# INVERSE ENTROPIC OPTIMAL TRANSPORT SOLVES SEMI-SUPERVISED LEARNING VIA DATA LIKELIHOOD MAXIMIZATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Learning conditional distributions $\pi^*(\cdot|x)$ is a central problem in machine learning, which is typically approached via supervised methods with paired data $(x, y) \sim \pi^*$. However, acquiring paired data samples is often challenging, especially in problems such as domain translation. This necessitates the development of *semi-supervised* models that utilize both limited paired data and additional unpaired i.i.d. samples $x \sim \pi_x^*$ and $y \sim \pi_y^*$ from the marginal distributions. The usage of such combined data is complex and often relies on heuristic approaches. To tackle this issue, we propose a new learning paradigm that integrates both paired and unpaired data seamlessly using data likelihood maximization techniques. We demonstrate that our approach also connects intriguingly with inverse entropic optimal transport (OT). This finding allows us to apply recent advances in computational OT to establish an *end-to-end* learning algorithm to get $\pi^*(\cdot|x)$. In addition, we derive the universal approximation property, demonstrating that our approach can theoretically recover true conditional distributions with arbitrarily small error. Finally, we demonstrate through empirical tests that our method effectively learns conditional distributions using paired and unpaired data simultaneously.

## 1 INTRODUCTION

Recovering conditional distributions $\pi^*(y|x)$ from data is one of the fundamental problems in machine learning, which appears both in predictive and generative modeling. In predictive modeling, the standard examples of such tasks are the classification, where $x \in \mathbb{R}^{D_x}$ is a feature vector and $y \in \{0, 1, \dots, K\}$ is a class label, and regression, in which case $x$ is also a feature vector and $y \in \mathbb{R}$ is a real number. In generative modeling, both $x$ and $y$ are feature vectors in $\mathbb{R}^{D_x}, \mathbb{R}^{D_y}$, respectively, representing complex objects, and the goal is to find a transformation between them.

In our paper, we focus on the setting where both $x$ and $y$ are multi-dimensional real-valued vectors, and the true joint data distribution $\pi^*(x, y)$ is continuous over the space $\mathbb{R}^{D_x} \times \mathbb{R}^{D_y}$. This excludes scenarios where $y$ is a discrete variable, e.g., a class label. Our focus is on multi-dimensional probabilistic regression, often called *domain translation*, since $x$ and $y$ usually correspond to feature vectors from different domains. The goal is to perform probabilistic prediction: given a new input $x_{\text{new}}$ from the source domain, we aim to predict the corresponding output $y_{\text{new}}$ from the target domain, according to the conditional distribution $\pi^*(y|x_{\text{new}})$.

It is natural to assume that learning the conditional distribution $\pi^*(y|x)$ requires access to input–target data pairs $(x, y) \sim \pi^*$, where $\pi^*$ denotes the true joint distribution of the data. In such cases, $\pi^*(y|x)$ can be modeled using standard supervised learning approaches, ranging from simple regression to conditional generative models (Mirza & Osindero, 2014; Winkler et al., 2019; Ardizzone et al., 2019; Hagemann et al., 2024). However, acquiring paired data can be expensive or impractical, whereas obtaining unpaired samples – $x \sim \pi_x^*$ or $y \sim \pi_y^*$ – from each domain separately is often much easier and more cost-effective. This challenge has motivated the development of unsupervised (or unpaired) learning methods (e.g., (Zhu et al., 2017)), which aim to recover the dependency structure $\pi^*(y|x)$ using unpaired data alone.

While both paired (supervised) and unpaired (unsupervised) domain translation approaches are being extremely well developed nowadays, surprisingly, the semi-supervised setup when both paired

and unpaired data is available is much less explored. This is due to the challenge of designing learning objective (loss) which can simultaneously take into account both paired and unpaired data. A common approach involves heuristically combining standard paired and unpaired losses (cf. (Tripathy et al., 2019, §3.5), (Jin et al., 2019, §3.3), (Yang & Chen, 2020, §C), (Vasluianu et al., 2021, §3), (Panda et al., 2023, Eq. 8), (Tang et al., 2024, Eq. 8), (Theodoropoulos et al., 2024, §3.2), (Gu et al., 2023, §3)). However, as demonstrated in §5.1, these composite objectives fail to recover the true conditional distribution even in simple cases $D_x = D_y = 2$. This raises the question: *Can we design a simple loss to learn $\pi^*(y|x)$ that naturally integrates both paired and unpaired data?*

In our paper, we positively answer the above-raised question. Our **main contributions** are:

1. We introduce a novel loss function designed to facilitate the learning of conditional distributions $\pi^*(\cdot|x)$ using both paired and unpaired training samples drawn from $\pi^*$ (see §3.1). This loss function is grounded in the well-established principle of likelihood maximization. A key advantage of our approach is its ability to support end-to-end learning, thereby *seamlessly* integrating both paired and unpaired data into the training process.
2. We demonstrate the theoretical equivalence between our proposed loss function and the inverse entropic optimal transport problem (see §3.2). This finding enables us to leverage established computational optimal transport methods to address challenges in semi-supervised learning.
3. Building upon recent advancements in the field of computational optimal transport, we provide *end-to-end* algorithm exploiting the Gaussian mixture parameterization specifically tailored to optimize our proposed likelihood-based loss function (see §3.3). For completeness, Appendix A shows that our loss function is also applicable to a fully neural network parametrization.
4. We prove that our proposed parameterization satisfies the universal approximation property, which theoretically allows our algorithm to recover $\pi^*$ arbitrarily well (see §3.4).

Our empirical validation in §5 demonstrates the impact of both unpaired and paired data on overall performance. In particular, our findings show that the conditional distributions $\pi^*(\cdot|x)$ can be effectively learned even with a modest amount of paired data $(x, y) \sim \pi^*$, provided that sufficient auxiliary unpaired data $x \sim \pi_x^*$ and $y \sim \pi_y^*$ is available.

**Notations.** Throughout the paper, $\mathcal{X}$ and $\mathcal{Y}$ represent Euclidean spaces, equipped with the standard norm $\|\cdot\|$, induced by the inner product $\langle\cdot,\cdot\rangle$, i.e., $\mathcal{X} \stackrel{\text{def}}{=} \mathbb{R}^{D_x}$ and $\mathcal{Y} \stackrel{\text{def}}{=} \mathbb{R}^{D_y}$. The set of absolutely continuous probability distributions on $\mathcal{X}$ is denoted by $\mathcal{P}_{\text{ac}}(\mathcal{X})$. For simplicity, we use the same notation for both the distributions and their corresponding probability density functions. The joint probability distribution over $\mathcal{X} \times \mathcal{Y}$ is denoted by $\pi$ with corresponding marginals $\pi_x$ and $\pi_y$. The set of joint distributions with given marginals $\alpha$ and $\beta$ is represented by $\Pi(\alpha, \beta)$. We use $\pi(\cdot|x)$ for the conditional distribution, while $\pi(y|x)$ represents the conditional density at a specific point $y$. The differential entropy is given by $\mathrm{H}(\beta) = -\int_{\mathcal{Y}} \beta(y) \log \beta(y)\, dy$.

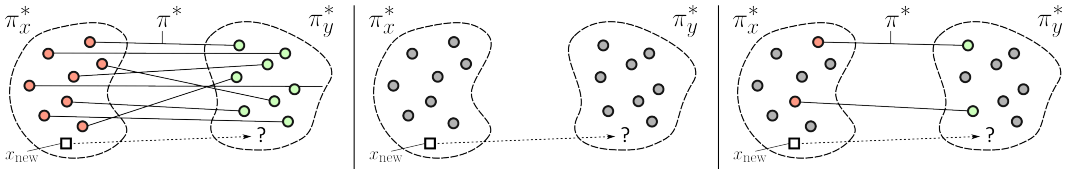
## 2 Background

First, we recall the formulation of the domain translation problem (§2.1). We remind the difference between its paired, unpaired, and semi-supervised setups. Next, we recall the basic concepts of the inverse entropic optimal transport, which are relevant to our paper (§2.2).

### 2.1 Domain Translation Problems

The goal of *domain translation* task is to transform data samples from the source domain to the target domain while maintaining the essential content or structure. This approach is widely used in applications like computer vision (Zhu et al., 2017; Lin et al., 2018; Peng et al., 2023), natural language processing (Jiang et al., 2021; Morishita et al., 2022), audio processing (Du et al., 2022), etc. Domain translation task setups can be classified into supervised (paired), unsupervised (unpaired), and semi-supervised approaches based on the data used for training (Figure 1).

**Supervised domain translation** relies on matched examples from both the source and target domains, where each input corresponds to a specific output, enabling direct supervision during the learning process. Formally, this setup assumes access to a set of $P$ empirical pairs $XY_{\text{paired}} \stackrel{\text{def}}{=} \{(x_1, y_1), \dots, (x_P, y_P)\} \sim \pi^*$ from some unknown joint distribution. The goal here is to recover the conditional distributions $\pi^*(\cdot|x)$ to generate samples $y|x_{\text{new}}$ for new inputs $x_{\text{new}}$ that are not




(a) *Supervised* (paired) domain translation setup.

(b) *Unsupervised* (unpaired) domain translation setup.

(c) *Semi-supervised* domain translation setup (**our focus**).

Figure 1: Visualization of domain translation setups. Red and green colors indicated paired training data $XY_{\text{paired}}$, while grey color indicates the unpaired training data $X_{\text{unpaired}}$, $Y_{\text{unpaired}}$.

present in the training data. While this task is relatively straightforward to solve, obtaining such paired training datasets can be challenging, as it often involves significant time, cost, and effort.

**Unsupervised domain translation**, in contrast, does not require direct correspondences between the source and target domains (Zhu et al., 2017, Figure 2). Instead, it involves learning to translate between domains using unpaired data, which offers greater flexibility but demands more advanced techniques to achieve accurate translation. Formally, we are given $Q$ unpaired samples $X_{\text{unpaired}} \stackrel{\text{def}}{=} \{x_1, \dots, x_Q\} \sim \pi_x^*$ from the source distribution, and $R$ unpaired samples $Y_{\text{unpaired}} \stackrel{\text{def}}{=} \{y_1, \dots, y_R\} \sim \pi_y^*$ from the target distribution. Our objective is to learn the conditional distributions $\pi^*(\cdot|x)$ of the unknown joint distribution $\pi^*$, whose marginals are $\pi_x^*$ and $\pi_y^*$, respectively. The unsupersvised setup is inherently ill-posed, often yielding ambiguous solutions (Moriakov et al., 2020). Accurate translation requires constraints and regularization (Yuan et al., 2018). Still, it is highly relevant due to the prevalence of unpaired data in practice.

**Semi-supervised domain translation** integrates both paired and unpaired data to enhance the translation process (Tripathy et al., 2019; Jiang et al., 2023a). This approach leverages the precision of paired data to guide the model while exploiting the abundance of unpaired data to improve performance and generalization. Formally, the setup assumes access to paired data $XY_{\text{paired}} \sim \pi^*$ as well as additional unpaired samples $X_{\text{unpaired}} \sim \pi_x^*$ and $Y_{\text{unpaired}} \sim \pi_y^*$. Note that paired samples can also be used in an unpaired manner. By convention, we assume $P \leq Q, R$, where the first $P$ unpaired samples are identical to the paired ones. The goal remains to learn the true conditional mapping $\pi^*(\cdot|x)$ using the available data. For extended discussion of real-world applications in which the semi-supervised setting arises naturally, see Appendix B.4.

### 2.2 Optimal Transport (OT)

The theoretical foundations of optimal transport are detailed in books (Villani et al., 2009; Santambrogio, 2015; Peyré et al., 2019). In what follows, we summarize the key concepts necessary to understand the connection between our loss function (§3.1) and inverse entropic optimal transport (Dupuy et al., 2019) established in §3.2. We emphasize that this section is intended solely to clarify this connection; it is not required for following the loss derivation itself, which is presented in a constructive manner to remain accessible to a broader audience. For a more detailed discussion of entropic, weak and inverse optimal transport, see Appendix B.1.

**Entropic OT** (Genevay, 2019). Given source and target distributions $\alpha \in \mathcal{P}_{\text{ac}}(\mathcal{X})$ and $\beta \in \mathcal{P}_{\text{ac}}(\mathcal{Y})$, and a cost function $c^* : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the *entropic* optimal transport (EOT) problem is defined as:

$$\text{OT}_{c^*,\varepsilon}(\alpha, \beta) \stackrel{\text{def}}{=} \min_{\pi \in \Pi(\alpha,\beta)} \mathbb{E}_{x,y\sim\pi}[c^*(x,y)] - \varepsilon \mathbb{E}_{x\sim\alpha} \text{H}(\pi(\cdot|x)), \tag{1}$$

where $\varepsilon > 0$ is the regularization parameter; setting $\varepsilon = 0$ recovers the classic OT formulation (Villani et al., 2009) originally proposed by (Kantorovich, 1942). Under mild assumptions, a unique minimizer $\pi^* \in \Pi(\alpha, \beta)$ exists and is known as the *entropic optimal transport plan*. We note that in the literature, the entropy regularization term in (1) is typically written as either $-\varepsilon \text{H}(\pi)$ or $+\varepsilon \text{KL}(\pi \| \alpha \otimes \beta)$. These formulations are equivalent up to additive constants; see the discussion in (Mokrov et al., 2024, §2) or (Gushchin et al., 2023b, §1). In this paper, we adopt the formulation in (1), which is also known as the *weak* form of entropic OT; see (Gozlan et al., 2017; Backhoff-Veraguas et al., 2019; Backhoff-Veraguas & Pammer, 2022).

**Semi-dual EOT**. Under mild assumptions on $c^*, \alpha, \beta$, the further semi-dual EOT formulation holds:

$$\mathrm{OT}_{c^*,\varepsilon}(\alpha,\beta) = \max_f \left\{ \mathbb{E}_{x\sim\alpha} f^{c^*}(x) + \mathbb{E}_{y\sim\beta} f(y) \right\}, \tag{2}$$

where $f$ ranges over a subset of continuous functions (dual potentials) subject to mild boundedness conditions; see (Backhoff-Veraguas & Pammer, 2022, Eq. 3.3) for details. The term $f^c$ denotes the so-called *weak entropic c-transform of f*, defined as:

$$f^{c^*}(x) \stackrel{\text{def}}{=} \min_{\mu\in\mathcal{P}(\mathcal{Y})} \left\{ \mathbb{E}_{y\sim\mu}[c^*(x,y)] - \varepsilon \mathrm{H}(\mu) - \mathbb{E}_{y\sim\mu} f(y) \right\}. \tag{3}$$

This transform admits a closed-form expression (Mokrov et al., 2024, Eq. 14):

$$f^c(x) = -\varepsilon \log \int_{\mathcal{Y}} \exp\left( \frac{f(y) - c(x,y)}{\varepsilon} \right) \mathrm{d}y. \tag{4}$$

**Inverse EOT.** The classical forward EOT problem (1) seeks an optimal transport plan $\pi^*$ between two given marginal distributions $\alpha$ and $\beta$ under a fixed cost function $c^*$. In contrast, the *inverse* EOT problem considers the reverse setting (Chan et al., 2025, §5.1): given a joint distribution $\pi^*$ with marginals $\pi_x^*$ and $\pi_y^*$, the goal is to recover a cost function $c^*$ such that $\pi^*$ is the EOT plan for $c^*$.

This inverse formulation is not uniquely defined in the literature – each version is typically tailored to specific applications (Stuart & Wolfram, 2020; Ma et al., 2020; Galichon & Salanié, 2022; Andrade et al., 2023). In this work, we adopt a version that aligns with our learning objective described in §3.1. This choice enables us, in §3.2, to formally relate our proposed loss to the inverse EOT framework. We further conjecture that this connection could potentially enable the application of advanced EOT solvers (e.g., diffusion Schrödinger bridges (Vargas et al., 2021; De Bortoli et al., 2021; Gushchin et al., 2023a; Shi et al., 2024; Gushchin et al., 2024b)) to enhance performance in semi-supervised learning scenarios, which we leave for future work.

With this motivation, we consider the *inverse* EOT problem as the following minimization problem:

$$c^* \in \arg\min_c \Bigg[ \underbrace{\mathbb{E}_{x,y\sim\pi^*}[c(x,y)] - \overbrace{\varepsilon \mathbb{E}_{x\sim\pi_x^*} \mathrm{H}\left(\pi^*(\cdot|x)\right)}^{\text{not depend on } c}}_{\geq \mathrm{OT}_{c,\varepsilon}\left(\pi_x^*,\pi_y^*\right)} - \mathrm{OT}_{c,\varepsilon}\left(\pi_x^*,\pi_y^*\right) \Bigg], \tag{5}$$

where $c : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ ranges over measurable cost functions. Consider the term $\mathrm{OT}_{c,\varepsilon}(\pi_x^*,\pi_y^*)$: due to entropic regularization, this expression admits a unique optimal transport plan $\pi_c^*$ for every quadruple $(c,\varepsilon,\pi_x^*,\pi_y^*)$. While $\pi_c^*$ matches the marginals of $\pi^*$, its internal structure – i.e., the conditional distributions – may differ. The term $\mathbb{E}_{x,y\sim\pi^*}[c(x,y)] - \varepsilon\mathbb{E}_{x\sim\pi_x^*}\mathrm{H}\left(\pi^*(\cdot|x)\right)$ represents the *transportation cost* of using $c$ to transport mass according to $\pi^*$ (cf. the minimization objective in (1)). If the "inner" part of $\pi_c^\star$ differs from that of $\pi^*$, this cost exceeds $\mathrm{OT}_{c,\varepsilon}(\pi_x^*,\pi_y^*)$. Therefore, the minimum of the full objective is achieved only when $\pi^*$ coincides with the optimal transport plan for some cost $c^*$, in which case the objective value is zero. Notably, the term $-\varepsilon\mathbb{E}_{x\sim\pi_x^*}\mathrm{H}\left(\pi^*(\cdot|x)\right)$ is independent of $c$ and can be omitted from the optimization. Additionally:

- **Invariance to $\varepsilon$.** Unlike the forward problem (1), the inverse problem is invariant to the entropic regularization parameter $\varepsilon > 0$. For any $\varepsilon' > 0$, the substitution $c(x,y) = \frac{\varepsilon}{\varepsilon'} c'(x,y)$ rescales the entire objective (5) by a constant, making solutions equivalent up to this change.
- **Multiple solutions.** The inverse problem (5) generally admits *many* valid cost functions. For instance, $c^*(x,y) = -\varepsilon\log\pi^*(x,y)$ achieves the minimum by construction. More generally, any function of the form $c'(x,y) = -\varepsilon\log\pi^*(x,y) + u(x) + v(y)$ is also valid, since additive terms depending only on $x$ or $y$ do not affect the resulting OT plan. In particular, setting $u(x) = \varepsilon\log\pi_x^*(x)$ and $v(y) = 0$ yields $c^*(x,y) = -\varepsilon\log\pi^*(y|x)$.

In practice, $\pi^*$ is known only through samples and not via its density. Therefore, closed-form expressions like $-\varepsilon\log\pi^*(x,y)$ or $-\varepsilon\log\pi^*(y|x)$ cannot be computed directly. This necessitates learning a parametric estimator $\pi^\theta$ to approximate the unknown conditional distributions.

## 3 Semi-supervised Domain Translation via Inverse EOT

In §3.1, we propose a novel loss function grounded in KL minimization. In §3.2, we demonstrate that proposed loss is equivalent to solving the inverse EOT problem (5), thereby connecting optimal

transport theory with our practical framework. To operationalize this approach, §3.3 introduces a lightweight parametrization. We subsequently prove in §3.4 that this parametrization, when combined with our loss minimization, guarantees arbitrarily accurate reconstruction of the true conditional plan under mild assumptions. Appendix A demonstrates how our framework extends to fully neural parametrization. All our proofs appear in Appendix E.

### 3.1 Loss Derivation

**Part I. Data likelihood maximization and its limitation.** Our goal is to approximate the true distribution $\pi^*$ by some parametric model $\pi^\theta$, where $\theta$ represents the parameters of the model. To achieve this, we would like to employ the standard KL-divergence minimization framework, also known as data likelihood maximization. Namely, we aim to minimize

$$\mathrm{KL}\left(\pi^*\|\pi^\theta\right) = \mathbb{E}_{x,y\sim\pi^*}\log\frac{\pi_x^*(x)\pi^*(y|x)}{\pi_x^\theta(x)\pi^\theta(y|x)} = \mathbb{E}_{x\sim\pi_x^*}\log\frac{\pi_x^*(x)}{\pi_x^\theta(x)} + \mathbb{E}_{x,y\sim\pi^*}\log\frac{\pi^*(y|x)}{\pi^\theta(y|x)} = \tag{6}$$

$$\mathrm{KL}\left(\pi_x^*\|\pi_x^\theta\right) + \mathbb{E}_{x\sim\pi_x^*}\mathbb{E}_{y\sim\pi^*(\cdot|x)}\log\frac{\pi^*(y|x)}{\pi^\theta(y|x)} = \underbrace{\mathrm{KL}\left(\pi_x^*\|\pi_x^\theta\right)}_{\text{Marginal}} + \underbrace{\mathbb{E}_{x\sim\pi_x^*}\mathrm{KL}\left(\pi^*(\cdot|x)\|\pi^\theta(\cdot|x)\right)}_{\text{Conditional}}. \tag{7}$$

It is clear that objective (7) splits into two independent components: the *marginal* and the *conditional* matching terms. Our focus will be on the conditional component $\pi^\theta(\cdot|x)$, as it is the necessary part for the domain translation. Note that the marginal part $\pi_x^\theta$ is not actually needed. The conditional part of (7) can further be divided into the following terms:

$$\mathbb{E}_{x\sim\pi_x^*}\mathbb{E}_{y\sim\pi^*(\cdot|x)}\left[\log\pi^*(y|x) - \log\pi^\theta(y|x)\right] = -\mathbb{E}_{x\sim\pi_x^*}\mathrm{H}\left(\pi^*(\cdot|x)\right) - \mathbb{E}_{x,y\sim\pi^*}\log\pi^\theta(y|x). \tag{8}$$

The first term is independent of $\theta$, so we obtain the following minimization objective:

$$\mathcal{L}(\theta) \stackrel{\text{def}}{=} -\mathbb{E}_{x,y\sim\pi^*}\log\pi^\theta(y|x). \tag{9}$$

It is important to note that minimizing (9) is equivalent to maximizing the conditional likelihood, a strategy utilized in conditional normalizing flows (Papamakarios et al., 2021, CondNF). However, a major limitation of this approach is its reliance solely on paired data from $\pi^*$, which can be difficult to obtain in real-world scenarios. In the following section, we modify this strategy to incorporate available unpaired data within a semi-supervised learning setup (see §2.1). We note that

**Part II. Solving the limitations via a tailored parameterization.** To address the above-mentioned issue and utilize unpaired data, we first use Gibbs-Boltzmann parametrization (LeCun et al., 2006):

$$\pi^\theta(y|x) \stackrel{\text{def}}{=} \frac{\exp\left(-E^\theta(y|x)\right)}{Z^\theta(x)}, \tag{10}$$

where $E^\theta(\cdot|x):\mathcal{Y}\to\mathbb{R}$ is *the Energy function*, and $Z^\theta(x) \stackrel{\text{def}}{=} \int_{\mathcal{Y}}\exp\left(-E^\theta(y|x)\right)\mathrm{d}y$ is the normalization constant. Substituting (10) into (9), we obtain:

$$\mathcal{L}(\theta) = \mathbb{E}_{x,y\sim\pi^*}E^\theta(y|x) + \mathbb{E}_{x\sim\pi_x^*}\log Z^\theta(x). \tag{11}$$

This objective already provides an opportunity to exploit the unpaired samples from the marginal distribution $\pi_x^*$ to learn the conditional distributions $\pi^\theta(\cdot|x)\approx\pi^*(\cdot|x)$. Namely, it helps to estimate the part of the objective related to the normalization constant $Z^\theta$. To incorporate independent samples from the second marginal distribution $\pi_y^*$, it is crucial to adopt a parametrization that separates the term in the energy function $E^\theta(y|x)$ that depends only on y. Thus, we propose:

$$E^\theta(y|x) \stackrel{\text{def}}{=} \frac{c^\theta(x,y) - f^\theta(y)}{\varepsilon}. \tag{12}$$

In fact, this parameterization allows us to decouple the cost function $c^\theta(x,y)$ and the potential function $f^\theta(y)$. Specifically, changes in $f^\theta(y)$ can be offset by corresponding changes in $c^\theta(x,y)$, resulting in the same energy function $E^\theta(y|x)$. For example, by setting $f^\theta(y)\equiv 0$ and $\varepsilon = 1$, the parameterization of the energy function $E^\theta(y|x)$ remains consistent, as it can be exclusively derived from $c^\theta(x,y)$. Substituting (12) into the energy term of (11), and using the identity $\mathbb{E}_{x,y\sim\pi^*}f^\theta(y) = \mathbb{E}_{y\sim\pi_y^*}f^\theta(y)$, yields *our final objective*, which integrates both paired and unpaired data:

$$\mathcal{L}(\theta) = \underbrace{\varepsilon^{-1}\mathbb{E}_{x,y\sim\pi^*}\left[c^\theta(x,y)\right]}_{\text{Joint, requires pairs } (x,y)\sim\pi^*} - \underbrace{\varepsilon^{-1}\mathbb{E}_{y\sim\pi_y^*}f^\theta(y)}_{\text{Marginal, requires } y\sim\pi_y^*} + \underbrace{\mathbb{E}_{x\sim\pi_x^*}\log Z^\theta(x)}_{\text{Marginal, requires } x\sim\pi_x^*} \to \min_\theta. \tag{13}$$

In Appendix E.1, we present a rigorous, step-by-step derivation starting from (6) and arriving at (13), using only *formal mathematical* transitions. Throughout this derivation, we initially assume that paired samples are drawn from the full joint distribution $\pi^*$. However, in practice the paired data may be restricted to a subset of $\pi^*$, which we discuss in detail in Appendix B.3.

At this point, a reader may come up with 2 reasonable questions regarding (13):

1. How to perform the optimization of the proposed objective? This question is not straightforward due to the existence of the (typically intractable) normalizing constant $Z_\theta$ in the objective.
2. To which extent do the separate terms in (13) (paired, unpaired data) contribute to the objective, and which type of data is the most important to get the correct solution?

We answer these questions in §3.3 and §5. Before doing that, we show a surprising finding that our proposed objective actually solves the inverse entropic OT problem (5).

### 3.2 Relation to Inverse EOT

We now show that (5) is equivalent to (13). Substituting the semi-dual formulation of EOT (2) into (5) (while omitting the constant entropy term) and using the identity $\min(-g) = -\max g$ gives:

$$\min_{c,f} \left\{ \mathbb{E}_{x,y\sim\pi^*}[c(x,y)] - \mathbb{E}_{x\sim\pi^*_x} f^c(x) - \mathbb{E}_{y\sim\pi^*_y} f(y) \right\}. \tag{14}$$

Assume that both the cost function $c$ and the potential function $f$ are parameterized as $c^\theta$ and $f^\theta$, respectively, with a parameter $\theta$. Using the definition from (4) and our energy function parameterization in (12), we can express $(f^\theta)^{c^\theta}(x)$ as $(f^\theta)^{c^\theta}(x) = -\varepsilon \log Z^\theta(x)$. This shows that the expression in (14) is equivalent to our proposed likelihood-based loss in (13), scaled by $\varepsilon$.

This result shows that *inverse entropic OT can be viewed as a likelihood maximization problem*, enabling the use of established techniques like ELBO and EM (Barber, 2012; Alemi et al., 2018; Bishop & Bishop, 2023). It also reframes inverse EOT as a semi-supervised domain translation task. Notably, prior work on inverse OT has largely focused on discrete, fully paired settings (see §4).

### 3.3 Practical Parameterization

The most computationally intensive aspect of optimizing the loss function in (13) lies in calculating the integral for the normalization constant $Z^\theta$. To tackle this challenge, we propose a lightweight parameterization that yields closed-form expressions for each term in the loss function. Our proposed cost function parameterization $c^\theta$ is based on the log-sum-exp function (Murphy, 2012, §3.5.3):

$$c^\theta(x,y) = -\varepsilon \log \sum_{m=1}^{M} v_m^\theta(x) \exp\left(\frac{\langle a_m^\theta(x), y\rangle}{\varepsilon}\right), \tag{15}$$

where $\{v_m^\theta(x) : \mathbb{R}^{D_x} \to \mathbb{R}_+, a_m^\theta(x) : \mathbb{R}^{D_x} \to \mathbb{R}^{D_y}\}_{m=1}^M$ are arbitrary parametric functions, e.g., *neural networks*, with learnable parameters denoted by $\theta_c$. The parametric form of the cost is motivated by (Korotin et al., 2024), from which we derived a more general functional form appropriate for our setting. Therefore, we adopt a Gaussian mixture parameterization for the dual potential $f^\theta$:

$$f^\theta(y) = \varepsilon \log \sum_{n=1}^{N} w_n^\theta \mathcal{N}(y \,|\, b_n^\theta, \varepsilon B_n^\theta), \tag{16}$$

where $\theta_f \stackrel{\text{def}}{=} \{w_n^\theta, b_n^\theta, B_n^\theta\}_{n=1}^N$ are learnable parameters of the potential, with $w_n^\theta \geq 0$, $b_n^\theta \in \mathbb{R}^{D_y}$, and $B_n^\theta \in \mathbb{R}^{D_y \times D_y}$ being a symmetric positive definite matrix. Thereby, our framework comprises a total of $\theta \stackrel{\text{def}}{=} \theta_f \cup \theta_c$ learnable parameters. For clarity and to avoid notation overload, we will omit the superscript $^\theta$ associated with learnable parameters and functions in the subsequent formulas.

**Proposition 3.1** (Tractable normalization constant)**.** *Our parametrization of the cost function* (15) *and dual potential* (16) *delivers* $Z^\theta(x) \stackrel{\text{def}}{=} \sum_{m=1}^M \sum_{n=1}^N z_{mn}(x)$*, where*

$$z_{mn}(x) \stackrel{\text{def}}{=} w_n v_m(x) \exp\left(\frac{a_m^\top(x) B_n a_m(x) + 2 b_n^\top a_m(x)}{2\varepsilon}\right).$$

The proposition offers a closed-form expression for $Z^\theta(x)$, which is essential for optimizing (13). Furthermore, the following proposition provides a method for sampling $y$ given a new sample $x_{\text{new}}$.

**Proposition 3.2** (Tractable conditional distributions)**.** *From our parametrization of the cost function* (15) *and dual potential* (16) *it follows that the* $\pi^\theta(\cdot|x)$ *are Gaussian mixtures:*

$$\pi^\theta(y|x)=\frac{1}{Z^\theta(x)}\sum_{m=1}^{M}\sum_{n=1}^{N} z_{mn}(x)\mathcal{N}(y\,|\,d_{mn}(x),\varepsilon B_n), \tag{17}$$

*where* $d_{mn}(x)\stackrel{\text{def}}{=} b_n + B_n a_m(x)$ *and* $z_{mn}(x)$ *defined in Proposition 3.1.*

TRAINING. As stated in §2.1, since we only have access to samples from the distributions, we minimize the empirical counterpart of (13) via the stochastic gradient descent w.r.t. $\theta$:

$$\mathcal{L}(\theta)\approx\widehat{\mathcal{L}}(\theta)\stackrel{\text{def}}{=}\varepsilon^{-1}\frac{1}{P}\sum_{p=1}^{P}c^\theta(x_p,y_p)-\varepsilon^{-1}\frac{1}{R}\sum_{r=1}^{R}f^\theta(y_r)+\frac{1}{Q}\sum_{q=1}^{Q}\log Z^\theta(x_q). \tag{18}$$

INFERENCE. According to our Proposition 3.2, the conditional distributions $\pi^\theta(\cdot|x)$ are Gaussian mixtures (17). As a result, sampling $y$ given $x$ is fast and straightforward.

### 3.4 UNIVERSAL APPROXIMATION OF THE PROPOSED PARAMETRIZATION

One may naturally wonder how expressive is our proposed parametrization of $\pi_\theta$ in §3.3. Below we show that this parametrization allows approximating any distribution $\pi^*$ that satisfies mild assumptions on boundness and regularity assumptions, see the details in Appendix E.4.

**Theorem 3.3** (Proposed parametrization guarantees universal conditional distributions)**.** *Under mild assumptions on the joint distribution* $\pi^*$*, for all* $\delta>0$ *there exists* ***(a)*** *an integer* $N>0$ *and a Gaussian mixture* $f^\theta$ (16) *with* $N$ *components,* ***(b)*** *an integer* $M>0$ *and cost* $c^\theta$(15) *defined by fully-connected neural networks* $a_m:\mathbb{R}^{D_x}\to\mathbb{R}^{D_y}, v_m:\mathbb{R}^{D_x}\to\mathbb{R}_+$ *with ReLU activations such that* $\pi^\theta$ *defined by* (10) *and* (12) *satisfies* $\text{KL}\left(\pi^*\|\pi^\theta\right)<\delta$.

## 4 RELATED WORKS

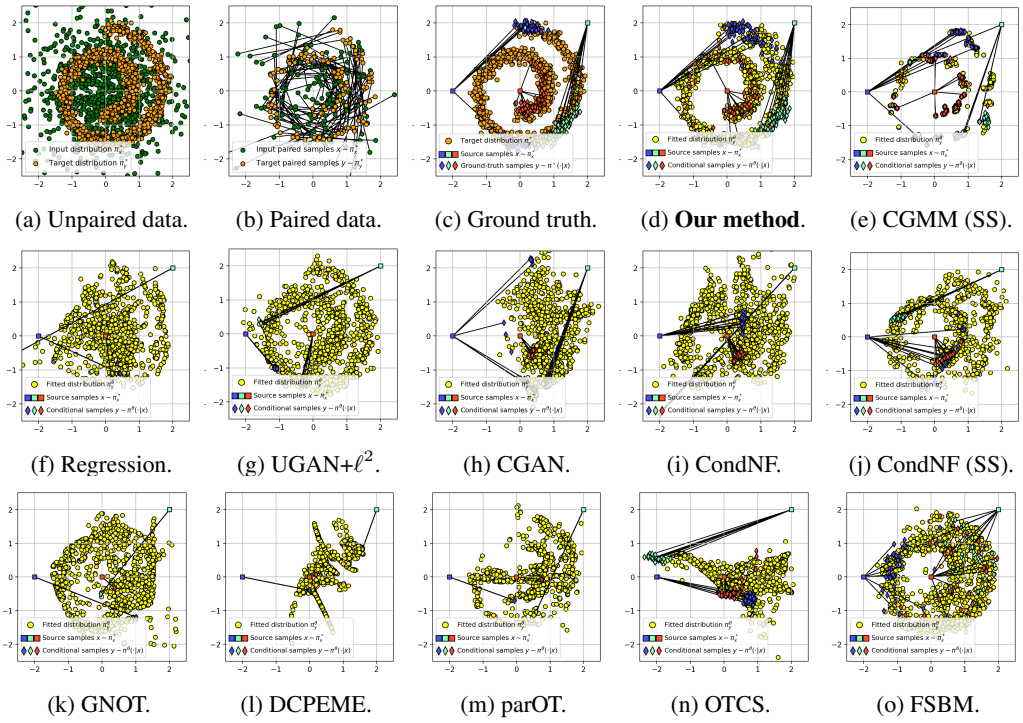
In this section, we briefly summarize the most relevant prior work; a more detailed discussion appears in Appendix B.5. Existing semi-supervised domain-translation approaches typically combine ad hoc objectives based on GAN losses and paired-data regularization (Chen et al., 2023; Panda et al., 2023), or use *keypoint-guided OT* (Gu et al., 2022), later extended to diffusion-based models (Gu et al., 2023; Theodoropoulos et al., 2024). Importantly, the paradigms outlined above do not offer any theoretical guarantees for reconstructing the conditional distribution $\pi^*(y|x)$, as they depend on heuristic loss constructions. We show that such approaches actually fail to recover the true plan even in toy 2-dimensional cases, refer to experiments in §5 for an illustrative example. **Inverse OT solvers:** works (Dupuy et al., 2019; Stuart & Wolfram, 2020) focuses on reconstructing cost functions (often in discrete settings), whereas our aim is to learn learn conditional distribution $\pi^\theta(\cdot|x)$. **Forward OT solvers:** Building on (Mokrov et al., 2024) and Gaussian-mixture parameterizations (Korotin et al., 2024; Gushchin et al., 2024a), our solver extends forward OT methods to general cost functions (Eq. (15)) and incorporates paired data through likelihood-based cost learning. Full details and additional discussion of **metric-learning** Cuturi & Avis (2014) provided in the Appendix B.5.

## 5 EXPERIMENTAL ILLUSTRATIONS

We evaluate our solver on synthetic data (§5.1), real-world data (§5.2), and on image-translation task (§5.3). The code is written using the PyTorch framework and will be made publicly available. It is provided in the supplemental materials. Experimental details are given in Appendix C and D.

### 5.1 GAUSSIAN TO SWISS ROLL MAPPING

**Setup.** For illustration, we adapt the experimental setup from (Korotin et al., 2024) to our purposes. We consider the task of learning conditional distributions from a Gaussian distribution $\pi_x^*$ to a Swiss Roll distribution $\pi_y^*$ (Figure 2a), guided by paired samples (Figure 2b) drawn from the ground-truth plan $\pi^*$. The ground-truth plan $\pi^*$ is obtained from a mini-batch OT plan after solving the *forward* OT problem with a specially designed cost that induces bi-modal conditionals $\pi^*(\cdot\mid x)$. Specifically,



(a) Unpaired data. (b) Paired data. (c) Ground truth. (d) **Our method**. (e) CGMM (SS).

(f) Regression. (g) UGAN+$\ell^2$. (h) CGAN. (i) CondNF. (j) CondNF (SS).

(k) GNOT. (l) DCPEME. (m) parOT. (n) OTCS. (o) FSBM.

Figure 2: Learned mapping on the *Gaussian* $\to$ *Swiss Roll* task for $P = 128$ and $Q = R = 1024$.

the cost matrix is defined as $C = \min(C^{+\varphi}, C^{-\varphi})$, where $C^{\pm\varphi}$ contains pairwise $\ell_2$ distances between $x$ and $-y^{\pm\varphi}$, with $-y^{\pm\varphi}$ denoting the vector $-y$ rotated by an angle of $\varphi = \pm 90°$. In other words, each $x \sim \pi_x^*$ is mapped to a point $y$ on the opposite side of the Swiss Roll, rotated either by $+\varphi$ or $-\varphi$ (Figure 2c). Further details on the paired data generation are provided in Appendix D.1. We evaluate each method's ability to capture these multi-modal conditional plans. During training, we use $P = 128$ paired and $Q = R = 1024$ unpaired samples, and in Appendix D.4 we analyze how varying the proportions of paired and unpaired data affects our method's performance.

**Baselines.** We evaluate our method against several baselines (see Appendix D.2 for details):

1. *Semi-supervised log-likelihood methods*: CondNF (SS) and CGMM (SS).
2. *Semi-supervised methods*: Neural OT with pair-guided cost (Asadulaev et al., 2024, GNOT, Appendix E), Differentiable Cost-Parameterized Entropic Mapping Estimator (Howard et al., 2024, DCPEME), (Panda et al., 2023, parOT), (Gu et al., 2023, OTCS), Feedback Schrödinger Bridge Matching (Theodoropoulos et al., 2024, FSBM).
3. *Standard generative & predictive models*: MLP regression with $\ell^2$ loss, Unconditional GAN with $\ell^2$ loss supplement (Goodfellow et al., 2014, UGAN+$\ell^2$), Conditional GAN (Mirza & Osindero, 2014, CGAN), Conditional Normalizing Flow (Winkler et al., 2019, CondNF).

Note that some baselines can fully utilize both paired and unpaired data during training, while others rely solely on paired data. Refer to Table 5 for specifics on data usage.

**Discussion.** The results of the aforementioned methods are depicted in Figure 2. Clearly, the Regression model simply predicts the conditional mean $\mathbb{E}_{y\sim\pi^*(\cdot|x)}y$, failing to capture the full distribution. The CGAN is unable to accurately learn the target distribution $\pi_y^*$, while the UGAN+$\ell^2$ fails to capture the underlying conditional distribution, resulting in suboptimal performance. The CondNF model suffers from overfitting, likely due to the limited availability of paired data $XY_{\text{paired}}$. Methods GNOT, DCPEME, parOT learn deterministic mapping and therefore are unable to capture the conditional distribution. Similar to parOT, both OTCS and FSBM build on the idea of key-points but are designed for stochastic setup. However, these methods fail to capture bi-modal conditional mappings, presumably due to a biased objective introduced by the artificial cost function that enforces alignment with key-points. The CondNF (SS) does not provide improvement compared to CondNF, and CGMM (SS) model learns a degenerate solution, which is presumably due to the overfitting. As a sanity check, we evaluate all baselines using a large amount of paired data. Details are given in Appendix D.3. In fact, even in this case, almost all the methods fail to learn true $\pi^*(\cdot|x)$.

### 5.2 WEATHER PREDICTION

Here we aim to evaluate our proposed approach on real-world data. We consider the *weather prediction* dataset (Malinin et al., 2021; Rubachev et al., 2024). The data is collected from weather stations across the world and weather forecast physical models. It consists of 94 meteorological features, e.g., pressure, wind, humidity, etc., which are measured over a period of one year at different spatial locations.

| # Paired \ # Unpaired | Baseline | Ours | | | | | |
|---|---|---|---|---|---|---|---|
| | 0 | 5 | 10 | 50 | 100 | 250 | 500 |
| 5 | diverged | 9.4 ±.1 | 14.2 ±1.7 | 15.47 ±.02 | 16.6 ±.0 | 17.91 ±.07 | 9.40 ±.03 |
| 10 | 0.4 ±.2 | 9.48 ±.02 | 17.9 ±.3 | 18.5 ±.4 | 18.4 ±.2 | 18.8 ±.2 | 19.2 ±.3 |
| 25 | 3.5 ±.09 | 9.40 ±.03 | 18.3 ±.06 | 18.7 ±.2 | 18.8 ±.07 | 19.5 ±.1 | 19.8 ±.1 |
| 50 | 6.4 ±.05 | 9.47 ±.01 | 18.7 ±.2 | 18.9 ±.04 | 19.2 ±.2 | 19.8 ±.03 | 20.3 ±.4 |
| 90 | 6.5 ±.1 | 9.30 ±.05 | 19 ±.01 | 19.4 ±.05 | 19.4 ±.2 | 20.3 ±.05 | **20.5** ±.09 |

Table 1: The values of the test *log-likelihood* ↑ on the *weather prediction* dataset obtained for a different number of paired and unpaired training samples.

**Setup.** Initially, the problem was formulated as the prediction and uncertainty estimation of the air temperature at a specific time and location. We expand this task to the probabilistic prediction of all meteorological features, thereby reducing reliance on measurement equipment in remote and difficult-to-access locations, e.g. the Polar regions (see Appendix C.3).

**Metrics and baselines**. We evaluate the performance of our approach by calculating the *log-likelihood* (LL) on the test target features. A natural baseline for this task is a probabilistic model that maximizes the likelihood of the target data. Thus, we implement an MLP that learns to predict the parameters of a mixture of Gaussians and is trained on the paired data only via the log-likelihood optimization (9). We also compare with semi-supervised log-likelihood methods CGMM (SS) and CondNF (SS). For completeness, we also add standard generative models. These models are trained using the available paired and unpaired data. Note that GAN models do not provide the density estimation and log-likelihood can not be computed for them. Therefore, we report Conditional Fréchet Distance (CFD): for each test $x$, we compute the Fréchet distance (Heusel et al., 2017, Eq. 6) between predicted and true features $y$, then average over all test inputs.

**Discussion.** Tables 1 and 2 summarize our findings. From Table 1, the main observation is that even a small amount of unpaired data leads to substantial performance gains, underscoring the effectiveness of our semi-supervised formulation. Furthermore, Table 2 shows that our method also yields samples that better match the true conditional distributions compared to competing approaches. For more detailed discussion regarding low-data regimes, see Appendix C.3.

| | Ours | CGAN | UGAN+$\ell^2$ | CondNF | Regression | CGMM (SS) | CondNF (SS) |
|---|---|---|---|---|---|---|---|
| LL↑ | **20.5** ±.09 | N/A | N/A | 1.29 ±.03 | N/A | 0.32 ±.03 | 0.52 ±.02 |
| CFD↓ | **7.21** ±.04 | 15.79 ±1.11 | 15.44 ±1.89 | 18.72 ±.09 | 8.29 ±.04 | **7.17** ±.07 | 28.5 ±.5 |

Table 2: The values of the test *Log-Likelihood* (LL) and *Conditional Fréchet distance* (CFD) on the *weather prediction* dataset of our approach and baselines (500 unpaired and 90 paired samples).

### 5.3 IMAGE TRANSLATION VIA ALAE

**Setup.** In this section, following the setup from (Theodoropoulos et al., 2024), we demonstrate our method capabilities for image translation in latent space of dimension 512 of ALAE encoder (Pidhorskyi et al., 2020) for 1024×1024 FFHQ dataset (Karras et al., 2019). Similarly, we generate 2K paired samples using (Korotin et al., 2024) and performed semi-supervised Woman-to-Man translation.

| Method | FID ↓ | SSIM ↑ | LPIPS ↓ |
|---|---|---|---|
| FSBM | $10.2 \pm 0.6$ | $0.5237 \pm 0.0005$ | $0.5625 \pm 0.0003$ |
| Ours | $\mathbf{9.3 \pm 0.1}$ | $\mathbf{0.5315 \pm 0.0002}$ | $\mathbf{0.5531 \pm 0.0006}$ |

Table 3: Metrics for Woman-to-Man translation described in §5.3.

**Discussion.** Visual results are shown in Figure 3, and quantitative test metrics computed against the target domain, averaged over three trainings with different seeds and rounded to the first significant digit (LPIPS (Zhang et al., 2018), FID (Heusel et al., 2017), SSIM (Wang et al., 2004)), are reported in Table 3. Additional examples are provided in Appendix C.4 Our method achieves comparable performance, while requiring only 3 minutes of training on an A100 GPU, compared to 5 hours for FSBM on the same hardware. Implementation and experimental details, refer to Appendix C.4

## 6 DISCUSSION

**Contributions & Potential impact.** Our framework offers a simple, non-minimax objective that naturally integrates both paired and unpaired data. We expect that these advantages, together with the connection to entropic optimal transport (EOT), will encourage adoption in more advanced semi-supervised methods, including approaches based on diffusion Schrödinger bridges (Vargas et al., 2021; De Bortoli et al., 2021; Shi et al., 2024) and flow matching (Chen et al., 2025; Balcerak et al., 2025). Moreover, this paper aims to advance the field of semi-supervised learning for domain translation, with a primary focus on the continuous target case $y \in \mathbb{R}^{D_y}$. In Appendix B.2, we discuss the potential extension of our loss to discrete targets $y \in \mathbb{K}^{D_y}$, where $\mathbb{K} = \{1, \dots, K\}$ represents a set of categories – an interesting direction for future work.

**Limitations & Future Work.** A limitation of our method is its reliance on Gaussian Mixture parameterization (§3.3), which may affect scalability. To address this, we provide a proof of concept for fully neural parameterizations of the cost and potential functions below, with a more detailed discussion in Appendix A. These parameterizations can be integrated into our loss via energy-based modeling (EBM) (Song & Kingma, 2021) and could, in principle, scale to large image domains (Schröder et al., 2023; Yu et al., 2023; Zhu et al., 2024). A full investigation of such large-scale applications, however, lies beyond the scope of our methodological work.


Figure 3: Comparison of our method and FSBM on the Woman-to-Man translation task described in §5.3.

As we discussed in §3.3, a key advantage of the proposed parametrization is that the normalizing constant $Z_\theta$ in (13) is available in closed form. In contrast, general parameterizations of $c^\theta$ and $f^\theta$ lack this property, requiring more advanced sampling techniques (Andrieu et al., 2003). While the objective (13) itself may be intractable, we can derive its gradient, which is essential for optimization. Proposition A.1 provides the gradient computation, enabling practical gradient-based training This motivates the procedure outlined in Algorithm 1, where the conditional distribution is modeled as $\pi^\theta(y|x) \propto \exp\left(\frac{f^\theta(y) - c^\theta(x,y)}{\varepsilon}\right)$.

**Experimental Setup.** To illustrate the scalability of our approach, we adapted an experiment from (Mokrov et al., 2024) using the colored MNIST dataset (Arjovsky et al., 2019). While the original task involved translating digit 2 into digit 3 using unpaired images, we modified the setup to demonstrate our method's ability to perform translations according to paired data. Namely, we created pairs by shifting the hue (Joblove & Greenberg, 1978) of the source images by 120°. Specifically, for a source image with a hue $h$ in the range $0° \le h < 360°$, the target image's hue was set to $(h + 120°)$ mod 360°. For implementation details, see Appendix A.3.

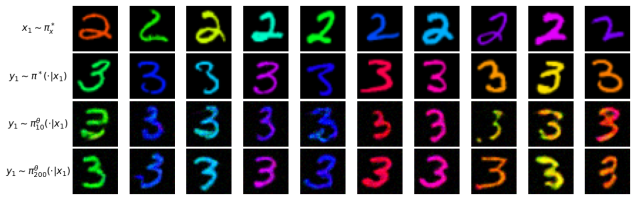



Figure 4: Performance of our Algorithm 1 on the colored MNIST (§6). Rows: source images, target images with ground-truth colors, results for $P = 10$ and $P = 200$.

**Results.** The results of this experiment are shown in Figure 4. Notably, our method successfully learned the color transformation using only 10 pairs (third row). Increasing the number of pairs to 200 further improved the quality of the translation (forth row).

**LLM Usage.** Large Language Models (LLMs) were employed solely to help rephrase sentences and enhance text clarity. All scientific content, results, and interpretations presented in this paper were developed entirely by the authors.

## REFERENCES


Beatrice Acciaio, Anastasis Kratsios, and Gudmund Pammer. Designing universal causal deep learning models: The geometric (hyper) transformer. Mathematical Finance, 34(2):671–735, 2024.

Abien Fred Agarap. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375, 2018.

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. In International conference on machine learning, pp. 159–168. PMLR, 2018.

Francisco Andrade, Gabriel Peyré, and Clarice Poon. Sparsistency for inverse optimal transport. arXiv preprint arXiv:2310.05461, 2023.

Francisco Andrade, Gabriel Peyré, and Clarice Poon. Learning from samples: Inverse problems over measures via sharpened fenchel-young losses. arXiv preprint arXiv:2505.07124, 2025.

Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. Machine learning, 50:5–43, 2003.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24). ACM, April 2024. doi: 10.1145/3620665.3640366. URL https://pytorch.org/assets/pytorch2-2.pdf.

Reza Arabpour, John Armstrong, Luca Galimberti, Anastasis Kratsios, and Giulia Livieri. Low-dimensional approximations of the conditional law of volterra processes: a non-positive curvature approach. arXiv preprint arXiv:2405.20094, 2024.

Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. arXiv preprint arXiv:1907.02392, 2019.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. arXiv preprint arXiv:1907.02893, 2019.

Arip Asadulaev, Alexander Korotin, Vage Egiazarian, Petr Mokrov, and Evgeny Burnaev. Neural optimal transport with general cost functionals. In The Twelfth International Conference on Learning Representations, 2024.

Andrei Atanov, Alexandra Volokhova, Arsenii Ashukha, Ivan Sosnovik, and Dmitry Vetrov. Semi-conditional normalizing flows for semi-supervised learning. arXiv preprint arXiv:1905.00505, 2019.

Janis Auffenberg, Jonas Bresch, Oleh Melnyk, and Gabriele Steidl. Unsupervised ground metric learning. arXiv preprint arXiv:2507.13094, 2025.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. Advances in neural information processing systems, 34:17981–17993, 2021.

Julio Backhoff-Veraguas and Gudmund Pammer. Applications of weak transport theory. Bernoulli, 28(1):370–394, 2022.

Julio Backhoff-Veraguas, Mathias Beiglböck, and Gudmun Pammer. Existence, duality, and cyclical monotonicity for weak transport costs. Calculus of Variations and Partial Differential Equations, 58(6):203, 2019.

Michal Balcerak, Tamaz Amiranashvili, Suprosanna Shit, Antonio Terpin, Sebastian Kaltenbach, Petros Koumoutsakos, and Bjoern Menze. Energy matching: Unifying flow matching and energy-based models for generative modeling. arXiv preprint arXiv:2504.10612, 2025.

David Barber. Bayesian reasoning and machine learning. Cambridge University Press, 2012.

Christopher M Bishop and Hugh Bishop. Deep learning: Foundations and concepts. Springer Nature, 2023.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. Advances in Neural Information Processing Systems, 35:28266–28279, 2022.

Davide Carbone. Hitchhiker's guide on energy-based models: a comprehensive review on the relation with other generative models, sampling and statistical physics. arXiv preprint arXiv:2406.13661, 2024.

Davide Carbone, Mengjian Hua, Simon Coste, and Eric Vanden-Eijnden. Efficient training of energy-based models using jarzynski equality. Advances in Neural Information Processing Systems, 36:52583–52614, 2023.

Timothy CY Chan, Rafid Mahmood, and Ian Yihang Zhu. Inverse optimization: Theory and applications. Operations Research, 73(2):1046–1074, 2025.

Chaofeng Chen, Wei Liu, Xiao Tan, and Kwan-Yee K Wong. Semi-supervised cycle-gan for face photo-sketch translation in the wild. Computer Vision and Image Understanding, 235:103775, 2023.

Hansheng Chen, Kai Zhang, Hao Tan, Zexiang Xu, Fujun Luan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. Gaussian mixture flow matching models. arXiv preprint arXiv:2504.05304, 2025.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. Mathematics of computation, 87(314):2563–2609, 2018.

Tong Cui, Qingyue Dai, Meng Zhang, Kairu Li, and Xiaofei Ji. Scl-dehaze: Toward real-world image dehazing via semi-supervised codebook learning. Electronics, 13(19):3826, 2024.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems, 26, 2013.

Marco Cuturi and David Avis. Ground metric learning. The Journal of Machine Learning Research, 15(1):533–564, 2014.

Marco Cuturi, Michal Klein, and Pierre Ablin. Monge, bregman and occam: Interpretable optimal transport in high-dimensions with feature-sparse maps. In International Conference on Machine Learning, pp. 6671–6682. PMLR, 2023.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. Advances in Neural Information Processing Systems, 34:17695–17709, 2021.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In International Conference on Learning Representations, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Yichao Du, Weizhi Wang, Zhirui Zhang, Boxing Chen, Tong Xu, Jun Xie, and Enhong Chen. Non-parametric domain adaptation for end-to-end speech translation. In Conference on Empirical Methods in Natural Language Processing (EMNLP), 2022.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. Advances in Neural Information Processing Systems, 32, 2019.

Yilun Du, Shuang Li, B. Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy based models. In Proceedings of the 38th International Conference on Machine Learning (ICML-21), 2021.

Arnaud Dupuy, Alfred Galichon, and Yifei Sun. Estimating matching affinity matrices under low-rank constraints. Information and Inference: A Journal of the IMA, 8(4):677–689, 2019.

William Falcon, Nicki Skafte, Justus Schock, et al. Torchmetrics: Machine learning metrics for pytorch, 2020. URL `https://github.com/Lightning-AI/metrics`. Version: latest.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. Journal of Machine Learning Research, 22(78):1–8, 2021.

Alfred Galichon and Bernard Salanié. Cupid's invisible hand: Social surplus and identification in matching models. The Review of Economic Studies, 89(5):2600–2629, 2022.

Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P Kingma. Learning energy-based models by diffusion recovery likelihood. In International Conference on Learning Representations, 2021.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. Advances in Neural Information Processing Systems, 37:133345–133385, 2024.

Aude Genevay. Entropy-regularized optimal transport for machine learning. PhD thesis, Université Paris sciences et lettres, 2019.

Aude Genevay, Lénaic Chizat, Francis Bach, Marco Cuturi, and Gabriel Peyré. Sample complexity of sinkhorn divergences. In The 22nd international conference on artificial intelligence and statistics, pp. 1574–1583. PMLR, 2019.

Cong Geng, Tian Han, Peng-Tao Jiang, Hao Zhang, Jinwei Chen, Søren Hauberg, and Bo Li. Improving adversarial energy-based model via diffusion process. In Forty-first International Conference on Machine Learning, 2024.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, 27, 2014.

Nathael Gozlan, Cyril Roberto, Paul-Marie Samson, and Prasad Tetali. Kantorovich duality for general transport costs and applications. Journal of Functional Analysis, 273(11):3327–3405, 2017.

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, Bernhard Schölkopf, et al. Covariate shift by kernel mean matching. Dataset shift in machine learning, 3 (4):5, 2009.

Xiang Gu, Yucheng Yang, Wei Zeng, Jian Sun, and Zongben Xu. Keypoint-guided optimal transport with applications in heterogeneous domain adaptation. Advances in Neural Information Processing Systems, 35:14972–14985, 2022.

Xiang Gu, Liwei Yang, Jian Sun, and Zongben Xu. Optimal transport-guided conditional score-based diffusion model. Advances in Neural Information Processing Systems, 36:36540–36552, 2023.

Nikita Gushchin, Alexander Kolesov, Alexander Korotin, Dmitry P Vetrov, and Evgeny Burnaev. Entropic neural optimal transport via diffusion processes. Advances in Neural Information Processing Systems, 36:75517–75544, 2023a.

Nikita Gushchin, Alexander Kolesov, Petr Mokrov, Polina Karpikova, Andrei Spiridonov, Evgeny Burnaev, and Alexander Korotin. Building the bridge of schrödinger: A continuous entropic optimal transport benchmark. Advances in Neural Information Processing Systems, 36:18932–18963, 2023b.

Nikita Gushchin, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Light and optimal schrödinger bridge matching. In Forty-first International Conference on Machine Learning, 2024a.

Nikita Gushchin, Daniil Selikhanovych, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Adversarial schrödinger bridge matching. In The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024b. URL https://openreview.net/forum?id=L3Knnigicu.

Paul Hagemann, Johannes Hertrich, Fabian Altekrüger, Robert Beinert, Jannis Chemseddine, and Gabriele Steidl. Posterior sampling based on gradient flows of the MMD with negative distance kernel. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=YrXHEb2qMb.

Matthieu Heitz, Nicolas Bonneel, David Coeurjolly, Marco Cuturi, and Gabriel Peyré. Ground metric learning on graphs. Journal of Mathematical Imaging and Vision, 63(1):89–107, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30, 2017.

Peter Holderrieth, Michael Samuel Albergo, and Tommi Jaakkola. LEAPS: A discrete neural sampler via locally equivariant networks. In Forty-second International Conference on Machine Learning, 2025. URL https://openreview.net/forum?id=Hq2RniQAET.

Qianwen Hou, Shilong Wang, and Jianlei Liu. Semi-supervised dehazing method based on image enhancement and multi-negative contrastive auxiliary learning. International Journal of Machine Learning and Cybernetics, pp. 1–14, 2025.

Samuel Howard, George Deligiannidis, Patrick Rebeschini, and James Thornton. Differentiable cost-parameterized monge map estimators. arXiv preprint arXiv:2406.08399, 2024.

Geert-Jan Huizing, Laura Cantini, and Gabriel Peyré. Unsupervised ground metric learning using wasserstein singular vectors. In International Conference on Machine Learning, pp. 9429–9443. PMLR, 2022.

Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-supervised learning with normalizing flows. In International conference on machine learning, pp. 4615–4630. PMLR, 2020.

Pratik Jawanpuria, Dai Shi, Bamdev Mishra, and Junbin Gao. A riemannian approach to ground metric learning for optimal transport. In ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE, 2025.

Qingnan Jiang, Mingxuan Wang, Jun Cao, Shanbo Cheng, Shujian Huang, and Lei Li. Learning kernel-smoothed machine translation with retrieved examples. In Conference on Empirical Methods in Natural Language Processing (EMNLP), 2021.

Yuxin Jiang, Liming Jiang, Shuai Yang, and Chen Change Loy. Scenimefy: Learning to craft anime scene via semi-supervised image-to-image translation. In IEEE International Conference on Computer Vision (ICCV), 2023a.

Yuxin Jiang, Liming Jiang, Shuai Yang, and Chen Change Loy. Scenimefy: Learning to craft anime scene via semi-supervised image-to-image translation. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 7357–7367, 2023b.

Cheng-Bin Jin, Hakil Kim, Mingjie Liu, Wonmo Jung, Seongsu Joo, Eunsik Park, Young Saem Ahn, In Ho Han, Jae Il Lee, and Xuenan Cui. Deep ct to mr synthesis using paired and unpaired data. Sensors, 19(10):2361, 2019.

George H Joblove and Donald Greenberg. Color spaces for computer graphics. In Proceedings of the 5th annual conference on Computer graphics and interactive techniques, pp. 20–25, 1978.

Leonid V Kantorovich. On the translocation of masses. In Dokl. Akad. Nauk. USSR (NS), volume 37, pp. 199–201, 1942.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4401–4410, 2019.

Tanguy Kerdoncuff, Rémi Emonet, and Marc Sebban. Metric learning in optimal transport for domain adaptation. In International joint conference on artificial intelligence, pp. 2162–2168. IJCAI, 2020.

Diederik P Kingma. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

Alexander Korotin, Nikita Gushchin, and Evgeny Burnaev. Light schrödinger bridge. In The Twelfth International Conference on Learning Representations, 2024.

Grigoriy Ksenofontov and Alexander Korotin. Categorical schrödinger bridge matching. In Forty-second International Conference on Machine Learning, 2025. URL https://openreview.net/forum?id=RBly0nOr2h.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. Predicting structured data, 1(0), 2006.

Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport. Discrete & Continuous Dynamical Systems-A, 34(4):1533–1574, 2014.

Lerenhan Li, Yunlong Dong, Wenqi Ren, Jinshan Pan, Changxin Gao, Nong Sang, and Ming-Hsuan Yang. Semi-supervised image dehazing. IEEE Transactions on Image Processing, 29:2766–2779, 2019a.

Ruilin Li, Xiaojing Ye, Haomin Zhou, and Hongyuan Zha. Learning to match via inverse optimal transport. Journal of machine learning research, 20(80):1–37, 2019b.

Wan Li and Chenyang Chang. Semi-supervised image-dehazing network based on a trusted library. Electronics, 14(15):2956, 2025.

Jianxin Lin, Yingce Xia, Tao Qin, Zhibo Chen, and Tie-Yan Liu. Conditional image-to-image translation. In Computer Vision and Pattern Recognition (CVPR), 2018.

Jianlei Liu, Qianwen Hou, Shilong Wang, and Xueqing Zhang. Semi-supervised single image dehazing based on dual-teacher-student network with knowledge transfer. Signal, Image and Video Processing, 18(6):5073–5087, 2024.

Shaojun Ma, Haodong Sun, Xiaojing Ye, Hongyuan Zha, and Haomin Zhou. Learning cost functions for optimal transport. arXiv preprint arXiv:2002.09650, 2020.

Andrey Malinin, Neil Band, German Chesnokov, Yarin Gal, Mark JF Gales, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, et al. Shifts: A dataset of real distributional shift across multiple large-scale tasks. arXiv preprint arXiv:2107.07455, 2021.

Simone Di Marino and Augusto Gerolin. An optimal transport approach for the schrödinger bridge problem and convergence of sinkhorn algorithm. Journal of Scientific Computing, 85(2):27, 2020.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. CoRR, abs/1411.1784, 2014. URL http://arxiv.org/abs/1411.1784.

Tiande Mo, Siqian Zheng, Wai-Yat Chan, and Renhua Yang. Review of ai image enhancement techniques for in-vehicle vision systems under adverse weather conditions. World Electric Vehicle Journal, 16(2):72, 2025.

Petr Mokrov, Alexander Korotin, Alexander Kolesov, Nikita Gushchin, and Evgeny Burnaev. Energy-guided entropic neural optimal transport. In The Twelfth International Conference on Learning Representations, 2024.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. Mem. Math. Phys. Acad. Royale Sci., pp. 666–704, 1781.

Nikita Moriakov, Jonas Adler, and Jonas Teuwen. Kernel of cyclegan as a principal homogeneous space. In International Conference on Learning Representations, 2020.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. Domain adaptation of machine translation with crowdworkers. In Conference on Empirical Methods in Natural Language Processing (EMNLP), 2022.

Kevin P Murphy. Machine learning: a probabilistic perspective. MIT press, 2012.

Aamir Mustafa and Rafał K Mantiuk. Transformation consistency regularization–a semi-supervised paradigm for image-to-image translation. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16, pp. 599–615. Springer, 2020.

Ryosuke Nagumo and Hironori Fujisawa. Density ratio estimation with doubly strong robustness. In Forty-first International Conference on Machine Learning, 2024.

Manan Oza, Himanshu Vaghela, and Sudhir Bagul. Semi-supervised image-to-image translation. In 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), pp. 16–20. IEEE, 2019.

Pauliina Paavilainen, Saad Ullah Akram, and Juho Kannala. Bridging the gap between paired and unpaired medical image translation. In MICCAI Workshop on Deep Generative Models, pp. 35–44. Springer, 2021.

Nishant Panda, Natalie Klein, Dominic Yang, Patrick Gasda, and Diane Oyen. Semi-supervised learning of pushforwards for domain translation & adaptation. arXiv preprint arXiv:2304.08673, 2023.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. Journal of Machine Learning Research, 22(57):1–64, 2021.

Duo Peng, Ping Hu, Qiuhong Ke, and Jun Liu. Diffusion-based image translation with label guidance for domain adaptive semantic segmentation. In IEEE International Conference on Computer Vision (ICCV), 2023.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. Foundations and Trends® in Machine Learning, 11(5-6):355–607, 2019.

Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 14104–14113, 2020.

Allan Pinkus. Approximation theory of the mlp model in neural networks. Acta numerica, 8:143–195, 1999.

Yuxi Ren, Jie Wu, Peng Zhang, Manlin Zhang, Xuefeng Xiao, Qian He, Rui Wang, Min Zheng, and Xin Pan. Ugc: Unified gan compression for efficient image-to-image translation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 17281–17291, 2023.

Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. Bernoulli, pp. 341–363, 1996.

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

Ivan Rubachev, Nikolay Kartashev, Yury Gorishniy, and Artem Babenko. Tabred: A benchmark of tabular machine learning in-the-wild. arXiv preprint arXiv:2406.19380, 2024.

Filippo Santambrogio. Optimal transport for applied mathematicians. Birkäuser, NY, 55(58-63):94, 2015.

Igal Sason. On reverse pinsker inequalities. arXiv preprint arXiv:1503.07118, 2015.

Christopher Scarvelis and Justin Solomon. Riemannian metric learning via optimal transport. In International Conference on Learning Representations. OpenReview, 2023.

Tobias Schröder, Zijing Ou, Jen Lim, Yingzhen Li, Sebastian Vollmer, and Andrew Duncan. Energy discrepancies: a score-independent loss for energy-based models. Advances in Neural Information Processing Systems, 36:45300–45338, 2023.

Liangliang Shi, Gu Zhang, Haoyu Zhen, Jintao Fan, and Junchi Yan. Understanding and generalizing contrastive learning from the inverse optimal transport perspective. In International conference on machine learning, pp. 31408–31421. PMLR, 2023.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. Advances in Neural Information Processing Systems, 36, 2024.

Henrik Skibbe, Akiya Watakabe, F Rachmadi, Carlos Enrique Gutierrez, Ken Nakae, and T Yamamori. Semi-supervised image-to-image translation for robust image registration. Medical Imaging with Deep Learning (MIDL), 2021.

Yang Song and Diederik P Kingma. How to train your energy-based models. arXiv preprint arXiv:2101.03288, 2021.

Andrew M Stuart and Marie-Therese Wolfram. Inverse optimal transport. SIAM Journal on Applied Mathematics, 80(1):599–619, 2020.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. Density ratio estimation in machine learning. Cambridge University Press, 2012.

Xiaole Tang, Xin Hu, Xiang Gu, and Jian Sun. Residual-conditioned optimal transport: Towards structure-preserving unpaired and paired image restoration. In Forty-first International Conference on Machine Learning, 2024. URL https://openreview.net/forum?id=irBHPlknxP.

Panagiotis Theodoropoulos, Nikolaos Komianos, Vincent Pacelli, Guan-Horng Liu, and Evangelos A Theodorou. Feedback schrödinger bridge matching. arXiv preprint arXiv:2410.14055, 2024.

Soumya Tripathy, Juho Kannala, and Esa Rahtu. Learning image-to-image translation using paired and unpaired training samples. In Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part II 14, pp. 51–66. Springer, 2019.

Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving schrödinger bridges via maximum likelihood. Entropy, 23(9):1134, 2021.

Florin-Alexandru Vasluianu, Andrés Romero, Luc Van Gool, and Radu Timofte. Shadow removal with paired and unpaired learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 826–835, 2021.

Cédric Villani et al. Optimal transport: old and new, volume 338. Springer, 2009.

Ke Wang, Michaël Gharbi, He Zhang, Zhihao Xia, and Eli Shechtman. Semi-supervised parametric real-world image harmonization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5927–5936, 2023.

Meilin Wang, Wei Huang, Mingming Gong, and Zheng Zhang. Projection pursuit density ratio estimation. In Forty-second International Conference on Machine Learning, 2025. URL https://openreview.net/forum?id=MgNeJO0PcF.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4):600–612, 2004.

Christina Winkler, Daniel E. Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. CoRR, abs/1912.00042, 2019. URL http://arxiv.org/abs/1912.00042.

Zaifeng Yang and Zhenghua Chen. Learning from paired and unpaired data: Alternately trained cyclegan for near infrared image colorization. In 2020 IEEE International Conference on Visual Communications and Image Processing (VCIP), pp. 467–470. IEEE, 2020.

Peiyu Yu, Yaxuan Zhu, Sirui Xie, Xiaojian Shawn Ma, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Learning energy-based prior model with diffusion-amortized mcmc. Advances in Neural Information Processing Systems, 36:42717–42747, 2023.

Yuan Yuan, Siyuan Liu, Jiawei Zhang, Yongbing Zhang, Chao Dong, and Liang Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 701–710, 2018.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 586–595, 2018.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, pp. 2223–2232, 2017.

Yaxuan Zhu, Jianwen Xie, Ying Nian Wu, and Ruiqi Gao. Learning energy-based models by cooperative diffusion recovery likelihood. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=AyzkDpuqcl.

# CONTENTS



# A NEURAL PARAMETERIZATION

Throughout the main text, we parameterized the cost $c^\theta$ and potential $f^\theta$ using log-sum-exp functions and Gaussian mixtures (see §3.3). At this point, a reader may naturally wonder whether more general parameterizations for $c^\theta$ and $f^\theta$ can be used in our method, such as directly parameterizing both with neural networks. In this section, we affirmatively address this question by providing a procedure to optimize our main objective $\mathcal{L}(\theta)$ in (13) with general parameterizations for $c^\theta$ and $f^\theta$.

### A.1 Algorithm Derivation

We note that a key advantage of our chosen parameterization (see §3.3) is that the normalizing constant $Z_\theta$ appearing in $\mathcal{L}(\theta)$ is available in the closed form. Unfortunately, this is not the case with general parameterizations of $c^\theta$ and $f^\theta$, necessitating the use of more advanced optimization techniques. While the objective $\mathcal{L}(\theta)$ itself may be intractable, we can derive its gradient, which is essential for optimization. The following proposition is derived in a manner similar to (Mokrov et al., 2024), who proposed methods for solving forward entropic OT problems with neural nets.

**Proposition A.1** (Gradient of our main loss (13))**.** *It holds that*

$$\frac{\partial}{\partial\theta}\mathcal{L}(\theta) = \varepsilon^{-1}\Bigg\{\mathbb{E}_{x,y\sim\pi^*}\left[\frac{\partial}{\partial\theta}c^\theta(x,y)\right] - \mathbb{E}_{y\sim\pi_y^*}\left[\frac{\partial}{\partial\theta}f^\theta(y)\right] \\ + \mathbb{E}_{x\sim\pi_x^*}\mathbb{E}_{y\sim\pi^\theta(y|x)}\left[\frac{\partial}{\partial\theta}\big(f^\theta(y) - c^\theta(x,y)\big)\right]\Bigg\}. \tag{19}$$

The gradient formula eliminates the need for the intractable normalizing constant $Z_\theta$, but computing it still requires sampling from the current model $y \sim \pi^\theta(\cdot|x)$. Unlike the Gaussian mixture case in §3.3, we now only access the unnormalized density defined by $c^\theta$ and $f^\theta$, which is not necessarily a Gaussian mixture. To address this, we rely on standard methods for sampling from unnormalized densities, such as Markov Chain Monte Carlo (MCMC) (Andrieu et al., 2003). This enables practical gradient estimation and motivates the training procedure in Algorithm 1, where the conditional distribution is modeled as $\pi^\theta(y|x) \propto \exp\left(\frac{f^\theta(y)-c^\theta(x,y)}{\varepsilon}\right)$, with energy $\varepsilon^{-1}(c^\theta(x,y) - f^\theta(y))$.

**Algorithm 1:** Semi-supervised Learning via Energy-Based Modeling

**Input :** Paired samples $XY_{\text{paired}} \sim \pi^*$; unpaired samples $X_{\text{unpaired}} \sim \pi_x^*$, $Y_{\text{unpaired}} \sim \pi_y^*$;
potential network $f^\theta : \mathbb{R}^{D_y} \to \mathbb{R}$, cost network $c^\theta(x,y) : \mathbb{R}^{D_x} \times \mathbb{R}^{D_y} \to \mathbb{R}$;
number of Langevin steps $K > 0$, Langevin discretization step size $\eta > 0$;
basic noise std $\sigma_0 > 0$; batch sizes $\hat{P}, \hat{Q}, \hat{R} > 0$.

**Output:** trained potential network $f^{\theta^*}$ and cost network $c^{\theta^*}$ recovering $\pi^{\theta^*}(y|x)$ from (10).

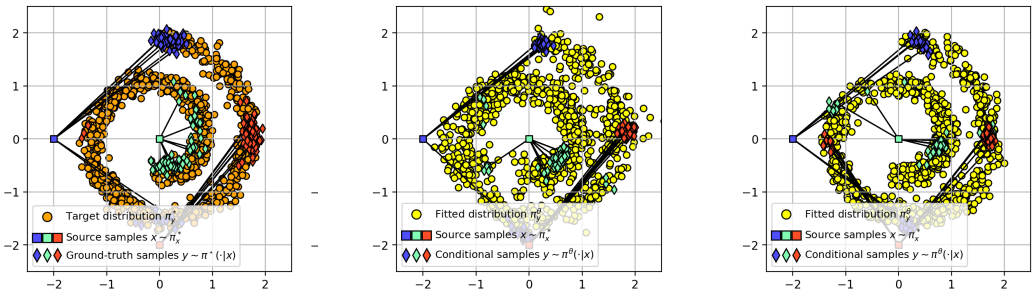
**for** $i = 1, 2, \dots$ **do**

- Derive batches $\{\hat{x}_p, \hat{y}_p\}_{p=1}^{\hat{P}} = XY \sim \pi^*$, $\{x_n\}_{q=1}^{\hat{Q}} = X \sim \pi_x^*$, $\{y_r\}_{r=1}^{\hat{R}} = Y \sim \pi_y^*$;
- Sample basic noise $Y^{(0)} \sim \mathcal{N}(0, \sigma_0)$ of size $\hat{Q}$;
- **for** $k = 1, 2, \dots, K$ **do**
  - Sample $Z^{(k)} = \{z_q^{(k)}\}_{q=1}^{\hat{Q}}$, where $z_q^{(k)} \sim \mathcal{N}(0,1)$;
  - Obtain $Y^{(k)} = \{y_q^{(k)}\}_{q=1}^{\hat{Q}}$ with Langevin step:
  - $y_q^{(k)} \leftarrow y_q^{(k-1)} + \frac{\eta}{2\varepsilon} \cdot \texttt{stop\_grad}\Big(\frac{\partial}{\partial y}\left[f^\theta(y) - c^\theta(x_q, y)\right]\big|_{y=y_q^{(k-1)}}\Big) + \sqrt{\eta} z_q^{(k)}$
- $\widehat{\mathcal{L}} \leftarrow \frac{1}{\hat{P}}\left[\sum_{x_p,y_p\in XY} c^\theta(x_p, y_p)\right] + \frac{1}{\hat{Q}}\left[\sum_{y_q^{(K)}\in Y^{(K)}} f^\theta\left(y_q^{(K)}\right)\right] - \frac{1}{\hat{R}}\left[\sum_{y_r\in Y} f_\theta(y_r)\right]$;
- Perform a gradient step over $\theta$ by using $\frac{\partial\widehat{\mathcal{L}}}{\partial\theta}$;

In Algorithm 1, we use the Unadjusted Langevin Algorithm (ULA) (Roberts & Tweedie, 1996), a standard MCMC method. For an in-depth discussion on EBM training methods, see the recent surveys (Song & Kingma, 2021; Carbone, 2024).

Our proposed *inverse* OT algorithm is closely related to the *forward* OT framework in (Mokrov et al., 2024, Algorithm 1), with key distinctions: **(1)** it learns the cost function $c^\theta$ during training, and **(2)** it leverages both paired and unpaired data.

Below, we demonstrate a proof-of-concept performance of Algorithm 1 on two setups: an illustrative 2D example.




(a) Ground truth. (b) $P = 128$; $Q = R = 1024$ (c) $P = 16K$; $Q = R = 16K$

Figure 5: Performance of our Algorithm 1 in the *Gaussian* $\rightarrow$ *Swiss Roll* mapping task (§5.1). We use MLPs to parametrize both the potential function $f^\theta$ and the cost function $c^\theta$.

### A.2 Illustrative Example

**Setup.** We begin with a 2D example to showcase the capability of Algorithm 1 to learn conditional plans using a fully neural network-based parametrization. Specifically, we conduct experiments on the *Gaussian* $\rightarrow$ *Swiss Roll* mapping problem (see §5.1) using two datasets: one containing 128 paired samples (described in §5.1) and another with 16K paired samples (detailed in Appendix D.3).

**Discussion.** It is worth noting that the model's ability to fit the target distribution is influenced by the amount of labeled data used during training. When working with partially labeled samples (as shown in Figure 5b), the model's fit to the target distribution is less accurate compared to using a larger dataset. However, even with limited labeled data, the model still maintains good accuracy in terms of the paired samples. On the other hand, when provided with fully labeled data (see Figure 5c), the model generates more consistent results and achieves a better approximation of the target distribution. A comparison of the results obtained using Algorithm 1 with neural network parametrization and those achieved using Gaussian parametrization (Figure 2d) reveals that Algorithm 1 exhibits greater instability. This observation aligns with the findings of (Mokrov et al., 2024, Section 2.2), which emphasize the instability and mode collapse issues commonly encountered when working with EBMs.

**Implementation Details.** We employ MLPs with hidden layer configurations of $[128, 128]$ and $[256, 256, 256]$, using $LeakyReLU(0.2)$ for the parametrization of the potential $f^\theta$ and the cost $c^\theta$, respectively. The learning rates are set to $lr_{\text{paired}} = 5 \times 10^{-4}$ and $lr_{\text{unpaired}} = 2 \times 10^{-4}$. The sampling parameters follow those specified in (Mokrov et al., 2024).

### A.3 Details on Colored Images Example

**Implementation Details.** We adopt the same parameters as in (Mokrov et al., 2024), except for the cost function:

$$c^\theta(x, y) = \frac{1}{D_y} \|U^\theta_{\text{net}} - y\|_2^2.$$

Here, the dimensions of source and target spaces are $D_x = D_y = 3 \times 32 \times 32$ and $U^\theta_{\text{net}} : \mathbb{R}^{D_x} \rightarrow \mathbb{R}^{D_y}$ is a neural net function with U-Net architecture (Ronneberger et al., 2015) with 16 layers. The first layer has 64 filters, and the number of filters doubles in each subsequent layer. The experiment was run for 10,000 iterations on a 2080 Ti GPU, completing in approximately 40 minutes.

### A.4 Conclusion

It is important to recognize that the field of Energy-Based Models has undergone significant advancements in recent years, with the development of numerous scalable approaches. For examples of such progress, we refer readers to recent works by (Geng et al., 2024; Carbone et al., 2023; Du et al., 2021; Gao et al., 2021) and other the references therein. Additionally, we recommend the comprehensive tutorial by (Song & Kingma, 2021; Carbone, 2024) for an overview of train-

ing methods for EBMs. Given these advancements, it is reasonable to expect that by incorporating more sophisticated techniques into our basic Algorithm 1, it may be possible to scale the method to handle high-dimensional setups, such as image data. However, exploring these scaling techniques is beyond the scope of the current paper, which primarily focuses on the general methodology for semi-supervised domain translation. The investigation of methods to further scale our approach as a promising future research avenue.

# B Additional Discussions

## B.1 Entropic/Weak/Inverse Optimal Transport

In this section, we explain our motivation for adopting the *entropic* OT formulation rather than the standard OT formulation (OT). Specifically, we focus on the *weak semi-dual* formulation of the entropic OT problem (Mokrov et al., 2024, §3.1), as opposed to its standard *semi-dual* form (Genevay, 2019, §4.3), and highlight its connections to the existing inverse entropic optimal transport frameworks in the literature.

**Classic OT.** Given source and target distributions $\alpha \in \mathcal{P}_{\text{ac}}(\mathcal{X})$ and $\beta \in \mathcal{P}_{\text{ac}}(\mathcal{Y})$, and a cost function $c^* : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the *primal* optimal transport problem (Villani et al., 2009) is defined as:

$$\text{OT}_{c^*}(\alpha, \beta) \stackrel{\text{def}}{=} \min_{\pi \in \Pi(\alpha,\beta)} \mathbb{E}_{x,y\sim\pi}[c^*(x, y)]. \tag{OT}$$

This formulation was originally introduced by Kantorovich (Kantorovich, 1942) as a relaxation of Monge's original problem (Monge, 1781), which is more restrictive because it does not allow mass to be split, resulting in *deterministic* solutions called optimal transport *maps*. However, as is well-known in optimal transport theory (see (Villani et al., 2009, §9)), solutions to problem (OT), called optimal transport *plans*, can still be deterministic. For example, when the cost is quadratic and the measures are absolutely continuous, Brenier's theorem (Remark 2.24 in (Peyré et al., 2019)) guarantees that the optimal transport plan is deterministic. Specifically, each $x$ is mapped deterministically to $y = T^*(x)$ for some optimal map $T^*$, meaning that the conditional distribution $\pi^*(y|x)$ collapses to a single point mass $\delta_{T^*(x)}$.

Such deterministic plans, however, are unsuitable for our semi-supervised domain translation setup, where a multimodal transport behavior of $\pi^*(y|x)$ may be necessary. Our synthetic experiments in §5.1 (Figure 2) illustrate these cases. To enforce mapping uniqueness while allowing *stochastic* (i.e., non-deterministic) mappings, a common approach is to regularize (OT) with an entropy term, which makes the objective strictly convex with respect to $\pi$, as discussed below.

**Entropic OT.** The work of (Cuturi, 2013) proposed regularizing (OT) with an entropy term, known as entropic OT (EOT), to improve computational tractability of OT (Genevay, 2019). Moreover, besides the computational advantages, the EOT problem has a connection to the *Static Schrödinger Bridge* (SB) problem (Léonard, 2014):

$$\pi^* = \arg\min_{\pi \in \Pi(\alpha,\beta)} \text{KL}\left(\pi \| \pi^{\text{ref}}\right), \tag{SB}$$

where the aim of the problem is to find the transport plan $\pi \in \Pi(\alpha, \beta)$ closest to $\pi^{\text{ref}}$ in terms of the Kullback-Leibler (KL) divergence. Observe that EOT and the static SB problem are equivalent:

$$\min_{\pi \in \Pi(\alpha,\beta)} \text{KL}\left(\pi \| \pi^{\text{ref}}\right) = \min_{\pi \in \Pi(\alpha,\beta)} \mathbb{E}_{x,y\sim\pi} \log \frac{\pi(x, y)}{\pi^{\text{ref}}(x, y)} = \tag{20}$$

$$\min_{\pi \in \Pi(\alpha,\beta)} \Big\{ \mathbb{E}_{x,y\sim\pi} \underbrace{\left[-\log \pi^{\text{ref}}(x, y)\right]}_{\stackrel{\text{def}}{=} c^*(x,y)} - \text{H}(\pi) \Big\} = \min_{\pi \in \Pi(\alpha,\beta)} \Big\{ \mathbb{E}_{x,y\sim\pi}[c^*(x, y)] - \text{H}(\pi) \Big\}. \tag{21}$$

Using the equivalence of the following formulations (see (Mokrov et al., 2024, Eq. 2–4) and (Gushchin et al., 2023b, Eq. 3–5) for details):

$$\begin{cases} \text{EOT}^{(1)}_{c^*,\varepsilon}(\alpha, \beta) \\ \text{EOT}^{(2)}_{c^*,\varepsilon}(\alpha, \beta) \\ \text{EOT}_{c^*,\varepsilon}(\alpha, \beta) \end{cases} = \min_{\pi \in \Pi(\alpha,\beta)} \mathbb{E}_{x,y\sim\pi}[c^*(x, y)] + \begin{cases} +\varepsilon \text{KL}\left(\pi \| \alpha \otimes \beta\right), \\ -\varepsilon H(\pi), \\ -\varepsilon \mathbb{E}_{x\sim\alpha} H(\pi(\cdot|x)), \end{cases}$$

we conclude that equation (21) is equivalent to (1) for $\varepsilon = 1$.

From the equations (20)–(21), we see that the cost function $c^*(x, y)$ defines a reference measure that determines the mapping we aim to reconstruct in the forward problem (1). Furthermore, since KL minimization is equivalent to maximum likelihood estimation, EOT is theoretically consistent with standard probabilistic modeling principles.

**Weak OT.** Following (Mokrov et al., 2024), we provide more details regarding *weak* OT. For a more rigorous treatment, see (Gozlan et al., 2017; Backhoff-Veraguas et al., 2019). Given a *weak* transport cost $C^* : \mathcal{X} \times \mathcal{P}(\mathcal{Y}) \to \mathbb{R}$, which penalizes the displacement of a point $x \in \mathcal{X}$ into a distribution $\pi(\cdot|x) \in \mathcal{P}(\mathcal{Y})$, the weak OT problem is defined as:

$$\text{WOT}_{C^*}(\alpha, \beta) \stackrel{\text{def}}{=} \min_{\pi \in \Pi(\alpha,\beta)} \mathbb{E}_{x\sim\alpha} C^*(x, \pi(\cdot|x)). \tag{p-WOT}$$

Just as in the classical OT problem (OT), the weak OT formulation (p-WOT) also enjoys *strong duality* under mild assumptions (see (Gozlan et al., 2017, Theorem 9.5); (Backhoff-Veraguas & Pammer, 2022, Theorem 3.3)). This means that the weak formulation (p-WOT) admits an equivalent *weak semi-dual* representation:

$$\text{WOT}_{C^*}(\alpha, \beta) = \max_{f \in \mathcal{C}(\mathcal{Y})} \left\{ \mathbb{E}_{x\sim\alpha} f^{C^*}(x) + \mathbb{E}_{y\sim\beta} f(y) \right\}, \tag{sd-WOT}$$

where $\mathcal{C}(\mathcal{Y})$ denotes the set of continuous functions over $\mathcal{Y}$ and $f^C$ so-called *weak $C$-transform*:

$$f^C(x) \stackrel{\text{def.}}{=} \min_{\mu \in \mathcal{P}(\mathcal{Y})} \left\{ C(x, \mu) - \mathbb{E}_{y\sim\mu} f(y) \right\}. \tag{22}$$

Futhermore, note that the EOT formulation in (1) can be seen as a special case of the weak OT problem (p-WOT), corresponding to the following weak transport cost $C^*_{\text{EOT}}$:

$$C^*_{\text{EOT}}(x, \pi(\cdot|x)) \stackrel{\text{def}}{=} \mathbb{E}_{y\sim\pi(\cdot|x)}[c^*(x, y)] - \varepsilon \text{H}\left(\pi(\cdot|x)\right). \tag{23}$$

Substituting expression above into (22), we obtain equation (3) for the weak *entropic* $c$-transform:

$$f^{c^*}(x) = \min_{\mu \in \mathcal{P}(\mathcal{Y})} \left\{ \mathbb{E}_{y\sim\mu}[c^*(x, y)] - \varepsilon \text{H}\left(\mu\right) - \mathbb{E}_{y\sim\mu} f(y) \right\},$$

which admits a closed-form expression given in (Mokrov et al., 2024, Eq. 14), and which we use in our work (4):

$$f^c(x) = -\varepsilon \log \int_{\mathcal{Y}} \exp\left( \frac{f(y) - c(x, y)}{\varepsilon} \right) \mathrm{d}y.$$

Furthermore, Appendix A.1 of (Mokrov et al., 2024) provides a detailed discussion of the relationship between the weak entropic $c$-transform and the so-called $(c, \varepsilon)$-*transform* (Genevay et al., 2019, 4.15), (Marino & Gerolin, 2020, Theorem 1.2):

$$v^{c,\varepsilon}(x) = -\varepsilon \log \mathbb{E}_{y\sim\beta} \left[ \exp\left( \frac{v(y) - c(x, y)}{\varepsilon} \right) \right], \tag{24}$$

which is used in the *semi-dual* formulation of the EOT problem (Genevay, 2019, §4.3):

$$\text{OT}^{\text{semi-dual}}_{c^*,\varepsilon}(\alpha, \beta) = \max_{v \in \mathcal{C}(\mathcal{Y})} \left\{ \mathbb{E}_{x\sim\alpha} v^{c^*,\varepsilon}(x) + \mathbb{E}_{y\sim\beta} v(y) \right\}. \tag{sd-EOT}$$

As noted in (Mokrov et al., 2024), the main difference between (4) and (24) lies in the integration measure: (24) integrates with respect to $\beta$, while (4) uses the standard Lebesgue measure.

For completeness, we present below the *dual* formulation of EOT with a slightly different regularization term, $+\varepsilon \text{KL}\left(\pi \| \alpha \otimes \beta\right)$. As noted above, this is equivalent to our choice of regularization, but it is the version commonly used in inverse problems and will be discussed later:

$$\text{OT}^{\text{dual}}_{c^*,\varepsilon}(\alpha, \beta) = \max_{\substack{u \in \mathcal{C}(\mathcal{X}) \\ v \in \mathcal{C}(\mathcal{Y})}} \left\{ \mathbb{E}_{x\sim\alpha} u(x) + \mathbb{E}_{y\sim\beta} v(y) - \mathbb{E}_{x,y\sim\alpha\otimes\beta} \left[ \exp\left( \frac{u(x) + v(y) - c(x, y)}{\varepsilon} \right) \right] \right\}, \tag{d-EOT}$$

where the optimization is performed over two Kantorovich potentials $u$ and $v$, in contrast to the single potential used in our formulation (13). With that said, we are ready to discuss the existing formulations of inverse entropic optimal transport.

**Inverse OT.** The use of the entropic formulation for inverse optimal transport was first proposed in (Du & Mordatch, 2019, Eq. 8). Their setup, identical to our formulation (5), restricted attention to bilinear cost functions of the form $c_A(x, y) = x^\top Ay$, (Eq. 5), with the goal of recovering the matrix $A$ in a discrete setting. A subsequent work (Ma et al., 2020, Eq. 21) extended this idea to the continuous setting by introducing a loss function for learning cost functions, based on the dual formulation (d-EOT) of the EOT problem. As shown in (Andrade et al., 2023, Appendix A.1), their formulation and ours are equivalent, and both admit a maximum likelihood interpretation, consistent with our derivation in §3.1.

The most directly related approach is that of (Andrade et al., 2025, Lemma 1), which addresses the unbalanced OT framework (Chizat et al., 2018) while still relying on the dual formulation (d-EOT). They employ a linearly parameterized cost function (Eq. 4), but their focus is on establishing bounds for cost recovery, in contrast to our emphasis on semi-supervised domain translation.

For further formulations of inverse OT, we refer readers to the works cited in the introduction of (Andrade et al., 2023).

### B.2 Discrete Spaces Extension

Our theoretical framework is not limited to continuous spaces $\mathcal{X}, \mathcal{Y}$. For instance, if the target space $\mathcal{Y}$ is discrete and takes values in a finite set $\mathbb{K} = \{1, 2, \dots, K\}$, such as a set of categories, our method remains directly applicable. In this case, the dual potential $f^\theta$ (16) can be represented as a vector of length $K$, and the cost function $c^\theta(x, y)$ (15) can be implemented with a standard neural network. The partition function $Z^\theta(x)$ can then be computed as a finite sum over the $K$ terms, making the implementation straightforward. Note that the input $x$ can be either continuous or discrete - it does not affect the formulation.

Challenges arise when $y$ is a more complex discrete object, such as a structured output like a sequence of $T$ tokens drawn from a dictionary of size $K$, i.e., $\mathbb{K}^T$. In such cases, parameterizing $f_\theta$, computing $Z_\theta$, and sampling from the associated energy-based model become significantly more difficult, requiring advanced inference and training techniques, see (Holderrieth et al., 2025) for details.

Discrete domains (Austin et al., 2021; Campbell et al., 2022; Gat et al., 2024; Ksenofontov & Korotin, 2025) have received considerable attention recently, and extending our methodology to such spaces represents a promising direction for future research.

### B.3 Partially Paired Data

A potential limitation of the formulation in equation (13) is that it implicitly relies on the paired data having marginals matching the true distributions $\pi^*_x$ and $\pi^*_y$. If the paired samples are artificially selected, so that their empirical $x$- and $y$-marginals deviate from $\pi^*_x$ and $\pi^*_y$—one might suspect that the objective no longer corresponds to the KL functional in §3.1. In practice, however, this is not a fundamental issue: the theoretical formulation remains valid, which we discuss below.

Assume that the observed pairs $(x, y)$ come from a joint distribution $\pi^*_{\text{subset}}$ supported on a limited subset of the support of $\pi^*$, with $x$-marginal $\mu_x$ and conditional density $\pi^*(y \mid x)$. In this setting, the induced $y$-marginal is $\nu_y(y) = \mathbb{E}_{x\sim\mu_x}\pi^*(y|x)$ and the *ground-truth joint density* becomes

$$\pi^*_{\text{subset}}(x, y) = \mu_x(x)\pi^*(y|x).$$

Applying the same derivation as in §3.1, we obtain:

$$\mathrm{KL}\left(\pi^*_{\text{subset}} \| \pi^\theta\right) = \underbrace{\mathrm{KL}\left(\mu_x \| \pi^\theta_x\right)}_{\text{Marginal}} + \underbrace{\mathbb{E}_{x\sim\mu_x}\mathrm{KL}\left(\pi^*(\cdot|x) \| \pi^\theta(\cdot|x)\right)}_{\text{Conditional}}. \tag{25}$$

Focusing on the conditional term:

$$\mathbb{E}_{x\sim\mu_x}\mathbb{E}_{y\sim\pi^*(\cdot|x)}\left[\log \pi^*(y|x) - \log \pi^\theta(y|x)\right] = -\mathbb{E}_{x\sim\mu_x}\mathrm{H}\left(\pi^*(\cdot|x)\right) - \mathbb{E}_{x,y\sim\pi^*_{\text{subset}}}\log \pi^\theta(y|x). \tag{26}$$

Thus, we recover the same conditional log-likelihood structure as in (9). Substituting the EBM parametrizations (10) and (12), we obtain

$$\mathbb{E}_{x,y\sim\pi^*_{\text{subset}}}[c(x,y)] - \mathbb{E}_{y\sim\nu_y} f(y) + \mathbb{E}_{x\sim\mu_x} \log Z^\theta(x) = \tag{27}$$

$$\mathbb{E}_{x,y\sim\pi^*_{\text{subset}}}[c(x,y)] - \mathbb{E}_{y\sim\pi^*_y}\left[\frac{\nu_y(y)}{\pi^*_y(y)} f(y)\right] + \mathbb{E}_{x\sim\pi^*_x}\left[\frac{\mu_x(x)}{\pi^*_x(x)} \log Z^\theta(x)\right]. \tag{28}$$

Introducing the *weights*:

$$w_x(x) = \frac{\mu_x(x)}{\pi^*_x(x)}, \qquad w_y(y) = \frac{\nu_y(y)}{\pi^*_y(y)}, \tag{29}$$

we obtain the corrected objective:

$$\mathcal{L}_q(\theta) = \underbrace{\varepsilon^{-1}\mathbb{E}_{x,y\sim\pi^*_{\text{subset}}}[c^\theta(x,y)]}_{\text{Joint, requires pairs } (x,y)\sim\pi^*_{\text{subset}}} \underbrace{-\varepsilon^{-1}\mathbb{E}_{y\sim\pi^*_y}[w_y(y) f^\theta(y)]}_{\text{Marginal, requires } y\sim\pi^*_y} + \underbrace{\mathbb{E}_{x\sim\pi^*_x}[w_x(x)\log Z^\theta(x)]}_{\text{Marginal, requires } x\sim\pi^*_x} \to \min_\theta. \tag{30}$$

A practical way to estimate the required ratios is classifier-based density ratio estimation, widely used in covariate-shift adaptation (Gretton et al., 2009; Sugiyama et al., 2012). To estimate a marginal ratio such as $w_x(x) = \mu_x(x)/\pi^*_x(x)$, we draw samples from the true marginal $\pi^*_x$ and the biased marginal $\mu_x$, label them as target (1) and observed (0), and train a probabilistic classifier $s_\varphi(x) = \text{Prob}(\text{target} \mid x)$. With balanced class priors, $\widehat{w}_x(x) = \frac{s_\varphi(x)}{1-s_\varphi(x)}$. The same holds for $w_y(y)$. This method is requires no density estimation. For recent advancement in density ratio estimation, please refer to (Nagumo & Fujisawa, 2024; Wang et al., 2025). Thus, even if the paired data are *artificially biased*, the loss remains correct as long as the true marginals are known and appropriate weights are applied.

### B.4 Examples of Semi-supervised Domain Translation Setups

In this section we outline some real-world scenarios, where *semi-supervised* setup are very natural.

- **Image Harmonization in Photo Editing** (Wang et al., 2023). Photo compositing often involves placing a foreground object into a new background, but realistic blending (e.g., matching lighting and color tone) is challenging. While only a small set of artist-labeled (paired) composites may be available, large collections of unlabeled (unpaired) composites can be gathered from the web.
- **Scene Stylization (e.g., Anime Rendering)** (Jiang et al., 2023b). Transforming real-world photos into anime-style renderings is popular in gaming and animation but is limited by the scarcity of labeled real–anime image pairs.
- **Image Enhancement for Outdoor Vision** (Li et al., 2019a; Liu et al., 2024; Cui et al., 2024; Li & Chang, 2025; Hou et al., 2025). Adverse weather and low-light conditions can compromise the visual systems of autonomous vehicles, such as self-driving cars and UAVs, leading to challenges in both decision-making and navigation. For a comprehensive overview of these scenarios and existing semi-supervised approaches, see (Mo et al., 2025).
- **Biomedical Image Registration (Microscopy)** (Skibbe et al., 2021). In neuroscience research, aligning images from different modalities (e.g., tracer vs. Nissl stain) is crucial but difficult due to modality shifts. Only a limited number of images can be manually registered (paired data), while many are unregistered (unpaired).

The examples above underscore the importance of developing methods for semi-supervised domain translation, which have applications in rendering, image editing, design, computer graphics, and autonomous driving, while also streamlining existing digital content creation pipelines. At the same time, it is important to recognize that the rapid advancement of generative models may have unintended consequences, potentially impacting certain jobs within these industries.

### B.5 Related Works

This section provides the detailed discussion of related work and methods that were only briefly summarized in §4, and includes additional coverage of metric learning.

**Semi-supervised models.** As discussed in §1, many existing semi-supervised domain translation methods combine paired and unpaired data by introducing multiple loss terms into *ad hoc optimization objectives*. Several works, such as (Jin et al., 2019, §3.3), (Tripathy et al., 2019, §3.5), (Oza et al., 2019, §C), (Paavilainen et al., 2021, §2), (Chen et al., 2023, §3.3), (Ren et al., 2023, §3) and (Panda et al., 2023, Eq. 8), employ GAN-base objectives, which incorporate the GAN losses (Goodfellow et al., 2014) augmented with specific regularization terms to utilize paired data. Although most of these methods were initially designed for the image-to-image translation, their dependence on GAN objectives enables their application to broader domain translation tasks. In contrast, the approaches introduced by (Mustafa & Mantiuk, 2020, §3.2) and (Tang et al., 2024, Eq. 8) employ loss functions specifically tailored for the image-to-image translation, making them unsuitable for the general domain translation problem described in §2.1.

Another line of research explores methods based on *key-point guided OT* (Gu et al., 2022), which integrates paired data information into the discrete transport plan. Building on this concept, (Gu et al., 2023) uses such transport plans as heuristics to train a conditional score-based model on unpaired or semi-paired data. Furthermore, recent work (Theodoropoulos et al., 2024) heuristically incorporates paired data into the cost function $c(x, y)$ in (1) with corresponding dynamical formulation.

Importantly, the paradigms outlined above do not offer any theoretical guarantees for reconstructing the conditional distribution $\pi^*(y|x)$, as they depend on heuristic loss constructions. We show that such approaches actually fail to recover the true plan even in toy 2-dimensional cases, refer to experiments in §5 for an illustrative example. We also note that there exist works addressing the question of incorporating unpaired data to the log-likelihood training (9) by adding an extra likelihood terms, see (Atanov et al., 2019; Izmailov et al., 2020). However, they rely on $x$ being a discrete object (e.g., a class label) and does not easily generalize to the continuous case, see Appendix D.2 for details.

**Inverse OT solvers**. As highlighted in §2.2, the task of inverse optimal transport (IOT) implies learning the cost function from samples drawn from an optimal coupling $\pi^*$. Existing IOT solvers (Dupuy et al., 2019; Li et al., 2019b; Stuart & Wolfram, 2020; Galichon & Salanié, 2022; Andrade et al., 2025) focus on reconstructing cost functions primarily from discrete marginal distributions, in particular, using the log-likelihood maximization techniques (Dupuy et al., 2019), see the introduction of (Andrade et al., 2023) for a review. Additionally, the recent work by (Shi et al., 2023) explores the IOT framework in the context of contrastive learning. In contrast, we develop a log-likelihood based approach aimed at learning conditional distributions $\pi^\theta(\cdot|x) \approx \pi^*(\cdot|x)$ using both paired and unpaired data but not the cost function itself.

**Forward OT solvers.** Our solver is based on the framework of (Mokrov et al., 2024), which proposed a *forward* solver for *unsupervised* domain translation. In contrast, our approach integrates the optimization of the cost function directly into the objective (equation (18)), allowing for effective utilization of paired data. Additionally, we extend the Gaussian Mixture parameterization proposed by (Korotin et al., 2024; Gushchin et al., 2024a), which was originally developed as a forward solver for entropic OT with a quadratic cost function $c^*(x, y) = \frac{1}{2}\|x - y\|_2^2$. Our work generalizes this solver to accommodate a wider variety of cost functions, as specified in equation (15). As a result, our approach also functions as a novel forward solver for these generalized cost functions.

Recent work by (Howard et al., 2024) proposes a framework for learning cost functions to improve the mapping between the domains. However, it is limited by the use of deterministic mappings, i.e., does not have the ability to model non-degenerate conditional distributions.

Another work by (Asadulaev et al., 2024) introduces a neural network-based OT framework for semi-supervised scenarios, utilizing general cost functionals for OT. However, their method requires *manually* constructing cost functions which can incorporate class labels or predefined pairs. In contrast, our method dynamically adjusts the cost function during training, offering a more flexibility.

**Metric-learning and OT.** In addition to purely inverse OT approaches, there is a line of work that aims to learn the ground metric used by optimal transport. A seminal work (Cuturi & Avis, 2014) introduced *ground metric learning* in a supervised setting, where they optimize over metric matrices so that OT distances between labeled histograms better reflect the class structure. Building on this, (Huizing et al., 2022) propose *unsupervised ground metric learning* via what they call Wasserstein singular vectors. They jointly learn a ground metric on features and a distance between samples by finding positive singular vectors of the mapping from metric matrices to OT distance matrices. Their method uses stochastic approximation with entropic regularization and is scalable to high-

dimensional data. More recently, the work (Auffenberg et al., 2025) analyze this fixed-point problem more deeply: they prove convergence for a stochastic fixed-point iteration (even in scenarios where classical contraction assumptions do not hold) and show that their framework naturally recovers Mahalanobis-type metrics and graph-Laplacian parameterizations as special cases.

In another direction, (Scarvelis & Solomon, 2023) introduce a *Riemannian metric-learning* framework: they parametrize a spatially-varying metric tensor as a neural network over a manifold, and optimize it so that OT distances under this learned geometry better explain meaningful interpolations, such as trajectories in scRNA data or bird migration. In graph-structured domains, (Heitz et al., 2021) learn ground metrics constrained to be geodesic distances on a graph, allowing a structured and efficient metric learning aligned with the graph topology.

Moreover, (Jawanpuria et al., 2025) propose to learn a symmetric positive definite (SPD) ground metric matrix by optimizing over the Riemannian manifold of SPD matrices, enabling the cost metric to adapt flexibly to data while jointly optimizing the OT distance. Finally, in the context of *domain adaptation*, (Kerdoncuff et al., 2020) present `MLOT`, which learns a global Mahalanobis metric that improves the alignment of source and target distributions under OT.

While these metric-learning works learn a distance function (or cost metric) via OT, they typically assume particular parametric forms (Mahalanobis, SPD matrices, or constructed on manifolds) and focus on matching distributions or aligning domains. In contrast, our approach learns conditional couplings $\pi^\theta(\cdot|x)$ (not just a ground cost), and integrates cost learning dynamically into a likelihood-based solver over paired and unpaired data. Moreover, our cost parameterization extends beyond classical metric forms, enabling more flexible and expressive cost functions (see Eq. (15)).

# C General Details of Experiments

## C.1 General Implementation Details

**Parametrization.** The depth and number of hidden layers vary depending on the experiment.

For $f^\theta$ (16) we represent:

- $w_n$ as $\log w_n$,
- $b_n$ directly as a vector,
- the matrix $B_n$ in diagonal form, with $\log(B_n)_{i,i}$ on its diagonal. This choice not only reduces the number of learnable parameters in $\theta_f$ but also enables efficient computation of $B_n^{-1}$ with a time complexity of $\mathcal{O}(D_y)$.

For $c^\theta$ (15), we represent:

- $v_m(x)$ as a multilayer perceptron (MLP) with ReLU activations (Agarap, 2018) and a LogSoftMax output layer,
- $a_m(x)$ as an MLP with ReLU activations.

**Optimizers.** We employ two separate Adam optimizers (Kingma, 2014) with different step sizes for paired and unpaired data to enhance convergence.

**Initialization.**

- $\log w_n$ as $\log \frac{1}{n}$,
- $b_n$ using random samples from $\pi_y^*$,
- $\log(B_n)_{j,j}$ with $\log(0.1)$,
- for the neural networks, we use the default PyTorch initialization (Ansel et al., 2024),
- $\varepsilon = 1$ for all experiments, since the solver is independent of $\varepsilon$, as discussed in §2.2.

## C.2 Gaussian To Swiss Roll Mapping

**Implementation Details.** We choose the parameters as follows: $N = 50$, $M = 25$, with learning rates $lr_{\text{paired}} = 3 \times 10^{-4}$ and $lr_{\text{unpaired}} = 0.001$. We utilize a two-layer MLP network for the function

$a_m(x)$ and a single-layer MLP for $v_m(x)$. The experiments are executed in parallel on a 2080 Ti GPU for a total of 25,000 iterations, taking approximately 20 minutes to complete.

### C.3 Weather Prediction

We select two distinct months from the dataset (Malinin et al., 2021; Rubachev et al., 2024) and translate the meteorological features from the source month (January) to the target month (June). To operate at the monthly scale, we represent a source data point $x \in \mathbb{R}^{188}$ as the mean and standard deviation of the features collected at a specific location over the source month. The targets $y \in \mathbb{R}^{94}$ correspond to individual measurements in the target month.

Pairs are constructed by aligning a source data point with the target measurements at the same location. Consequently, multiple target data points $y$ may correspond to a single source point $x$ and represent samples from conditional distributions $\pi^*(y|x)$. The measurements from non-aligned locations are treated as unpaired. Such unpaired data naturally arise because stations may not provide reliable measurements in both months, for example, due to maintenance, sensor failures, extreme weather, or connectivity issues.

We obtain 500 unpaired and 192 paired data samples. For testing, 100 pairs are randomly selected.

**Implementation Details.** In general, we consider the same setting as in C.2. Specifically, we set $N = 10, M = 1$ and the number of optimization steps to $30,000$. The baseline uses an MLP network with the same number of parameters, predicting the parameters of a mixture of 10 Gaussians.

**Extremely Low-Data Regimes Discussion.** As it clear from Table 1, our method diverges when trained on very few samples (e.g., 5 paired and no unpaired). This is not surprising given the high dimensionality of the data ($D = 94$) and the number of learnable parameters ($|\theta| = 2668$). In such low-data regimes, the model likely overfits the cost function $c^\theta$ to the small paired dataset, which can cause instability. This issue could potentially be alleviated by simplifying the model, for instance by using a shallow or even linear parameterization of $c^\theta$ (Andrade et al., 2025). However, for consistency and fairness, we kept the architecture fixed across all experiments in the table.

### C.4 Image Translation via ALAE

Finally, we review experiments on two types of image translation tasks: **(i)** Gender translation and **(ii)** Age translation. Extended results for the Woman-to-Man task are shown in Figure 8, and for Old-to-Young in Figure 9 and Table 4.

**Setup.** We follow the experimental setup of (Theodoropoulos et al., 2024), using the pre-trained ALAE autoencoder (Pidhorskyi et al., 2020) on the $1024 \times 1024$ FFHQ dataset (Karras et al., 2019). Translation is performed in the 512-dimensional latent space.

**Baseline method.** We used the publicly available FSBM (Theodoropoulos et al., 2024) implementation from GitHub[1]. However, due to reproducibility issues in the repository, we generated 2K paired samples ourselves via the procedure described in Appendix C.3 of the original paper.

**Metric computation.** Metrics were computed using `TorchMetrics` (Falcon et al., 2020) with a batch size of 128. All metrics measure similarity between the generated and target distributions and are averaged across three independent runs with different seeds. Results are reported rounded to the first significant digit.

**Implementation Details.** We largely follow the setup in Appendix C.2, setting $N = 10$, $M = 1$, and using 10K optimization steps. Our method employs a single-layer MLP to predict the parameters of a mixture of 10 Gaussians.

| Method | FID ↓ | SSIM ↑ | LPIPS ↓ |
|---|---|---|---|
| FSBM | $11.5 \pm 0.6$ | $0.5285 \pm 0.0008$ | $0.5628 \pm 0.0004$ |
| Ours | $\mathbf{9.4 \pm 0.2}$ | $\mathbf{0.5361 \pm 0.0004}$ | $\mathbf{0.5560 \pm 0.0005}$ |

Table 4: Metrics for Old-to-Young translation .

[1] https://github.com/panostheo98/FSBM

# D GAUSSIAN TO SWISS ROLL MAPPING

## D.1 PAIRED DATA GENERATION

**Generation process.** To create the ground truth plan $\pi^*$, we utilize the following procedure: sample a mini-batch of size 64 and then determine the optimal mapping using the entropic Sinkhorn algorithm, as outlined in (Cuturi, 2013) and implemented in (Flamary et al., 2021). This process is repeated $P$ times to generate the required number of pairs.

| Method | Paired $(x, y) \sim \pi^*$ | Unpaired $x \sim \pi_x^*$ | Unpaired $y \sim \pi_y^*$ |
|---|---|---|---|
| Regression | ✓ | ✗ | ✗ |
| UGAN + $\ell^2$ | ✓ | ✓ | ✓ |
| CGAN | ✓ | ✓ | ✗ |
| CondNF | ✓ | ✗ | ✗ |
| CondNF (SS) | ✓ | ✓ | ✓ |
| GNOT | ✓ | ✓ | ✓ |
| DCPEME | ✓ | ✓ | ✓ |
| parOT | ✓ | ✓ | ✓ |
| OTCS | ✓ | ✓ | ✓ |
| FSBM | ✓ | ✓ | ✓ |
| CGMM (SS) | ✓ | ✓ | ✓ |
| **Our method** | ✓ | ✓ | ✓ |

Table 5: The ability to use paired/unpaired data by various models.

**Cost Matrix.** Let $x \in \mathbb{R}^2$ and $y \in \mathbb{R}^2$ be points from the source and target distributions, respectively. Define the rotated vectors as

$$y^{\pm\varphi} = R_{\pm\varphi}(y) = \begin{bmatrix} \cos(\pm\varphi) & -\sin(\pm\varphi) \\ \sin(\pm\varphi) & \cos(\pm\varphi) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix},$$

where $\varphi$ is a given rotation angle, in our case, it's $\pm 90°$. The corresponding elements of mini-batch OT cost matrices are then

$$C_{ij}^{+\varphi} = \|x_i - y_j^{+\varphi}\|_2, \quad C_{ij}^{-\varphi} = \|x_i - y_j^{-\varphi}\|_2,$$

and the final cost matrix is

$$C_{ij} = \min\left(C_{ij}^{+\varphi}, C_{ij}^{-\varphi}\right), \quad \forall i, j.$$

In other words, each $x_i \sim \pi_x^*$ is mapped to a point $y_j$ on the opposite side of the Swiss Roll, rotated either by $+90°$ or $-90°$, depending on which distance is smaller.

## D.2 BASELINE DETAILS

This section details the loss functions employed by the baseline models, providing context and explanation for the data usage summarized in Table 5. Furthermore, it explains a straightforward adaptation of the log-likelihood loss function presented in (9) to accommodate unpaired data, offering a natural comparative approach to the method proposed in our work. Finally, it includes details about our reproduction of other methods and their discussion.

1. **Standard generative & predictive models:**
   - **Regression Model** (MLP) uses the following simple $\ell^2$ loss

     $$\min_\theta \mathbb{E}_{(x,y)\sim\pi^*} ||y - G_\theta(x)||^2,$$

     where $G_\theta : \mathcal{X} \to \mathcal{Y}$ is a generator MLP with trainable parameters $\theta$. Clearly, such a model can use only paired data. Furthermore, it is known that the optimal regressor $G^*$ coincides with $\mathbb{E}_{y\sim\pi^*(\cdot|x)} y$, i.e., predicts the conditional expectation. Therefore, such a model will never learn the true data distribution unless all $\pi^*(\cdot|x)$ are degenerate.
   - **Conditional GAN** uses the following $\min\max$ loss:

     $$\min_\theta \max_\phi \left[ \underbrace{\mathbb{E}_{x,y\sim\pi^*} \log\left(D_\phi(y|x)\right)}_{\text{Joint, requires pairs } (x,\,y)\,\sim\,\pi^*} + \underbrace{\mathbb{E}_{x\sim\pi_x^*}\mathbb{E}_{z\sim p_z(z)} \log\left(1 - D_\phi(G_\theta(z|x)|x)\right)}_{\text{Marginal, requires } x\,\sim\,\pi_x^*} \right],$$

     where $G_\theta : \mathcal{Z} \times \mathcal{X} \to \mathcal{Y}$ is the conditional generator with parameters $\theta$, $p_z$ is a distribution on latent space $\mathcal{Z}$, and $D_\phi : \mathcal{Y} \times \mathcal{X} \to (0, 1)$ is the conditional discriminator with parameters $\phi$. It is clear that the model can use not only paired data during the training, but also samples from $\pi_x^*$. The minimum of this loss is achieved when $G_\theta(\cdot|x)$ generates $\pi^*(\cdot|x)$ from $p_z$.
   - **Unconditional GAN + $\ell^2$ loss** optimizes the following $\min\max$ objective:

     $$\min_\theta \max_\phi \left[ \underbrace{\lambda\,\mathbb{E}_{(x,y)\sim\pi^*}\mathbb{E}_{z\sim p_z} ||y - G_\theta(x, z)||^2}_{\text{Joint, requires pairs } (x,\,y)\,\sim\,\pi^*} + \underbrace{\mathbb{E}_{y\sim\pi_y^*} \log\left(D_\phi(y)\right)}_{\text{Marginal, requires } y\,\sim\,\pi_y^*} + \underbrace{\mathbb{E}_{x\sim\pi_x^*}\mathbb{E}_{z\sim p_z} \log\left(1 - D_\phi(G_\theta(x, z))\right)}_{\text{Marginal, requires } x\,\sim\,\pi_x^*} \right],$$

where $\lambda > 0$ is a hyperparameter and $G_\theta : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$ is the stochastic generator. Compared to the unconditional case, the main idea here is to use the unconditional disctiminator $D_\phi : \mathcal{Y} \to (0, 1)$. This allows using unpaired samples from $\pi_y^*$. However, using only GAN loss would ignore the paired information in any form, this is why the supervised $\ell^2$ loss is added ($\lambda = 1$).

We note that this model has a trade-off between the target matching loss (GAN loss) and regression loss (which suffers from averaging). Hence, the model is unlikely to learn the true paired data distribution and can be considered as a heuristic loss for using both paired and unpaired data. Overall, we believe this baseline is representative of many existing GAN-based solutions (Tripathy et al., 2019, §3.5), (Jin et al., 2019, §3.3), (Yang & Chen, 2020, §C), (Vasluianu et al., 2021, §3), which use objectives that are *ideologically* similar to this one for paired and unpaired data.

- **Conditional Normalizing Flow** (Winkler et al., 2019) learns an explicit density model

$$\pi^\theta(y|x) = p_z(G_\theta^{-1}(y|x)) \left| \frac{\partial G_\theta^{-1}(y|x)}{\partial y} \right|$$

via optimizing log-likelihood (9) of the paired data. Here $G_\theta : \mathcal{Z} \times \mathcal{X} \to \mathcal{Y}$ is the conditional generator function. It is assumed that $\mathcal{Z} = \mathcal{Y}$ and $G_\theta(\cdot|x)$ is invertible and differentiable. In the implementation, we use the well-celebrated RealNVP neural architecture (Dinh et al., 2017). The optimal values are attained when the generator $G_\theta(\cdot|x)$ indeed generates $\pi^\theta(\cdot|x) = \pi^*(\cdot|x)$.

The conditional flow is expected to accurately capture the true conditional distributions, provided that the neural architecture is sufficiently expressive and there is an adequate amount of paired data available. However, as mentioned in §3.1, a significant challenge arises in integrating unpaired data into the learning process. For instance, approaches such as those proposed by (Atanov et al., 2019; Izmailov et al., 2020) aim to extend normalizing flows to a semi-supervised context. However, these methods primarily assume that the input conditions $x$ are discrete, making it difficult to directly apply their frameworks to our continuous case. For completeness, below we discuss a variant of the log-likelihood loss (Atanov et al., 2019, Eq. 1) when both $x, y$ are continuous.

2. **Semi-supervised log-likelihood** methods (Atanov et al., 2019; Izmailov et al., 2020):
   - **Semi-supervised Conditional Normalizing Flows**. As noted by the the authors, a natural strategy for log-likelihood semi-supervised training that leverages both paired and unpaired data is to optimize the following loss:

$$\max_\theta \Bigg[ \underbrace{\mathbb{E}_{(x,y)\sim\pi^*} \log\, \pi^\theta(y|x)}_{\text{Joint, requires pairs } (x,\, y)\, \sim\, \pi^*} + \underbrace{\mathbb{E}_{y\sim\pi_y^*} \log\, \pi^\theta(y)}_{\text{Marginal, requires } y\, \sim\, \pi_y^*} \Bigg]. \tag{31}$$

This straightforward approach involves adding the unpaired data component, $\mathbb{E}_{y\sim\pi_y^*} \log \pi^\theta(y)$ to the loss function alongside the standard paired data component (9). While loss (31) looks natural, its optimization is *highly non-trivial* since the marginal log-likelihood $\log \pi^\theta(y)$ is not directly available. In fact, (Atanov et al., 2019; Izmailov et al., 2020) use this loss exclusively in the case when $x$ is a discrete object, e.g., the class label $x \in \{1, 2, ..., K\}$. In this case $\log \pi^\theta(y)$ can be analytically computed:

$$\log \pi^\theta(y) = \log \mathbb{E}_{x\sim\pi_x^*} \pi^\theta(y|x) = \log \sum_{k=1}^{K} \pi^\theta(y|x = k)\pi_x^*(x = k),$$

and $\pi^*(x = k)$ are known class probabilities. Unfortunately, in the continuous case $\pi_x^*(x)$ is typically not available explicitly, and one has to exploit *approximations* such as

$$\log \pi^\theta(y) = \log \mathbb{E}_{x\sim\pi_x^*} \pi^\theta(y|x) \approx \log \frac{1}{Q} \sum_{q=1}^{Q} \log \pi^\theta(y|x_q),$$

where $x_q$ are train (unpaired) samples. However, such Monte-Carlo estimates are generally **biased** (because of the logarithm) and do not lead to good results, especially in high dimensions. Nevertheless, for completeness, we also test how this approach performs. In our 2D

example (Figure 2j), we found there is no significant difference between this loss and the fully supervised loss (9): both models incorrectly map to the target and fail to learn conditional distributions.

- **Semi-supervised Conditional Gaussian Mixture Model**. Using the natural loss (31) for semi-supervised learning, one could also consider a (conditional) Gaussian mixture parametrization for $\pi^\theta(y|x)$ instead of the normalizing flow. For completeness, we include this baseline for comparison. Using the same Gaussian mixture parametrization (17) as in our method, we observed that this loss quickly overfits and leads to degenerate solutions, see Figure 2e.

3. **Semi-supervised Methods.** These methods are designed to learn deterministic OT maps with general cost functions and, as a result, cannot capture stochastic conditional distributions.

   - **Neural optimal transport with pair-guided cost functional** (Asadulaev et al., 2024, GNOT). This method employs a general cost function for the neural optimal transport approach, utilizing a neural network parametrization for the mapping function and potentials. In our experiments, we focus on the paired cost function setup, enabling the use of both paired and unpaired data. We use the publicly available implementation[2], which has been verified through toy experiments provided in the repository.
   - **Differentiable cost-parameterized entropic mapping estimator** (Howard et al., 2024, DCPEME). We obtained the implementation from the authors but were unable to achieve satisfactory performance. This is likely due to the deterministic map produced by their method based on the entropic map estimator from (Cuturi et al., 2023). In particular, scenarios where nearby or identical points are mapped to distant locations may introduce difficulties, potentially leading to optimization stagnation during training.
   - **Parametric Pushforward Estimation With Map Constraints** (Panda et al., 2023, parOT)[3]. We evaluated this method using the $\ell_2$ cost function, where it performed as expected. However, on our setup, the method occurred unsuitable because it learns a fully deterministic transport map, which lacks the flexibility needed to model stochastic multimodal mapping. This limitation is visually evident in Figure 6g.
   - **Optimal Transport-guided Conditional Score-based diffusion model** (Gu et al., 2023, OTCS). We evaluated this method on a two-dimensional example from their GitHub repository[4], where it performed as expected. However, when applied to our setup (described in §5.1), the method failed to yield satisfactory results, even when provided with a large amount of training data (refer to Figure 6h and detailed in Appendix D.3).
   - **Feedback Schrödinger Bridge Matching** (Theodoropoulos et al., 2024, FSBM). We first tested the method on a two-dimensional example from their GitHub repository[5], where it performed as reported in the original paper. However, as shown in Figure 2o, the learned target distribution is very noisy with a small amount of data. With more samples (Figure 6i), the method approximates the target distribution better but still fails to capture the ground-truth conditional distribution, presumably due to misleading guidance from the key-points.

### D.3 Baselines for Swiss Roll with the Large Amount of Data (16K)

In this section, we show the results of training of the baselines on the large amount of both paired (16K) and unpaired (16K) data (Figure 6). Recall that the ground truth $\pi^*$ is depicted in Figure 2c.

As expected, Regression fails to learn anything meaningful due to the averaging effect (Figure 6a). In contrast, the unconditional GAN+$\ell^2$ (Figure 6b) nearly succeeds in generating the target data $\pi^*_y$, but the learned plan is still incorrect, also due to the averaging effect. Given a sufficient amount of training data, Conditional GAN (Figure 6c) nearly succeeds in learning the true conditional distributions $\pi^*(\cdot|x)$. The same applies to the conditional normalizing flow (Figure 6d), but its results are slightly worse, presumably due to the limited expressiveness of invertible flow architecture.

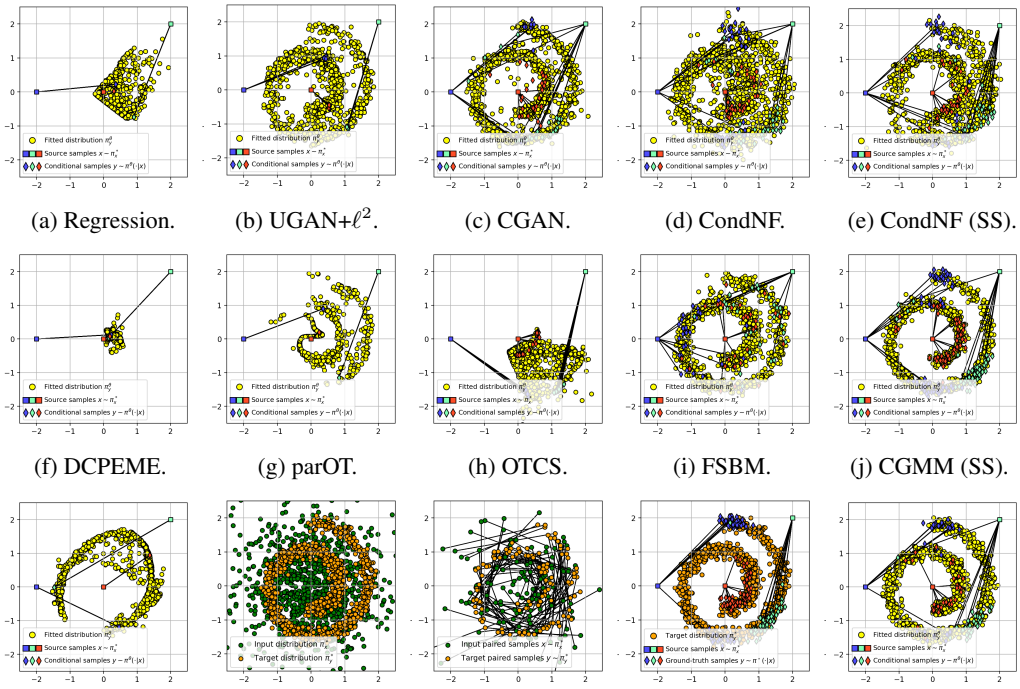
Experiments using the natural semi-supervised loss function in (31) demonstrate that this loss function can reasonably recover the conditional mapping with both CondNF (Figure 6e) and CGMM

[2] https://github.com/machinestein/GNOT

[3] https://github.com/natalieklein229/uq4ml/tree/parot

[4] https://github.com/XJTU-XGU/OTCS/

[5] https://github.com/panostheo98/FSBM



(a) Regression. (b) UGAN+$\ell^2$. (c) CGAN. (d) CondNF. (e) CondNF (SS).

(f) DCPEME. (g) parOT. (h) OTCS. (i) FSBM. (j) CGMM (SS).

(k) GNOT. (l) Unpaired data. (m) Paired data. (n) Ground truth. (o) **Our method.**

Figure 6: Comparison of the mapping learned by baselines on *Gaussian* $\rightarrow$ *Swiss Roll* task (§5.1). We use $P = 16K$ paired data, $Q = R = 16K$ unpaired data for training.

(Figure 6j) parameterizations. However, it requires significantly more training data compared to our proposed loss function (13). This conclusion is supported by the observation that the CGMM model trained with (31) tends to overfit, as shown in Figure 2e. In contrast, our method, which uses the objective (13), achieves strong results, as illustrated in Figure 2d.

Other methods, unfortunately, also struggle to handle this illustrative 2D task effectively, despite their success in large-scale problems. This discrepancy raises questions about the theoretical justification and general applicability of these methods, particularly in scenarios where simpler tasks reveal limitations not evident in more complex settings.

### D.4 Ablation Study

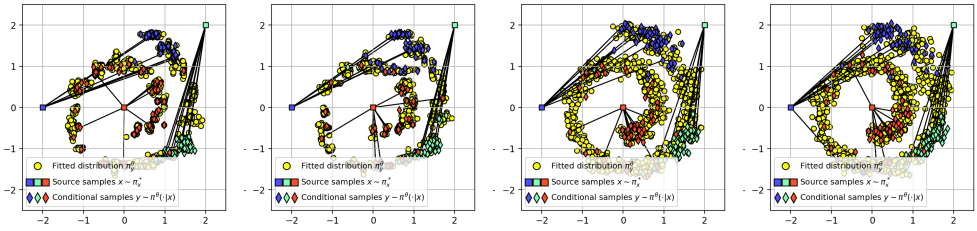
In this section, we conduct an ablation study to address the question posed in §3.1 regarding how the number of source and target samples influences the quality of the learned mapping. The results, shown in Figure 7, indicate that the quantity of target points $R$ has a greater impact than the number of source points $Q$ (compare Figure 7c with Figure 7b). Additionally, it is evident that the inclusion of unpaired data helps mitigate over-fitting, as demonstrated in Figure 7a.

## E Proofs

### E.1 Loss Derivation

Below, we present a step-by-step derivation of the mathematical transitions, allowing the reader to follow and verify the validity of our approach. We denote as $C_1, C_2$ all terms that are not involved in learning the conditional plan $\pi^\theta(y|x)$, i.e., not dependent on $\theta$ or marginal distributions such as $\pi^*_x$. Starting from (6), we deduce

$$\mathrm{KL}\left(\pi^* \| \pi^\theta\right) = \mathbb{E}_{x,y\sim\pi^*} \log \frac{\pi^*_x(x)\pi^*(y|x)}{\pi^\theta_x(x)\pi^\theta(y|x)} = \mathbb{E}_{x\sim\pi^*_x} \log \frac{\pi^*_x(x)}{\pi^\theta_x(x)} + \mathbb{E}_{x,y\sim\pi^*} \log \frac{\pi^*(y|x)}{\pi^\theta(y|x)} =$$




(a) $Q = 0, R = 0$ (b) $Q = 1024, R = 0$ (c) $Q = 0, R = 1024$ (d) $Q = 1024, R = 1024$

Figure 7: Ablation study analyzing the impact of varying source and target data point quantities on the learned mapping for the *Gaussian* $\rightarrow$ *Swiss Roll* task (using $P = 128$ paired samples).

$$\mathrm{KL}\left(\pi_x^*\|\pi_x^\theta\right) + \mathbb{E}_{x\sim\pi_x^*}\mathbb{E}_{y\sim\pi^*(\cdot|x)}\log\frac{\pi^*(y|x)}{\pi^\theta(y|x)} = \underbrace{\mathrm{KL}\left(\pi_x^*\|\pi_x^\theta\right)}_{\text{Marginal}} + \underbrace{\mathbb{E}_{x\sim\pi_x^*}\mathrm{KL}\left(\pi^*(\cdot|x)\|\pi^\theta(\cdot|x)\right)}_{\text{Conditional}} =$$

$$C_1 + \mathbb{E}_{x\sim\pi_x^*}\mathbb{E}_{y\sim\pi^*(\cdot|x)}\log\frac{\pi^*(y|x)}{\pi^\theta(y|x)} = C + \mathbb{E}_{x\sim\pi_x^*}\mathbb{E}_{y\sim\pi^*(\cdot|x)}\left[\log\pi^*(y|x) - \log\pi^\theta(y|x)\right] =$$

$$C_1 - \mathbb{E}_{x\sim\pi_x^*}\mathrm{H}\left(\pi^*(\cdot|x)\right) - \mathbb{E}_{x,y\sim\pi^*}\log\pi^\theta(y|x) = C_2 - \mathbb{E}_{x,y\sim\pi^*}\log\pi^\theta(y|x) \overset{(10)}{=}$$

$$C_2 - \mathbb{E}_{x,y\sim\pi^*}\log\frac{\exp\left(-E^\theta(y|x)\right)}{Z^\theta(x)} = C_2 + \mathbb{E}_{x,y\sim\pi^*}E^\theta(y|x) + \mathbb{E}_{x,y\sim\pi^*}\log Z^\theta(x) \overset{(12)}{=}$$

$$C_2 + \mathbb{E}_{x,y\sim\pi^*}\frac{c^\theta(x,y) - f^\theta(y)}{\varepsilon} + \mathbb{E}_{x,y\sim\pi^*}\log Z^\theta(x) =$$

$$C_2 + \varepsilon^{-1}\mathbb{E}_{x,y\sim\pi^*}[c^\theta(x,y)] - \varepsilon^{-1}\mathbb{E}_{x,y\sim\pi^*}f^\theta(y) + \mathbb{E}_{x,y\sim\pi^*}\log Z^\theta(x) =$$

$$C_2 + \varepsilon^{-1}\mathbb{E}_{x,y\sim\pi^*}[c^\theta(x,y)] - \varepsilon^{-1}\mathbb{E}_{y\sim\pi_y^*}\mathbb{E}_{x\sim\pi^*(\cdot|y)}f^\theta(y) + \mathbb{E}_{x\sim\pi_x^*}\mathbb{E}_{y\sim\pi^*(\cdot|x)}\log Z^\theta(x) =$$

$$C_2 + \varepsilon^{-1}\mathbb{E}_{x,y\sim\pi^*}[c^\theta(x,y)] - \varepsilon^{-1}\mathbb{E}_{y\sim\pi_y^*}f^\theta(y)\underbrace{\mathbb{E}_{x\sim\pi^*(\cdot|y)}1}_{=1} + \mathbb{E}_{x\sim\pi_x^*}\log Z^\theta(x)\underbrace{\mathbb{E}_{y\sim\pi^*(\cdot|x)}1}_{=1} =$$

$$C_2 + \varepsilon^{-1}\mathbb{E}_{x,y\sim\pi^*}[c^\theta(x,y)] - \varepsilon^{-1}\mathbb{E}_{y\sim\pi_y^*}f^\theta(y) + \mathbb{E}_{x\sim\pi_x^*}\log Z^\theta(x).$$

The mathematical derivation presented above demonstrates that our defined loss function (13) is essentially a framework for minimizing KL-divergence. In other words, when the loss (13) equals to $-C_2$, it implies that we have successfully recovered the true conditional plan $\pi^*$ in the KL sense.

### E.2 Expressions for the Gaussian Parametrization

*Proof of Proposition 3.1.* Our parametrization of the cost $c^\theta$ (15) and the dual potential $f^\theta$ (16) gives:

$$\exp\left(\frac{f^\theta(y) - c^\theta(x,y)}{\varepsilon}\right) = \exp\left(\log\sum_{n=1}^N w_n\mathcal{N}(y\,|\,b_n,\varepsilon B_n) + \log\sum_{m=1}^M v_m(x)\exp\left(\frac{\langle a_m(x),y\rangle}{\varepsilon}\right)\right)$$

$$= \sum_{m=1}^M\sum_{n=1}^N \frac{v_m(x)w_n}{\sqrt{\det\left(2\pi\varepsilon^{-1}B_n^{-1}\right)}}\exp\left(-\frac{1}{2}(y-b_n)^\top\frac{B_n^{-1}}{\varepsilon}(y-b_n) + \frac{\langle a_m(x),y\rangle}{\varepsilon}\right)$$

We now rewrite the expression inside the exponent, scaled by $-2\varepsilon$, using the symmetry of $B_n$, to cast it into a Gaussian mixture form:

$$(y-b_n)^\top B_n^{-1}(y-b_n) - 2\langle a_m(x),y\rangle = y^\top B_n^{-1}y - 2b_n^\top B_n^{-1}y + b_n^\top B_n^{-1}b_n - 2\langle a_m(x),y\rangle =$$

$$y^\top B_n^{-1}y - 2\underbrace{\left(b_n + B_n a_m(x)\right)^\top}_{\overset{\text{def}}{=}d_{mn}^\top(x)}B_n^{-1}y + b_n^\top B_n^{-1}b_n =$$

$$(y - d_{mn}(x))^\top B_n^{-1}(y - d_{mn}(x)) + b_n^\top B_n^{-1}b_n - d_{mn}^\top(x)B_n^{-1}d_{mn}(x).$$

Afterwards, we rewrite the last two terms:

$$b_n^\top B_n^{-1} b_n - d_{mn}^\top(x) B_n^{-1} d_{mn}(x) = b_n^\top B_n^{-1} b_n - (b_n + B_n a_m(x))^\top B_n^{-1}(b_n + B_n a_m(x)) =$$
$$\underbrace{b_n^\top B_n^{-1} b_n - b_n^\top B_n^{-1} b_n}_{=0} - b_n^\top \underbrace{B_n^{-1} B_n}_{=I} a_m(x) - a_m^\top(x) \underbrace{B_n B_n^{-1}}_{=I} b_n - a_m^\top(x) \underbrace{B_n B_n^{-1}}_{=I} B_n a_m(x) =$$
$$-a_m^\top(x) B_n a_m(x) - 2 b_n^\top a_m(x).$$

Finally, we get

$$\exp\left(\frac{f^\theta(y) - c^\theta(x,y)}{\varepsilon}\right) = \sum_{m=1}^{M}\sum_{n=1}^{N} \underbrace{w_n v_m(x) \exp\left(\frac{a_m^\top(x) B_n a_m(x) + 2 b_n^\top a_m(x)}{2\varepsilon}\right)}_{\stackrel{\text{def}}{=} z_{mn}(x)}$$
$$\cdot \underbrace{\frac{1}{\sqrt{\det\left(2\pi\varepsilon^{-1} B_n^{-1}\right)}} \exp\left(-\frac{1}{2}(y - d_{mn}(x))^\top \frac{B_n^{-1}}{\varepsilon}(y - d_{mn}(x))\right)}_{=\mathcal{N}(y \,|\, d_{mn}(x), \varepsilon B_n)},$$

and, since $\int_{\mathcal{Y}} \mathcal{N}(y \,|\, d_{mn}(x), \varepsilon B_n) \mathrm{d}y = 1$, the normalization constant simplifies to the sum of $z_{mn}(x)$:

$$Z^\theta(x) = \int_{\mathcal{Y}} \exp\left(\frac{f^\theta(y) - c^\theta(x,y)}{\varepsilon}\right) \mathrm{d}y$$
$$= \int_{\mathcal{Y}} \sum_{m=1}^{M}\sum_{n=1}^{N} z_{mn}(x) \mathcal{N}(y \,|\, d_{mn}(x), \varepsilon B_n) \mathrm{d}y = \sum_{m=1}^{M}\sum_{n=1}^{N} z_{mn}(x).$$

□

*Proof of Proposition 3.2.* Combining equations (10), (12) and derivation above, we seamlessly obtain the expression (17) needed for Proposition 3.2. □

### E.3 Gradient of our Loss for Energy-Based Modeling

*Proof of Proposition A.1.* Direct differentiation of (13) gives:

$$\frac{\partial}{\partial\theta}\mathcal{L}(\theta) = \varepsilon^{-1}\mathbb{E}_{x,y\sim\pi^*}\left[\frac{\partial}{\partial\theta} c^\theta(x,y)\right] - \varepsilon^{-1}\mathbb{E}_{y\sim\pi_y^*}\left[\frac{\partial}{\partial\theta} f^\theta(y)\right] + \mathbb{E}_{x\sim\pi_x^*}\left[\frac{\partial}{\partial\theta}\log Z^\theta(x)\right]. \quad (32)$$

Recalling expression for the normalization constant, the last term can be expressed as follows:

$$\mathbb{E}_{x\sim\pi_x^*}\left[\frac{1}{Z^\theta(x)}\frac{\partial}{\partial\theta} Z^\theta(x)\right] = \mathbb{E}_{x\sim\pi_x^*}\left[\frac{1}{Z^\theta(x)}\int_{\mathcal{Y}} \frac{\partial}{\partial\theta}\exp\left(\frac{f^\theta(y) - c^\theta(x,y)}{\varepsilon}\right)\mathrm{d}y\right] =$$
$$\mathbb{E}_{x\sim\pi_x^*}\left[\frac{1}{Z^\theta(x)}\int_{\mathcal{Y}} \frac{\frac{\partial}{\partial\theta}\left(f^\theta(y) - c^\theta(x,y)\right)}{\varepsilon}\exp\left(\frac{f^\theta(y) - c^\theta(x,y)}{\varepsilon}\right)\mathrm{d}y\right] =$$
$$\varepsilon^{-1}\mathbb{E}_{x\sim\pi_x^*}\left[\int_{\mathcal{Y}} \frac{\partial}{\partial\theta}\left(f^\theta(y) - c^\theta(x,y)\right) \underbrace{\left\{\frac{1}{Z^\theta(x)}\exp\left(\frac{f^\theta(y) - c^\theta(x,y)}{\varepsilon}\right)\right\}}_{\pi^\theta(y|x)} \mathrm{d}y\right].$$

From equation above we obtain:

$$\frac{\partial}{\partial\theta}\mathcal{L}(\theta) = \varepsilon^{-1}\Bigg\{\mathbb{E}_{x,y\sim\pi^*}\left[\frac{\partial}{\partial\theta} c^\theta(x,y)\right] - \mathbb{E}_{y\sim\pi_y^*}\left[\frac{\partial}{\partial\theta} f^\theta(y)\right]$$
$$+ \mathbb{E}_{x\sim\pi_x^*}\mathbb{E}_{y\sim\pi^\theta(y|x)}\left[\frac{\partial}{\partial\theta}\left(f^\theta(y) - c^\theta(x,y)\right)\right]\Bigg\},$$

which concludes the proof. □

### E.4 Universal Approximation

Our objective is to set up and use the very general universal approximation result in (Acciaio et al., 2024, Theorem 3.8). Hereinafter, we use the following notation that slightly abuse notation from the main text.

**Intra-Section Notation.** For any $D \in \mathbb{N}$ we denote the Lebesgue measure on $\mathbb{R}^D$ by $\lambda_D$, suppressing the subscript $D$ whenever clear from its context, we use $L^1_+(\mathbb{R}^D)$ to denote the set of Lebesgue integrable (equivalence class of) functions $f : \mathbb{R}^D \to \mathbb{R}$ for which $\int f(x)\,\lambda(dx) = 1$ and $f \geq 0$ $\lambda$-a.e; i.e. Lebesgue-densities of probability measures. We use $\mathcal{P}^+_1(\mathbb{R}^D)$ to denote the space of all Borel probability measures on $\mathbb{R}^D$ which are absolutely continuous with respect to $\lambda$, metrized by the total variation distance $d_{TV}$. For any $D \in \mathbb{N}$, we denote the set of $D \times D$ positive-definite matrices by $\mathrm{PD}_D$. Additionally, for any $N \in \mathbb{N}$, we define the $N$-simplex by $\Delta_N \overset{\text{def.}}{=} \{u \in [0,1]^N : \sum_{n=1}^N u_n = 1\}$. We also denote floor operation for any $x \in \mathbb{R}$ as $\lfloor x \rfloor \overset{\text{def.}}{=} \max\{n \in \mathbb{Z} | n \leq x\}$.

**Lemma E.1** (The Space $(\mathcal{P}^+_1(\mathbb{R}^D), d_{TV})$ is quantizable by Gaussian Mixtures)**.** *For every $N \in \mathbb{N}$, let $D_N \overset{\text{def.}}{=} \frac{N}{2}((D^2+3D+2))$ and define the map*

$$GMM_N : \mathbb{R}^{D_N} = \mathbb{R}^N \times \mathbb{R}^{ND} \times \mathbb{R}^{\frac{N}{2}D(D+1))} \to \mathcal{P}^+_1(\mathbb{R}^D)$$
$$\big(w, \{b_n\}_{n=1}^N, \{B_n\}_{n=1}^N\big) \mapsto \sum_{n=1}^N Proj_{\Delta_N}(w)_n\, \nu\big(b_n, \varphi(B_n)\big),$$

*where $Proj_{\Delta_N} : \mathbb{R}^N \mapsto \Delta_N$ is the $\ell^2$ orthogonal projection of $\mathbb{R}^N$ onto the $N$-simplex $\Delta_N$ and $\nu(b_n, \varphi(B_n))$ is the Gaussian measure on $\mathbb{R}^D$ with mean $b_n$, and non-singular covariance matrix given by $\varphi(B_n)$ where $\varphi : \mathbb{R}^{D(D+1)/2} \to \mathrm{PD}_D$ is given for each $B \in \mathbb{R}^{D(D+1)/2}$ by*

$$\varphi(B) \overset{\text{def.}}{=} \exp\left(\begin{pmatrix} B_1 & B_2 & \dots & B_D \\ B_2 & B_3 & \dots & B_{2D-1} \\ \vdots & \ddots & & \vdots \\ B_D & B_{2D-1} & \dots & B_{D(D+1)/2} \end{pmatrix}\right), \tag{33}$$

*where* exp *is the matrix exponential on the space of $D \times D$ matrices. Then, the family $(GMM_n)_{n=1}^\infty$ is a quantization of $(P^+_1(\mathbb{R}^D), d_{TV})$ in the sense of (Acciaio et al., 2024, Definition 3.2).*

*Proof.* As implied by (Arabpour et al., 2024, Equation (3.10) in Proposition 7) every Gaussian measure $\mathcal{N}(m, \Sigma) := \mu$ on $\mathbb{R}^D$ with mean $m \in \mathbb{R}^D$, symmetric positive-definite covariance matrix $\Sigma$ can be represented as

$$\mu = \mathcal{N}(m, \varphi(X)) \tag{34}$$

for some (unique) vector $X \in \mathbb{R}^{D(D+1)/2}$. Therefore, by definition of a quantization, see (Acciaio et al., 2024, Definition 3.2), it suffices to show that the family of Gaussian mixtures is dense in $(\mathcal{P}^+_1(\mathbb{R}^D), d_{TV})$.

Now, let $\nu \in \mathcal{P}^+_1(\mathbb{R}^D)$ be arbitrary. By definition of $\mathcal{P}^+_1(\mathbb{R}^D)$ the measure $\nu$ admits a Radon-Nikodym derivative $f \overset{\text{def.}}{=} \frac{D\mu}{D\lambda}$, with respect to the $D$-dimensional Lebesgue measure $\lambda$. Moreover, by the Radon-Nikodym theorem, $f \in L^1_\mu(\mathbb{R}^D)$; and by since $\mu$ is a probability measure then $\nu \in L^1_+(\mathbb{R}^D)$.

Since compactly-supported smooth functions are dense in $L^1_+(\mathbb{R}^D)$ then, for every $\varepsilon > 0$, there exists some $\tilde{f} \in C^\infty_c(\mathbb{R}^D)$ with $\tilde{f} \geq 0$ such that

$$\|f - \tilde{f}\|_{L^1(\mathbb{R}^D)} < \frac{\varepsilon}{3}. \tag{35}$$

Since $C^\infty_c(\mathbb{R}^D)$ is dense in $L^1(\mathbb{R}^D)$ then we may without loss of generality re-normalize $\tilde{f}$ to ensure that it integrates to 1.

Since $\tilde{f}$ is compactly supported and approximates $f$, then (if $f$ is non-zero, which it cannot be as it integrates to 1) then it cannot be analytic, and thus it is non-polynomial. For every $\delta > 0$, let $\varphi_\delta$

denote the density of the $D$-dimensional Gaussian probability measure with mean $0$ and isotropic covariance $\delta\, I_D$ (where $I_D$ is the $D \times D$ identity matrix). Therefore, the proof of (Pinkus, 1999, Proposition 3.7) (or any standard mollification argument) shows that we can pick $\delta \overset{\text{def.}}{=} \delta(\varepsilon) > 0$ small enough so that the convolution $\tilde{f} \star \varphi_\delta$ satisfies

$$\left\| \tilde{f} - \tilde{f} \star \varphi_\delta \right\|_{L^1(\mathbb{R}^D)} < \frac{\varepsilon}{3}. \tag{36}$$

Note that $\tilde{f} \star \varphi_\delta$ is the density of probability measure on $\mathbb{R}^D$; namely, the law of a random variable which is the sum of a Gaussian random variance with law $\mathcal{N}(0, \delta I_N)$ and a random variable with law $\mu$. That is, $\tilde{f} \star \varphi_\delta \lambda \in L^1_+(\mathbb{R}^D)$. Together (35) and (36) imply that

$$\left\| f - \tilde{f} \star \varphi_\delta \right\|_{L^1(\mathbb{R}^D)} < \frac{2\varepsilon}{3}. \tag{37}$$

Recall the definition of the convolution: for each $x \in \mathbb{R}^D$ we have

$$\tilde{f}(x) \star \varphi_\delta \overset{\text{def.}}{=} \int_{u \in \mathbb{R}^D} \tilde{f}(u) \varphi_\delta(x-u) \lambda(du). \tag{38}$$

Since $\tilde{f}, \varphi_\delta \in C^\infty_c(\mathbb{R}^D)$ then Lebesgue integral of their product coincides with the Riemann integral of their product; whence, there is an $N \overset{\text{def.}}{=} N(\varepsilon) \in \mathbb{N}$ "large enough" so that

$$\left\| \int_{u \in \mathbb{R}^D} \tilde{f}(u) \varphi_\delta(x-u) \lambda(du) - \sum_{n=1}^N \tilde{f}(u_n) \varphi_\delta(x-u_n) \lambda(du) \right\|_{L^1(\mathbb{R}^D)} < \frac{\varepsilon}{3} \tag{39}$$

for some $u_1, \dots, u_N \in \mathbb{N}$. Note that, $\sum_{n=1}^N \tilde{f}(u_n) \varphi_\delta(x-u_n)$ is the law of a Gaussian mixture. Therefore, combining (37) and (39) implies that

$$\left\| f - \sum_{n=1}^N \tilde{f}(u_n) \varphi_\delta(x-u_n) \lambda(du) \right\|_{L^1(\mathbb{R}^D)} < \varepsilon. \tag{40}$$

Finally, recalling that the total variation distance between two measures with integrable Lebesgue density equals the $L^1(\mathbb{R}^D)$ norm of the difference of their densities; yields the conclusion; i.e.

$$d_{TV}\big(\nu, \hat{\nu}\big) = \left\| f - \sum_{n=1}^N \tilde{f}(u_n) \varphi_\delta(x-u_n) \lambda(du) \right\|_{L^1(\mathbb{R}^D)} < \varepsilon$$

where $\frac{D\hat{\nu}}{D\lambda} \overset{\text{def.}}{=} \sum_{n=1}^N \tilde{f}(u_n) \varphi_\delta(x-u_n) \lambda(du)$. □

**Lemma E.2** (The space $(\mathcal{P}_1^+(\mathbb{R}^D), d_{TV})$ is Approximate Simplicial)**.** *Let* $\hat{\mathcal{Y}} \overset{\text{def.}}{=} \bigcup_{N \in \mathbb{N}} \Delta_N \times [\mathcal{P}_1^+(\mathbb{R}^D)]^N$ *and define the map* $\eta : \hat{\mathcal{Y}} \mapsto \mathcal{P}_1^+(\mathbb{R}^D)$ *by*

$$\eta(w, (r_n)_{n=1}^N) \overset{\text{def.}}{=} \sum_{n=1}^N w_n\, r_n.$$

*Then,* $\eta$ *is a mixing function, in the sense of (Acciaio et al., 2024, Definition 3.1). Consequentially,* $(\mathcal{P}_1^+(\mathbb{R}^D), \eta)$ *is approximately simplicial.*

*Proof.* Let $\mathcal{M}^+(\mathbb{R}^D)$ denote the Banach space of all finite signed measures on $\mathbb{R}^D$ with finite total variation norm $\|\cdot\|_{TV}$. Since $\|\cdot - \cdot\|_{TV} = d_{TV}$ when restricted to $\mathcal{P}_1^+(\mathbb{R}^D) \times \mathcal{P}_1^+(\mathbb{R}^D)$ and since $\|\cdot\|_{TV}$ is a norm, then the conclusion follows from (Acciaio et al., 2024, Example 5.1) and since $\mathcal{P}_1^+(\mathbb{R}^D)$ is a convex subset of $\mathcal{M}^+(\mathbb{R}^D)$. □

Together, Lemmata E.1 and E.2 imply that $(\mathcal{P}_1^+(\mathbb{R}^D), d_{TV}, \eta, Q)$ is a QAS space in the sense of (Acciaio et al., 2024, Definition 3.4), where $Q \overset{\text{def.}}{=} (GMM_M)_{M \in \mathbb{N}}$. Consequently, the following is a geometric attention mechanism in the sense of(Acciaio et al., 2024, Definition 3.5)

$$\hat{\eta} : \cup_{N \in \mathbb{N}} \Delta_N \times \mathbb{R}^{N \times D_M} \to \mathcal{P}_1^+(\mathbb{R}^D)$$

$$\Big(w, \big(v_m, (b_{mn})_{n=1}^N, (B_{mn})_{n=1}^N\big)_{m=1}^M\Big) \mapsto \sum_{n=1}^N w_n \sum_{m=1}^M Proj_{\Delta_M}(v_m)_n\, \nu\big(b_{mn}, \varphi(B_{mn})\big).$$

Before presenting our main theorem, we first introduce several definitions of activation functions that will be used in the theorem. These definitions, which are essential for completeness, are taken from (Acciaio et al., 2024, Definitions 2.2-2.4).

**Definition E.3** (Trainable Activation Function: Singular-ReLU Type)**.** A trainable activation function $\sigma$ is of *ReLU+Step type* if

$$\sigma_\alpha : \mathbb{R} \ni x \mapsto \alpha_1 \max\{x, \alpha_2 x\} + (1-\alpha_1)\lfloor x \rfloor \in \mathbb{R}$$

**Definition E.4** (Trainable Activation Function: Smooth-ReLU Type)**.** A trainable activation function $\sigma$ is of *smooth non-polynomial type* if there is a non-polynomial $\sigma^\star \in C_c^\infty(\mathbb{R})$,for which

$$\sigma_\alpha : \mathbb{R} \ni x \mapsto \alpha_1 \max\{x, \alpha_2 x\} + (1-\alpha_1)\sigma^\star(x) \in \mathbb{R}$$

**Definition E.5** (Classical Activation Function)**.** Let $\sigma^\star \in C_c^\infty(\mathbb{R})$ be non-affine and such that there is some $x \in \mathbb{R}$ at which $\sigma$ is differentiable and has non-zero derivative. Then $\sigma$ is is a classical regular activation function if,for every $\alpha \in \mathbb{R}^2$, $\sigma_\alpha = \sigma^\star$.

Further in the text, we assume that activation functions are applied element-wise to each vector $x \in \mathbb{R}^D$. We are now ready to prove the first part of our approximation theorem.

**Proposition E.6** (Deep Gaussian Mixtures are Universal Conditional Distributions in the TV Distance)**.** *Let $\pi : (\mathbb{R}^D, \|\cdot\|_2) \to (\mathcal{P}_1^+(\mathbb{R}^D), d_{TV})$ be Hölder. Then, for every compact subset $K \subseteq \mathbb{R}^D$, every approximation error $\varepsilon > 0$ there exists $M, N \in \mathbb{N}$ and a MLP $\hat{f} : \mathbb{R}^D \mapsto \mathbb{R}^{N \times N D_M}$ with activations as in Definitions E.3, E.4, E.5 such that the (non-degenerate) Gaussian-mixture valued map*

$$\hat{\pi}(\cdot|x) \overset{\text{def.}}{=} \hat{\eta} \circ \hat{f}(x)$$

*satisfies the uniform estimate*

$$\max_{x \in K} d_{TV}\big(\hat{\pi}(\cdot|x) \| \pi(\cdot|x)\big) < \varepsilon.$$

*Proof.* Since Lemmata E.2 and E.1 imply that $(\mathcal{P}_1^+(\mathbb{R}^D), d_{TV}, \eta, Q)$, is a QAS space in the sense of (Acciaio et al., 2024, Definition 3.4), then the conclusion follows directly from (Acciaio et al., 2024, Theorem 3.8). □

Since many of our results are formulated in the Kullback-Leibler divergence, then our desired guarantee is obtained only under some additional mild regularity requirements of the target conditional distribution $\hat{\pi}$ being approximated.

**Assumption E.7** (Regularity of Conditional Distribution)**.** Let $\pi : (\mathbb{R}^D, \|\cdot\|_2) \to (\mathcal{P}_1^+(\mathbb{R}^D), d_{TV})$ be Hölder and, for each $x \in \mathbb{R}^D$, $\pi(\cdot|x)$ is absolutely continuous with respect to the Lebesgue measure $\lambda$ on $\mathbb{R}^D$. Suppose that there exist some $0 < \delta \leq \Delta$ such that its conditional Lebesgue density satisfies

$$\delta \leq \frac{d\pi(\cdot|x)}{d\lambda} \leq \Delta \qquad \text{for all } x \in \mathbb{R}^D. \tag{41}$$

**Theorem E.8** (Deep Gaussian Mixtures are Universal Conditional Distributions)**.** *Suppose that $\pi$ satisfies Assumption E.7. Then, for every compact subset $K \subseteq \mathbb{R}^{D_x}$, every approximation error $\varepsilon > 0$ there exists $M, N \in \mathbb{N}$ such that: for each $m = 1, \dots, M$ and $n = 1, \dots, N$ there exist MLPs: $a_m : \mathbb{R}^{D_x} \mapsto \mathbb{R}^{D_y}, v_m : \mathbb{R}^{D_x} \mapsto \mathbb{R}^M$ with ReLU activation functions and $w_n, B_n$ learnable parameters such that the (non-degenerate) Gaussian-mixture valued map*

$$\hat{\pi}(\cdot|x) \overset{\text{def.}}{=} \sum_{n=1}^{N} \sum_{m=1}^{M} z_{mn}(x)\, \nu\big(d_{mn}(x), \varphi(D_{mn}(x))\big)$$

*satisfies the uniform estimate*

$$\max_{x \in K} d_{TV}\big(\pi(\cdot|x), \hat{\pi}(\cdot|x)\big) < \varepsilon. \tag{42}$$

*If, moreover, $\hat{\pi}$ also satisfies* (41) *(with $\hat{\pi}$ in place of $\pi$) then additionally*

$$\max_{x \in K} \mathrm{KL}\left(\pi(\cdot|x), \hat{\pi}(\cdot|x)\right) \in \mathcal{O}(\varepsilon), \tag{43}$$

*where $\mathcal{O}$ hides a constant independent of $\varepsilon$ and of the dimension $D$.*

The proof of Theorem E.8 makes use of the *symmetrized Kullback-Leibler divergence* $\mathrm{KL}_{sym}$ which is defined for any two $\alpha, \beta \in \mathcal{P}(\mathbb{R}^D)$ by $\mathrm{KL}_{sym}(\mu, \nu) \stackrel{\text{def.}}{=} \mathrm{KL}(\alpha\|\beta) + \mathrm{KL}(\beta\|\alpha)$; note, if $\mathrm{KL}_{sym}(\alpha, \beta) = 0$ then $\mathrm{KL}_{sym}(\alpha\|\beta) = 0$. We now prove our main approximation guarantee.

*Proof of Theorem E.8.* To simplify the explanation of our first claim, we provide the expression for $\hat{\pi}(y|x)$ from (17):

$$\hat{\pi}(y|x) = \sum_{n=1}^{N} w_n \sum_{m=1}^{M} v_m(x) \exp\left(\frac{a_m^\top(x) B_n a_m(x) + 2b_n^\top a_m(x)}{2\varepsilon}\right) \mathcal{N}(y \mid d_{mn}(x), \varepsilon B_n)$$

Thanks to the wide variety of activation functions available from Definitions E.3, E.4, E.5, we can construct the map $\hat{f}$ and directly apply Proposition E.6. This completes the proof of the first claim.

Under Assumption E.7, $\pi(\cdot|x)$ and $\hat{\pi}(\cdot|x)$ are equivalent to the $D$-dimensional Lebesgue measure $\lambda$. Consequently, for all $x \in \mathbb{R}^{D_x}$:

$$\pi(\cdot|x) \ll \hat{\pi}(\cdot|x)$$

Therefore, the Radon-Nikodym derivative $\frac{\hat{\pi}(\cdot|x)}{\pi(\cdot|x)}$ is a well-defined element of $L^1(\mathbb{R}^{D_x})$, for each $x \in \mathbb{R}^{D_x}$; furthermore, we have

$$\frac{\pi(\cdot|x)}{\hat{\pi}(\cdot|x)} = \frac{\pi(\cdot|x)}{d\lambda} \frac{d\lambda}{\hat{\pi}(\cdot|x)}. \tag{44}$$

Again, leaning on Assumption (41) and the Hölder inequality, we deduce that

$$\begin{aligned}
\sup_{a \in \mathbb{R}^D} \left|\frac{\pi(\cdot|x)}{\hat{\pi}(\cdot|x)}(a)\right| &= \sup_{a \in \mathbb{R}^D} \left|\frac{\pi(\cdot|x)}{d\lambda}(a) \frac{d\lambda}{\hat{\pi}(\cdot|x)}(a)\right| \\
&\leq \sup_{a \in \mathbb{R}^D} \left|\frac{\pi(\cdot|x)}{d\lambda}(a)\right| \sup_{a \in \mathbb{R}^D} \left|\frac{d\lambda}{\hat{\pi}(\cdot|x)}(a)\right| \\
&\leq \sup_{a \in \mathbb{R}^D} \left|\frac{\pi(\cdot|x)}{d\lambda}(a)\right| \frac{1}{\delta} \\
&\leq \frac{\Delta}{\delta}
\end{aligned} \tag{45}$$

where the final inequality under the assumption that $\hat{\pi}$ also satisfies Assumption 41. Importantly, we emphasize that the right-hand side of (45) holds *independently of* $x \in \mathbb{R}^{D_x}$ ("which we are conditioning on"). A nearly identical estimate holds for the corresponding lower-bound. Therefore, we may apply (Sason, 2015, Theorem 1) to deduce that: there exists a constant $C > 0$ (independent of $x \in \mathbb{R}^{D_x}$ and depending only on the quantities $\frac{\Delta}{\delta}$ and $\frac{\delta}{\Delta}$; thus only on $\delta, \Delta$) such that: for each $x \in \mathbb{R}^{D_x}$

$$\mathrm{KL}\left(\pi(\cdot|x), \hat{\pi}(\cdot|x)\right) \leq C\, d_{TV}\left(\pi(\cdot|x), \hat{\pi}(\cdot|x)\right). \tag{46}$$

The conclusion now follows, since the right-hand side of (46) was controllable by the first statement; i.e. since (42) holds we have

$$\mathrm{KL}\left(\pi(\cdot|x), \hat{\pi}(\cdot|x)\right) \leq C\, d_{TV}\left(\pi(\cdot|x), \hat{\pi}(\cdot|x)\right) \leq C\varepsilon. \tag{47}$$

A nearly identical derivation shows that

$$\mathrm{KL}\left(\hat{\pi}(\cdot|x), \pi(\cdot|x)\right) \leq C\varepsilon. \tag{48}$$

Combining (47) and (48) yields the following bound

$$\max_{x \in K} \mathrm{KL}_{sym}\left(\pi(\cdot|x), \hat{\pi}(\cdot|x)\right) \in \mathcal{O}(\varepsilon). \tag{49}$$

Since $\mathrm{KL}(\alpha\|\beta) \leq \mathrm{KL}_{sym}(\alpha, \beta)$ for every pair of Borel probability measures $\alpha$ and $\beta$ on $\mathbb{R}^{D_x}$ then (49) implies (43).

□


Figure 8: Extended visual comparisons between the FSBM (Theodoropoulos et al., 2024) method (3rd column) and our method (4th column) for Woman-to-Man translation are shown here. The task is described in §5.3, with further implementation details in Appendix C.4. The first column shows the source image and the second column the target image.


Figure 9: Visual comparisons for the Old-to-Young translation task between the FSBM (Theodoropoulos et al., 2024) method (3rd column) and our method (4th column). The task is described in Appendix C.4, which also provides additional implementation details. The first column displays the source image, and the second column shows the target image.