(\boldsymbol{z}|d) = q(\boldsymbol{z}|d') \in \mathcal{P}$. Notably, this result holds regardless of $\mathcal{P}$.*

B Illustration of Noisy JSD for Alleviating Vanishing Gradient and Local Minimum

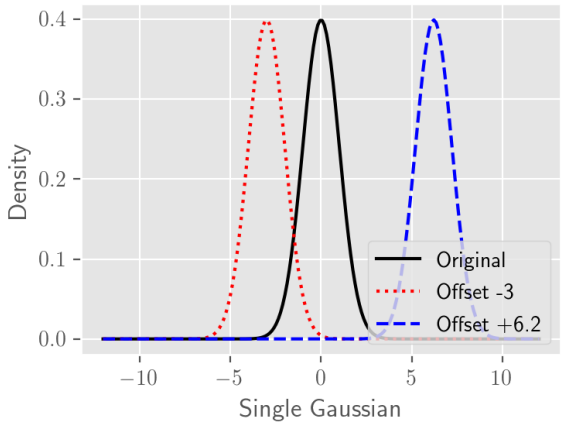
We present a toy example illustrating how adding noise to JSD can alleviate plateaus in the theoretic JSD that can cause vanishing gradient issues (Fig. 3(a-b)) and can smooth over local minimum in the optimization landscape (Fig. 3(c-d)).

C VAE-Based Distribution Alignment Comparison Table

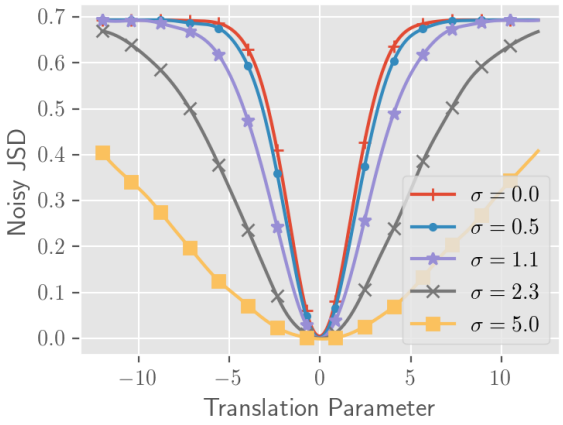
We present the full comparison table between VAE-based methods below in Table 5. We add the notation where ϕ are the parameters of the encoder distributions q_ϕ and θ are the parameters of the decoder distribution p_θ to emphasize that our VAUB objectives allow training of the prior distribution $p_\theta(\boldsymbol{z})$, while prior methods assume it is a standard normal distribution.

Table 5: Variational Upper Bounds Comparison: Prior bounds have one or more similarities to our bounds but have several key differences as noted below and explained in this section. C is a constant and CE stands for Contrastive Estimation which equals to $-\log \frac{\exp(f(\boldsymbol{z},\boldsymbol{x},d)}{\mathbb{E}_{\boldsymbol{z}'\sim q(\boldsymbol{z}|c)}[\exp(f(\boldsymbol{z}',\boldsymbol{x},d))]}$.

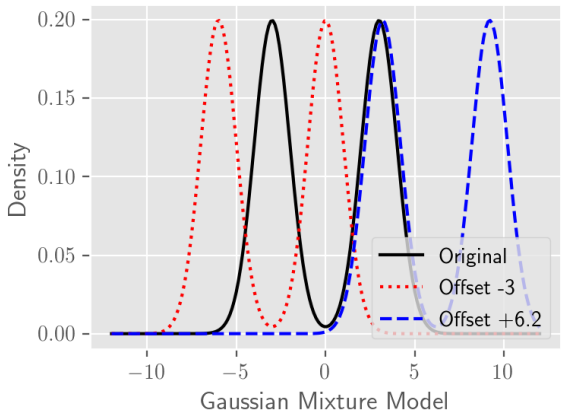
Method	$I(\boldsymbol{z},\boldsymbol{x}) \le \cdots$	$-I(\boldsymbol{z},\boldsymbol{x}\|d) \le \cdots + C$
Louizos et al. (2015)	$\mathbb{E}_q[\mathrm{KL}(q_\phi(\boldsymbol{z}\|\boldsymbol{x},d), p_{\mathcal{N}}(\boldsymbol{z}))]$	$\mathbb{E}_q[-\log p_\theta(\boldsymbol{x}\|\boldsymbol{z},d)]$
Moyer et al. (2018)	$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')\sim q}[\mathrm{KL}(q_\phi(\boldsymbol{z}\|\boldsymbol{x}), q_\phi(\boldsymbol{z}\|\boldsymbol{x}'))]$	$\mathbb{E}_q[-\log p_\theta(\boldsymbol{x}\|\boldsymbol{z},d)]$
Gupta et al. (2021)	$\mathbb{E}_q[\mathrm{KL}(q_\phi(\boldsymbol{z}\|\boldsymbol{x}), p_{\mathcal{N}}(\boldsymbol{z}))]$	$\mathbb{E}_q[-\mathrm{CE}(\boldsymbol{z},\boldsymbol{x},d)]$
Gupta et al. (2021) recon[6]	$\mathbb{E}_q[\mathrm{KL}(q_\phi(\boldsymbol{z}\|\boldsymbol{x}), p_{\mathcal{N}}(\boldsymbol{z}))]$	$\mathbb{E}_q[-\log p_\theta(\boldsymbol{x}\|\boldsymbol{z},d)]$
(ours) VAUB	$\mathbb{E}_q[\mathrm{KL}(q_\phi(\boldsymbol{z}\|\boldsymbol{x},d), p_\theta(\boldsymbol{z}))]$	$\mathbb{E}_q[-\log p_\theta(\boldsymbol{x}\|\boldsymbol{z},d)]$
(ours) β-VAUB ($\beta \le 1$)	$\mathbb{E}_q[\beta\,\mathrm{KL}(q_\phi(\boldsymbol{z}\|\boldsymbol{x},d), p_\theta(\boldsymbol{z}))]$	$\mathbb{E}_q[-\log p_\theta(\boldsymbol{x}\|\boldsymbol{z},d)]$
(ours) Noisy β-VAUB	$\mathbb{E}_q[\underbrace{\beta}_{(2)}\mathrm{KL}(q_\phi(\boldsymbol{z}\|\boldsymbol{x},d), \underbrace{p_\theta}_{(1)}(\boldsymbol{z}\underbrace{+\epsilon}_{(3)}))]$	$\mathbb{E}_q[-\log p_\theta(\boldsymbol{x}\|\boldsymbol{z},d)]$



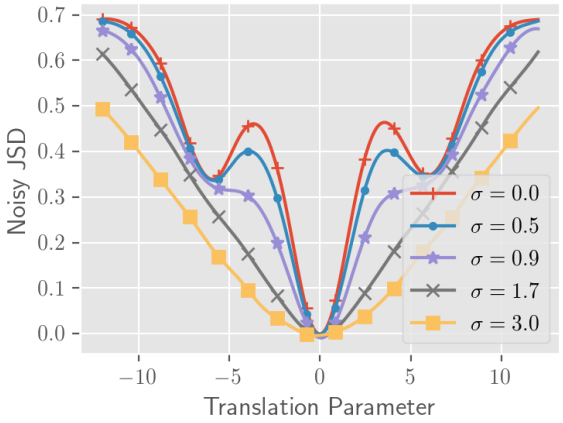
(a) Case 1: Gaussian distributions.



(b) Case 1: Opt. landscape for different noise levels.



(c) Case 2: Mixture of Gaussian distributions.

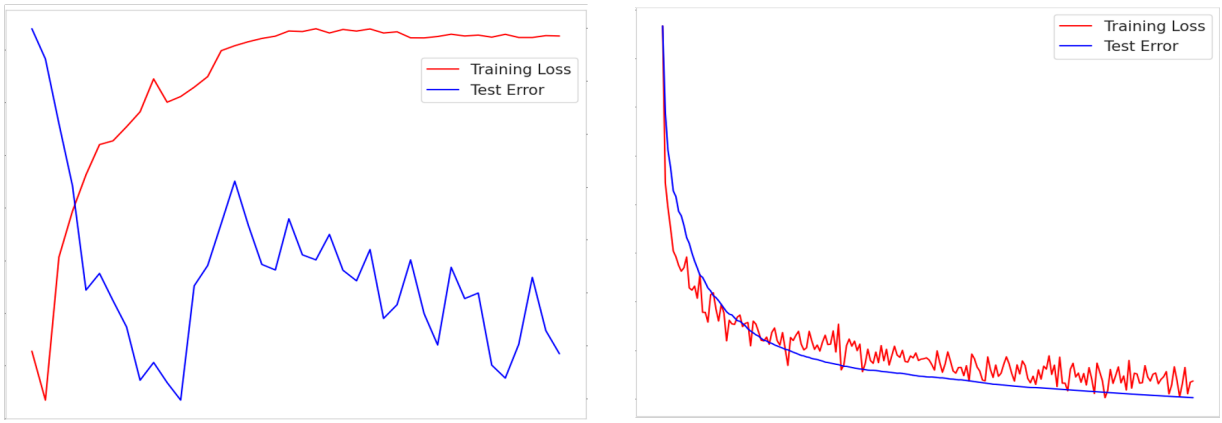


(d) Case 2: Opt. landscape for different noise levels

Figure 3: This figure illustrates that Noisy JSD can reduce the vanishing gradient problem and smooth over local minimum compared to theoretic JSD. In Case 1 (a), we consider the (Noisy) JSD between two Gaussian distributions whose variance are the same but whose means are different. The gradient of JSD can vanish to zero as it reaches its maximum value as seen by the plateau regions on the top curve in (b) but this can be alleviated with noise as seen in bottom curves in (b). In Case 2 (c), we consider the (Noisy) JSD between a Gaussian mixture model where the mixture components are the same but the overall means are different. For this case, a local minimum of JSD occurs when only one mixture component overlaps as seen in the top curve of (d). However, Noisy JSD can smooth out this local minimum so that there are no local minimum.

D Illustration of Differences Between Adversarial vs VAUB Loss

In Fig. 4, we illustrate that using an adversarial loss in DANN does not provide a good measure of test performance while our VAUB-based loss has a strong correlation with test error.



(a) DANN loss vs test accuracy

(b) VAUB loss vs test accuracy

Figure 4: This figure illustrates the tendency of test error and training loss for both models. For both figures, x-axis represents the epochs while the y-axis indicates the value of corresponding metric.

E Proofs

E.1 Proof that JSD is Invariant Under Invertible Transformations

See proof from Theorem 1 of Tran et al. (2021).

E.2 Proof of Entropy Change of Variables For Probabilistic Autoencoders

An entropy change of variables bound inspired by the derivations of surjective VAE flows Nielsen et al. (2020) can be derived so that we can apply a similar proof as in Cho et al. (2022).

Lemma 9 (Entropy Change of Variables for Probabilistic Autoencoders)**.** *Given an encoder $q(\boldsymbol{z}|\boldsymbol{x})$ and a variational decoder $p(\boldsymbol{x}|\boldsymbol{z})$, the latent entropy can be lower bounded as follows:*

$$\mathrm{H}(q(\boldsymbol{z})) = \mathrm{H}(q(\boldsymbol{x})) + \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z})}\left[\log \frac{p(\boldsymbol{x}|\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathbb{E}_{q(\boldsymbol{z})}[\mathrm{KL}(q(\boldsymbol{x}|\boldsymbol{z}), p(\boldsymbol{x}|\boldsymbol{z})] \geq \mathrm{H}(q(\boldsymbol{x})) + \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z})}\left[\log \frac{p(\boldsymbol{x}|\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})}\right], \quad (5)$$

where the bound gap is exactly $\mathbb{E}_{q(\boldsymbol{z})}[\mathrm{KL}(q(\boldsymbol{x}|\boldsymbol{z}), p(\boldsymbol{x}|\boldsymbol{z})]$, which can be made tight if the right hand side is maximized w.r.t. $p(\boldsymbol{x}|\boldsymbol{z})$.

Proof. Similar to the bounds for ELBO, we inflate by both the encoder $q(z|x)$ and the decoder $p(x|z)$ and

eventually bring out an expectation over a KL term.

$$
\begin{aligned}
\mathrm{H}(q(x)) &= \mathbb{E}_{q(x)}[-\log q(x)] && (6)\\
&= \mathbb{E}_{q(x)q(z|x)}[-\log q(x)] && (7)\\
&= \mathbb{E}_{q(x)q(z|x)}[\log q(z|x) - \log q(z|x)q(x)] && (8)\\
&= \mathbb{E}_{q(x)q(z|x)}[\log q(z|x) - \log q(z)q(x|z)] && (9)\\
&= \mathbb{E}_{q(x)q(z|x)}[-\log q(z) + \log q(z|x) - \log q(x|z)] && (10)\\
&= \mathrm{H}(q(z)) + \mathbb{E}_{q(x)q(z|x)}\left[\log \frac{q(z|x)}{q(x|z)}\right] && (11)\\
&= \mathrm{H}(q(z)) + \mathbb{E}_{q(x)q(z|x)}\left[\log \frac{q(z|x)p(x|z)}{p(x|z)q(x|z)}\right] && (12)\\
&= \mathrm{H}(q(z)) + \mathbb{E}_{q(x)q(z|x)}\left[\log \frac{q(z|x)}{p(x|z)}\right] + \mathbb{E}_{q(x)q(z|x)}\left[\log \frac{p(x|z)}{q(x|z)}\right] && (13)\\
&= \mathrm{H}(q(z)) + \mathbb{E}_{q(x)q(z|x)}\left[\log \frac{q(z|x)}{p(x|z)}\right] - \mathbb{E}_{q(z)}\left[\mathbb{E}_{q(x|z)}\left[\log \frac{q(x|z)}{p(x|z)}\right]\right] && (14)\\
&= \mathrm{H}(q(z)) + \mathbb{E}_{q(x)q(z|x)}\left[\log \frac{q(z|x)}{p(x|z)}\right] - \mathbb{E}_{q(z)}[\mathrm{KL}(q(x|z), p(x|z))]\,. && (15)
\end{aligned}
$$

By rearranging the above equation, we can easily derive the result from the non-negativity of KL:

$$
\begin{aligned}
\mathrm{H}(q(z)) &= \mathrm{H}(q(x)) - \mathbb{E}_{q(x)q(z|x)}\left[\log \frac{q(z|x)}{p(x|z)}\right] + \mathbb{E}_{q(z)}[\mathrm{KL}(q(x|z), p(x|z))] && (16)\\
&= \mathrm{H}(q(x)) + \mathbb{E}_{q(x)q(z|x)}\left[\log \frac{p(x|z)}{q(z|x)}\right] + \mathbb{E}_{q(z)}[\mathrm{KL}(q(x|z), p(x|z))] && (17)\\
&\geq \mathrm{H}(q(x)) + \mathbb{E}_{q(x)q(z|x)}\left[\log \frac{p(x|z)}{q(z|x)}\right]\,, && (18)
\end{aligned}
$$

where it is clear that the bound gap is exactly $\mathbb{E}_{q(z)}[\mathrm{KL}(q(x|z), p(x|z))]$. Furthermore, we note that by maximizing over all possible $p(x|z)$ (or minimizing the negative objective), we can make the bound tight:

$$
\begin{aligned}
&\arg\max_{p(x|z)} \mathrm{H}(q(x)) + \mathbb{E}_{q(x)q(z|x)}\left[\log \frac{p(x|z)}{q(z|x)}\right] && (19)\\
&= \arg\max_{p(x|z)} \mathbb{E}_{q(x)q(z|x)}[\log p(x|z)] && (20)\\
&= \arg\min_{p(x|z)} \mathbb{E}_{q(x)q(z|x)}[-\log p(x|z)] && (21)\\
&= \arg\min_{p(x|z)} \mathbb{E}_{q(x)q(z|x)}[-\log p(x|z) + \log q(x|z)] && (22)\\
&= \arg\min_{p(x|z)} \mathbb{E}_{q(z)}[\mathrm{KL}(q(x|z), p(x|z))]\,, && (23)
\end{aligned}
$$

where the minimum is clearly when $p(x|z) = q(x|z)$ and the KL terms become 0. Thus, the gap can be reduced by maximizing the right hand side w.r.t. $p(x|z)$ and can be made tight if $p(x|z) = q(x|z)$. □

E.3 Proof of Theorem 1 (VAUB is an Upper Bound on GJSD)

Proof. The proof is straightforward using Theorem 6 from Cho et al. (2022) and Lemma 9 applied to each domain-conditional distribution $q(\boldsymbol{z}|d)$:

$$
\begin{aligned}
&\mathrm{GJSD}(\{q(\boldsymbol{z}|d)\}_{d=1}^{k}) && (24)\\
&\leq \min_{p(\boldsymbol{z})} \mathrm{H_c}(q(\boldsymbol{z}), p(\boldsymbol{z})) - \mathbb{E}_{q(d)}[\mathrm{H}(q(\boldsymbol{z}|d))] && \text{(Theorem 6)}\\
&\leq \min_{\substack{p(\boldsymbol{z})\\ p(\boldsymbol{x}|\boldsymbol{z},d)}} \mathrm{H_c}(q(\boldsymbol{z}), p(\boldsymbol{z})) - \mathbb{E}_{q(d)}\left[\mathrm{H}(q(\boldsymbol{x}|d)) + \mathbb{E}_{q(\boldsymbol{x}|d)q(\boldsymbol{z}|\boldsymbol{x},d)}\left[\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}\right]\right] && \text{(Lemma 9)}\\
&= \min_{\substack{p(\boldsymbol{z})\\ p(\boldsymbol{x}|\boldsymbol{z},d)}} \mathbb{E}_q[-\log p(\boldsymbol{z})] - \mathbb{E}_{q(d)}[\mathrm{H}(q(\boldsymbol{x}|d))] + \mathbb{E}_q\left[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}\right] && (25)\\
&= \min_{\substack{p(\boldsymbol{z})\\ p(\boldsymbol{x}|\boldsymbol{z},d)}} \mathbb{E}_q\left[-\log\left(\frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}\cdot p(\boldsymbol{z})\right)\right] - \mathbb{E}_{q(d)}[\mathrm{H}(q(\boldsymbol{x}|d))] && (26)\\
&\triangleq \mathrm{VAUB}(q(\boldsymbol{z}|\boldsymbol{x},d)), && (27)
\end{aligned}
$$

where the last two equals are just by definition of cross entropy and rearrangement of terms. From Theorem 6 and Lemma 9, we know that the bound gaps for both inequalities is:

$$
\mathrm{KL}(q(\boldsymbol{z}), p(\boldsymbol{z})) + \mathbb{E}_{q(d)q(\boldsymbol{z}|d)}[\mathrm{KL}(q(\boldsymbol{x}|\boldsymbol{z},d), p(\boldsymbol{x}|\boldsymbol{z},d))], \tag{28}
$$

where both KL terms can be made 0 by minimizing the VAUB over all possible $p(\boldsymbol{z})$ and $p(\boldsymbol{x}|\boldsymbol{z},d)$ respectively. □

E.4 Proof of Proposition 2 (Ratio of Decoder and Encoder Term Interpretation)

Proof. Given that the variational optimization is solved perfectly so that $p^*(\boldsymbol{x}|\boldsymbol{z},d) = q(\boldsymbol{x}|\boldsymbol{z},d)$, we can notice that this ratio has a simple form as the ratio of marginal densities:

$$
\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},d)}\Big[-\log\frac{p^*(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}\Big] = \mathbb{E}_q\Big[-\log\frac{q(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}\Big] = \mathbb{E}_q\Big[-\log\frac{q(\boldsymbol{x},\boldsymbol{z}|d)}{q(\boldsymbol{z}|d)}\frac{q(\boldsymbol{x}|d)}{q(\boldsymbol{x},\boldsymbol{z}|d)}\Big] = \mathbb{E}_q\Big[-\log\frac{q(\boldsymbol{x}|d)}{q(\boldsymbol{z}|d)}\Big], \tag{29}
$$

where the first equals is just by assumption that $p^*(\boldsymbol{x}|\boldsymbol{z},d)$ is optimized perfectly. Now if the encoder is an invertible and deterministic function, i.e., $q(\boldsymbol{z}|\boldsymbol{x},d)$ is a Dirac delta at $g(\boldsymbol{x}|d)$, then we can derive that the marginal ratio is simply the Jacobian in this special case:

$$
\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x},d)}\Big[-\log\frac{q(\boldsymbol{x}|d)}{q(\boldsymbol{z}|d)}\Big] = -\log\frac{q(\boldsymbol{x}|d)}{q(\boldsymbol{z}=g(\boldsymbol{x}|d)|d)} = -\log\frac{|J_g(\boldsymbol{x}|d)|q(\boldsymbol{z}=g(\boldsymbol{x}|d)|d)}{q(\boldsymbol{z}=g(\boldsymbol{x}|d)|d)} = -\log|J_g(\boldsymbol{x}|d)|, \tag{30}
$$

where the first equals is because the encoder $q(\boldsymbol{z}|\boldsymbol{x},d)$ is a Dirac delta function such that $\boldsymbol{z}=g(\boldsymbol{x}|d)$, and the second equals is by the change of variables formula. □

E.5 Proof that Mutual Information is Bounded by Reconstruction Term

We include a simple proof that the mutual information can be bounded by a probabilistic reconstruction term.

Proof. For any $\tilde{p}(\boldsymbol{x}|\boldsymbol{z},d)$, we know the following:

$$
\begin{aligned}
I(\boldsymbol{x},\boldsymbol{z}|d) &= \mathrm{H}(\boldsymbol{x}|d) - \mathrm{H}(\boldsymbol{x}|\boldsymbol{z},d) && (31)\\
&= \mathrm{H}(\boldsymbol{x}|d) - \mathbb{E}_q[-\log q(\boldsymbol{x}|\boldsymbol{z},d)] && (32)\\
&= \mathrm{H}(\boldsymbol{x}|d) + \mathbb{E}_q\Big[\log\frac{q(\boldsymbol{x}|\boldsymbol{z},d)\tilde{p}(\boldsymbol{x}|\boldsymbol{z},d)}{\tilde{p}(\boldsymbol{x}|\boldsymbol{z},d)}\Big] && (33)\\
&= \mathbb{E}_{q(\boldsymbol{x}|\boldsymbol{z})}[\log\tilde{p}(\boldsymbol{x}|\boldsymbol{z},d)] + \mathrm{KL}(q(\boldsymbol{x}|\boldsymbol{z},d), \tilde{p}(\boldsymbol{x}|\boldsymbol{z},d)) + \mathrm{H}(\boldsymbol{x}|d) && (34)\\
&\geq \mathbb{E}_{q(\boldsymbol{x}|\boldsymbol{z})}[\log\tilde{p}(\boldsymbol{x}|\boldsymbol{z},d)] + C, && (35)
\end{aligned}
$$

where $C \triangleq \mathrm{H}(\boldsymbol{x}|d)$, which is constant in the optimization. Therefore, we can optimize over all $\tilde{p}$ and still get a lower bound on mutual information:

$$I(\boldsymbol{x}, \boldsymbol{z}|d) \geq \max_{\tilde{p}(\boldsymbol{x}|\boldsymbol{z},d)} \mathbb{E}_{q(\boldsymbol{x}|\boldsymbol{z})}[\log \tilde{p}(\boldsymbol{x}|\boldsymbol{z}, d)] + C \tag{36}$$

where C in constant w.r.t. to the parameters of interest. □

E.6 Proof of Proposition 3 (Noisy JSD is a Statistical Divergence)

We prove a slightly more general version of noisy JSD here where the added Gaussian noise can come from a distribution over noise levels. While the Noisy JSD definition uses a single noise value, this can be generalized to an expectation over different noise scales as in the next definition.

Definition 5 (Noised-Smoothed JSD)**.** Given a distribution over noise variances p_σ that has support on the positive real numbers, the noise-smoothed JSD (NSJ) is defined as:

$$\mathrm{NSJ}(p, q) = \mathbb{E}_\sigma[\mathrm{NJSD}_\sigma(p, q)] = \mathbb{E}_\sigma[\mathrm{JSD}(\tilde{p}_\sigma, \tilde{q}_\sigma)]\,, \tag{37}$$

where $\tilde{p}_\sigma \triangleq p * \mathcal{N}(0, \sigma^2 I)$ and similarly for $\tilde{q}_\sigma$.

Note that NJSD is a special case of NSJ where p_σ is a Dirac delta distribution at a single σ value. Now we give the proof that NSJ (and thus NJSD) is a statistical divergence.

Proof. NSJ is non-negative because Eqn.6 (in main paper) is merely an expectation over JSDs, which are non-negative by the property of JSD. Now we prove the identity property for divergences, i.e., that $NSJ(p, q) = 0 \Leftrightarrow p = q$. If $p = q$, then it is simple to see that all the inner JSD terms will be 0 and thus $NSJ(p, q) = 0$. For the other direction, we note that if $NSJ(p, q) = 0$, we know that every NJSD term in the expectation in Eqn.6 (in main paper) must be 0, i.e., $\forall \sigma \in \mathrm{supp}(p_\sigma), NJSD(p, q) = 0$. Thus, we only need to prove for NJSD. For NJSD, we note that convolution with a Gaussian kernel $k \triangleq \mathcal{N}(0, \sigma^2 I)$ is invertible, and thus:

$$NJSD(p, q) = 0 \Rightarrow JSD(p * k, q * k) = 0 \tag{38}$$

$$\Rightarrow p * k = q * k \Rightarrow p = q\,. \tag{39}$$

□

E.7 Proof of Theorem 4 (Noisy AUB and Noisy VAUB Upper Bounds)

We would like to show that the noisy JSD can be upper bounded by a noisy version of AUB. Again, the key here is considering the latent entropy terms. So we provide one lemma and a corollary to setup the main proof.

Lemma 10 (Noisy entropy inequality)**.** *The entropy of a noisy random variable is greater than the entropy of its clean counterpart, i.e., if $\tilde{z} \triangleq z + \epsilon \sim q(\tilde{z})$ where $z \sim q(z)$ and ϵ are independent random variables, then $\mathrm{H}(q(\tilde{z})) \geq \mathrm{H}(q(z))$. (Proof are provided in the appendix)*

Proof.

$$\begin{aligned}
\mathrm{H}(z + \epsilon) &\geq \mathrm{H}(z + \epsilon|\epsilon) && \text{(Conditioning reduces entropy)} \\
&= \mathrm{H}(z|\epsilon) && \text{(Entropy is invariant under constant shift)} \\
&= \mathrm{H}(z)\,. && \text{(Independence of } z \text{ and } \epsilon\text{)}
\end{aligned}$$

□

Corollary 11 (Noisy entropy inequalities)**.** *Given a noisy random variable $\tilde{z} \triangleq z + \epsilon \sim q(\tilde{z})$, the following inequalities hold for deterministic invertible mappings g and stochastic mappings $q(z|x)$, respectively:*

$$\mathrm{H}(q(\tilde{z})) \geq \mathrm{H}(q(z)) \geq \mathrm{H}(q(x)) + \mathbb{E}_{q(x)}[\log |J_g(x)|] \tag{40}$$

$$\mathrm{H}(q(\tilde{z})) \geq \mathrm{H}(q(z)) \geq \mathrm{H}(q(x)) + \mathbb{E}_{q(x)q(z|x)}\left[\log \frac{p(x|z)}{q(z|x)}\right]\,. \tag{41}$$

Proof. Inequalities follow directly from Lemma 10 and the entropy inequalities in Lemma 7 and Lemma 9 respectively. □

Given these entropy inequalities, we now provide the proof that NAUB and NVAUB are upper bounds of the noisy JSD counterparts using the same techniques as (Cho et al., 2022) and the proof for VAUB above.

Proof. Proof of upper bound for noisy version of flow-based AUB:

$$\begin{aligned}
&\mathrm{NGJSD}(\{q(\boldsymbol{z}|d)\}_{d=1}^k;\sigma^2) && (42)\\
&\quad\equiv \mathrm{GJSD}(\{q(\tilde{\boldsymbol{z}}|d)\}_{d=1}^k) && (43)\\
&\quad\le \min_{p(\tilde{\boldsymbol{z}})\in\mathcal{P}_{\tilde{\boldsymbol{z}}}} \mathrm{H_c}(q(\tilde{\boldsymbol{z}}),p(\tilde{\boldsymbol{z}})) - \mathbb{E}_{q(d)}[\mathrm{H}(q(\tilde{\boldsymbol{z}}|d))] && (44)\\
&\quad\le \min_{p(\tilde{\boldsymbol{z}})\in\mathcal{P}_{\tilde{\boldsymbol{z}}}} \mathrm{H_c}(q(\tilde{\boldsymbol{z}}),p(\tilde{\boldsymbol{z}})) - \mathbb{E}_{q(d)}[\mathbb{E}_{q(x|d)}[\log|J_g(\boldsymbol{x}|d)|] + \mathrm{H}(q(\boldsymbol{x}|d))] && (45)\\
&\quad= \min_{p(\tilde{\boldsymbol{z}})\in\mathcal{P}_{\tilde{\boldsymbol{z}}}} \mathbb{E}_{q(\tilde{\boldsymbol{z}})}[-\log p(\tilde{\boldsymbol{z}})] - \mathbb{E}_{q(\boldsymbol{x},d)}[\log|J_g(\boldsymbol{x}|d)|] - \mathbb{E}_{q(d)}[\mathrm{H}(q(\boldsymbol{x}|d))] && (46)\\
&\quad= \min_{p(\tilde{\boldsymbol{z}})\in\mathcal{P}_{\tilde{\boldsymbol{z}}}} \mathbb{E}_{q(\boldsymbol{x},d)q(\epsilon;\sigma^2)}[-\log|J_g(\boldsymbol{x}|d)|p(g(\boldsymbol{x}|d)+\epsilon)] - \mathbb{E}_{q(d)}[\mathrm{H}(q(\boldsymbol{x}|d))] && (47)\\
&\quad\triangleq \mathrm{NAUB}(q(\boldsymbol{z}|\boldsymbol{x},d);\sigma^2)\,. && (48)
\end{aligned}$$

Proof of upper bound for noisy version of VAUB:

$$\begin{aligned}
&\mathrm{NGJSD}(\{q(\boldsymbol{z}|d)\}_{d=1}^k;\sigma^2) && (49)\\
&\quad\equiv \mathrm{GJSD}(\{q(\tilde{\boldsymbol{z}}|d)\}_{d=1}^k) && (50)\\
&\quad\le \min_{p(\tilde{\boldsymbol{z}})\in\mathcal{P}_{\tilde{\boldsymbol{z}}}} \mathrm{H_c}(q(\tilde{\boldsymbol{z}}),p(\tilde{\boldsymbol{z}})) - \mathbb{E}_{q(d)}[\mathrm{H}(q(\tilde{\boldsymbol{z}}|d))] && (51)\\
&\quad\le \min_{p(\tilde{\boldsymbol{z}})\in\mathcal{P}_{\tilde{\boldsymbol{z}}}} \mathrm{H_c}(q(\tilde{\boldsymbol{z}}),p(\tilde{\boldsymbol{z}})) - \mathbb{E}_{q(d)}\left[\max_{p(\boldsymbol{x}|\boldsymbol{z},d)\in\mathcal{P}_{\boldsymbol{x}|\boldsymbol{z},d}} \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z}|d)}\left[\log\frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}\right] + \mathrm{H}(q(\boldsymbol{x}|d))\right] && (52)\\
&\quad= \min_{\substack{p(\tilde{\boldsymbol{z}})\in\mathcal{P}_{\tilde{\boldsymbol{z}}}\\ p(\boldsymbol{x}|\boldsymbol{z},d)\in\mathcal{P}_{\boldsymbol{x}|\boldsymbol{z},d}}} \mathbb{E}_{q(\tilde{\boldsymbol{z}})}[-\log p(\tilde{\boldsymbol{z}})] - \mathbb{E}_{q(d)}\left[\mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z}|d)}\left[\log\frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}\right] + \mathrm{H}(q(\boldsymbol{x}|d))\right] && (53)\\
&\quad= \min_{\substack{p(\tilde{\boldsymbol{z}})\in\mathcal{P}_{\tilde{\boldsymbol{z}}}\\ p(\boldsymbol{x}|\boldsymbol{z},d)\in\mathcal{P}_{\boldsymbol{x}|\boldsymbol{z},d}}} \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z},d,\tilde{\boldsymbol{z}})}\left[-\log\left(\frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}\cdot p(\tilde{\boldsymbol{z}})\right)\right] - \mathbb{E}_{q(d)}[\mathrm{H}(q(\boldsymbol{x}|d))] && (54)\\
&\quad= \min_{\substack{p(\tilde{\boldsymbol{z}})\in\mathcal{P}_{\tilde{\boldsymbol{z}}}\\ p(\boldsymbol{x}|\boldsymbol{z},d)\in\mathcal{P}_{\boldsymbol{x}|\boldsymbol{z},d}}} \mathbb{E}_{q(\boldsymbol{x},\boldsymbol{z},d)q(\epsilon;\sigma^2)}\left[-\log\left(\frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}\cdot p(\boldsymbol{z}+\epsilon)\right)\right] - \mathbb{E}_{q(d)}[\mathrm{H}(q(\boldsymbol{x}|d))] && (55)\\
&\quad\triangleq \mathrm{NVAUB}(q(\boldsymbol{z}|\boldsymbol{x},d);\sigma^2)\,. && (56)
\end{aligned}$$

□

E.8 Proof of Proposition 5 (Fixed Prior is VAUB Plus Regularization Term)

We first remember the Fair VAE objective (Louizos et al., 2015) (note q is encoder distribution and p is decoder distribution and $q(\boldsymbol{z})$ is the marginal distribution of $q(\boldsymbol{x},d,\boldsymbol{z}) := q(\boldsymbol{x},d)q(\boldsymbol{z}|\boldsymbol{x},d)$):

$$\min_{q(\boldsymbol{z}|\boldsymbol{x},d)}\min_{p(\boldsymbol{x}|\boldsymbol{z},d)} \mathbb{E}_q\big[-\log\frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)}p_{\mathcal{N}(0,I)}(\boldsymbol{z})\big] \tag{57}$$

We show that this objective can be decomposed into an alignment objective and a prior regularization term if we assume the optimization class of prior distributions from the VAUB includes all possible distributions (and is solved theoretically). This gives a precise characterization of how the Fair VAE bound w.r.t. alignment and compares it to the VAUB alignment objective.

This decomposition exposes two insights. First, the Fair VAE objective is an alignment loss plus a regularization term. Thus, the *Fair VAE objective is sufficient for alignment but not necessary*—it adds an additional

constraint/regularization that is not necessary for alignment. Second, it reveals that by fixing the prior distribution, it can be viewed as perfectly solving the optimization over the prior for the alignment objective but requiring an unnecessary prior regularization. Finally, it should be noted that it is not possible in practice to compute the prior regularization term because $q(\boldsymbol{z})$ is not known. Therefore, this decomposition is only useful to understand the structure of he objective theoretically.

Proof of Proposition 5.

$$\min_{q(\boldsymbol{z}|\boldsymbol{x},d)} \min_{p(\boldsymbol{x}|\boldsymbol{z},d)} \mathbb{E}_q\big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)} p_{\mathcal{N}(0,I)}(\boldsymbol{z})\big] \qquad \text{(Fair VAE)}$$

$$= \min_{q(\boldsymbol{z}|\boldsymbol{x},d)} \min_{p(\boldsymbol{x}|\boldsymbol{z},d)} \mathbb{E}_q\big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)} \frac{p_{\mathcal{N}(0,I)}(\boldsymbol{z}) q(\boldsymbol{z})}{q(\boldsymbol{z})}\big] \qquad \text{(Inflate with true marginal } q(\boldsymbol{z}))$$

$$= \min_{q(\boldsymbol{z}|\boldsymbol{x},d)} \min_{p(\boldsymbol{x}|\boldsymbol{z},d)} \mathbb{E}_q\big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)} q(\boldsymbol{z})\big] + \mathbb{E}_q\big[-\log \frac{p_{\mathcal{N}(0,I)}(\boldsymbol{z})}{q(\boldsymbol{z})}\big] \qquad \text{(Rearrange)}$$

$$= \min_{q(\boldsymbol{z}|\boldsymbol{x},d)} \min_{p(\boldsymbol{x}|\boldsymbol{z},d)} \mathbb{E}_q\big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)} q(\boldsymbol{z})\big] + \mathbb{E}_q\big[\log \frac{q(\boldsymbol{z})}{p_{\mathcal{N}(0,I)}(\boldsymbol{z})}\big] \qquad \text{(Push negative inside)}$$

$$= \min_{q(\boldsymbol{z}|\boldsymbol{x},d)} \underbrace{\min_{p(\boldsymbol{x}|\boldsymbol{z},d)} \mathbb{E}_q\big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)} q(\boldsymbol{z})\big]}_{\text{VAUB Alignment with perfect prior optimization}} + \underbrace{\mathrm{KL}(q(\boldsymbol{z}), p_{\mathcal{N}(0,I)}(\boldsymbol{z}))}_{\text{Prior regularization}} \qquad \text{(Definition of KL)}$$

$$= \min_{q(\boldsymbol{z}|\boldsymbol{x},d)} \min_{p(\boldsymbol{x}|\boldsymbol{z},d)} \Big(\min_{p(\boldsymbol{z})} \mathbb{E}_q\big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)} p(\boldsymbol{z})\big]\Big) + \mathrm{KL}(q(\boldsymbol{z}), p_{\mathcal{N}(0,I)}(\boldsymbol{z}))$$

(Replace $q(\boldsymbol{z})$ with optimization over $p(\boldsymbol{z})$)

$$= \min_{q(\boldsymbol{z}|\boldsymbol{x},d)} \underbrace{\Big(\min_{p(\boldsymbol{x}|\boldsymbol{z},d)} \min_{p(\boldsymbol{z})} \mathbb{E}_q\big[-\log \frac{p(\boldsymbol{x}|\boldsymbol{z},d)}{q(\boldsymbol{z}|\boldsymbol{x},d)} p(\boldsymbol{z})\big]\Big)}_{\text{VAUB Alignment Objective}} + \underbrace{\mathrm{KL}(q(\boldsymbol{z}), p_{\mathcal{N}(0,I)}(\boldsymbol{z}))}_{\text{Prior Regularization}} \qquad \text{(Regroup to show structure)}$$

The last line is by noticing that $\mathrm{KL}(q(\boldsymbol{z}), p_{\mathcal{N}(0,I)}(\boldsymbol{z}))$ does not depend on $p(\boldsymbol{x}|\boldsymbol{z},d)$ or $p(\boldsymbol{z})$, i.e., it only depends on $q(\boldsymbol{z}|\boldsymbol{x},d)$ and the original data distribution $q(\boldsymbol{x},d)$. □

F Experimental Setup

All experiments were conducted on a computing setup with 24 processors, each having 12 cores running at 3.5GHz. Additionally, 2 NVIDIA RTX 3090 graphics cards were utilized when needed.

F.1 Simulated Experiments

F.1.1 Non-Matching Dimensions between Latent Space and Input Space

Dataset: We have two datasets, namely X_1 and X_2, each consisting of 500 samples. X_1 represents the original moon dataset, which has been perturbed by adding a noise scale of 0.05. X_2 is created by applying a transformation to the moon dataset. First, a rotation matrix of $\frac{3\pi}{8}$ is applied to the moons dataset which is generated using the same noise scale as in X_1. Then, scaling factors of 0.75 and 1.25 are applied independently to each dimension of the dataset. This results in a rotated-scaled version of the original moon dataset distribution.

Model: Encoders consist of three fully connected layers with hidden layer size as 20. Decoders are the reverse setup of the encoders. P_z is a learnable one-dimensional mixture of Gaussian distribution with 10 components and diagonal covariance matrix.

F.1.2 Noisy-AUB Helps Mitigate the Vanishing Gradient Problem

Dataset: X_1 Gaussian distribution with mean -20 and unit variance, X_2 Gaussian distribution with mean 20 and unit variance. Each dataset has 500 samples.

Model: Encoders consist of three fully connected layers with hidden layer size as 10. Decoders are the reverse setup of the encoders. P_z is a learnable one-dimensional mixture of Gaussian distribution with 2 components and

diagonal covariance matrix. The NVAUB has the added noise level of 10 while the VAUB has no added noise.

F.2 Comparison Between Other Non-adversarial Bounds

Dataset: We adopted the preprocessed Adult Dataset from Zhao et al. (2020), where the processed data has input dimensions 114 , targeted attribute as *income* and sensitive attribute as *gender*.

Model: Since all baseline models have only one encoder, we also adapt our model to have shared encoders. All models have encoder consists of three fully connected layers with hidden layer size as 84 and latent features as 60. For Moyer et al. (2018) and ours, decoders are the reverse setup of the encoder. For Gupta et al. (2021), we adapt the same network setup for the contrastive estimation model. Again, for Moyer et al. (2018) and Gupta et al. (2021) we used a fixed Gaussian distribution, and for our model, we use a learnable mixture of Gaussian distribution with 5 components and diagonal covariance matrix. For this experiment, we manually delete the classifier loss in all baseline models for the purpose of comparing only the bound performances.

Metric: For SWD, we randomly project 10^3 directions to one-dimensional vectors and compute the 1-Wasserstein distance between the projected vectors. Here is the table for the corresponding mean and standard deviation. The models are all significantly different (i.e., p-value is less than 0.01) when using an unpaired t-test on the 10^3 SWD values for each method.

Table 6: SWD for each method where * denotes statistically different at a 99% confidence level.

	Moyer	Gupta	VAUB
Sample Mean	9.71*	6.7*	5.64*
σ	0.54	0.74	0.64

For SVM, we first split the test dataset with 80% for training the SVM model and 20% for evaluating the SVM model. We use the scikit-learn package to grid search over the logspace of the C and γ parameters to choose the best hyperparameters in terms of accuracy.

F.3 Replacing Adversarial Losses

F.3.1 Replacing the Domain Adaption Objective

Model: We use the same encoder setup (referred as feature extraction layers) and the same classifier structure as in Ganin et al. (2016)).

F.3.2 Replacing the Fairness Representation Objective

Dataset: We adopted the preprocessed Adult Dataset from Zhao et al. (2020), where the processed data has input dimensions 114 , targeted attribute as *income* and sensitive attribute as *gender*.

Model: Since all baseline models have only one encoder, we also adapt our model to have a shared encoder. All models have encoder consists of three fully connected layers with hidden layer size as 84 and latent features as 60. For Moyer et al. (2018) and ours, decoders are the reverse setup of the encoder. For Gupta et al. (2021), we adapt the same network setup for the contrastive estimation model. Again, for Moyer et al. (2018) and Gupta et al. (2021) we used a fixed Gaussian distribution, and for our model, we use a learnable mixture of Gaussian distribution with 5 components and diagonal covariance matrix.