DYNAMIC NEURAL OPTIMAL TRANSPORT FOR SUBSET ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

We study optimal transport between distributions when only a subset of the target population is relevant to the source. The goal is to map the source distribution to a reweighted target distribution whose density ratio with respect to the full target lies in $[0, c]$. This constraint enables the transported distribution to select an appropriate target subpopulation while limiting how strongly any region of the target can be emphasized. We propose a dynamic neural optimal transport framework for this setting using two neural networks. The first parameterizes a time-dependent potential whose spatial gradient defines the velocity field and whose terminal value enforces the density-ratio constraint. The second parameterizes an interpolation between the source distribution and the learned terminal distribution. Unlike prior neural dynamic optimal transport methods, which typically assume a fixed terminal distribution, our formulation learns the terminal distribution within the support of the target. We demonstrate the method on positive-unlabeled learning, unpaired domain translation, and trajectory inference, where meaningful alignments exist between source and target distributions, but direct distribution matching is inappropriate because the target contains additional or irrelevant components.

1 INTRODUCTION

Distribution alignment is a problem of immense importance across machine learning, statistics, and optimization. Wasserstein distance and f-divergences are two widely used families of measures of disparity between distributions. Given source and target distributions μ and ν defined on subsets of a convex set $\Omega \subseteq \mathbb{R}^d$, specifically on $\mathrm{supp}(\mu)$ and $\mathrm{supp}(\nu)$, respectively, the objective of optimal transport is to obtain a mapping $T : \mathrm{supp}(\mu) \to \mathrm{supp}(\nu)$ from the source distribution to the target distribution that minimizes the expected transport cost, thereby characterizing the Wasserstein distance. This can be relaxed by mapping the source to a distribution near the target, using an f-divergence to quantify the disparity between the output of the map and the target distribution. In particular, we consider a dynamic deterministic formulation of T that uses a neural network to parameterize the velocity field. We also use a particular choice of f-divergence that constrains the mapped density to lie within a subset of $\mathrm{supp}(\nu)$ while satisfying a prescribed density-ratio bound. While previous work has studied relaxed neural optimal transport with static transport maps (Gazdieva et al., 2023; Choi et al., 2023; 2024a) and unbalanced dynamic optimal transport with mass growth (Neklyudov et al., 2023b), there is no existing neural dynamic optimal transport (OT) formulation that learns a map from the source distribution to a subset of the target distribution.

Specifically, our formulation of support subset alignment relies on a relaxation of the Benamou-Brenier problem, formulated in terms of a probability path ρ_t and velocity field $\mathbf{v}_t$ (Benamou & Brenier, 2000; Santambrogio, 2015; Figalli & Glaudo, 2023). For probability measures $\mu, \nu \in \mathcal{P}(\Omega)$, the semi-relaxed dynamic OT problem we study is

$$\inf_{\rho_t, \mathbf{v}_t} \int_0^1 \mathbb{E}_{\mathbf{x}\sim\rho_t}\left[\frac{1}{2}\|\mathbf{v}_t(\mathbf{x})\|^2\right] dt, \quad \text{s.t.} \quad \partial_t \rho_t(\mathbf{x}) + \mathrm{div}(\rho_t(\mathbf{x})\mathbf{v}_t(\mathbf{x})) = 0,\ \rho_0 = \mu,\ \rho_1 \leq c\nu, \tag{1}$$

where the initial marginal constraint $\rho_0 = \mu$ is preserved, whereas the terminal marginal ρ_1 is allowed to vary subject to the density-ratio constraint $\rho_1/\nu \in [0, c]$. This allows some regions of $\mathrm{supp}(\nu)$ to be up-weighted while others are down-weighted or removed. The relaxed terminal constraint is equivalent to a partial optimal transport relaxation of static optimal transport using f-divergences,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

as introduced by Séjourné et al. (2023). The case $c = 1$ corresponds to prescribing the endpoint marginal as $\rho_1 = \nu$. Under suitable regularity, weak solutions of the continuity equation induce deterministic flows via mass conservation (Villani et al., 2009, p. 15), while locally Lipschitz flows induce unique probability paths (Villani, 2021, Theorem 5.34).

In practice, when only samples are available, we use neural networks to parameterize both the velocity field $\mathbf{v}_t$ and the probability path ρ_t. Dynamic neural OT is closely connected to flow matching, diffusion-type models, trajectory inference, and generative modeling (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023; Tong et al., 2020; Neklyudov et al., 2023a;b).

To position our formulation, we introduce necessary notation and discuss unbalanced optimal transport (UOT) and the dynamic version. Let $\mathcal{P}(\mathcal{X})$ and $\mathcal{M}_+(\mathcal{X})$ denote probability and finite nonnegative measures on $\mathcal{X} \subseteq \Omega$. For a map S, $S_{\#}\mu$ is the pushforward; for $\pi \in \mathcal{M}_+(\mathcal{X} \times \mathcal{Y})$, $(\mathrm{pr}_{\mathcal{X}})_{\#}\pi$ and $(\mathrm{pr}_{\mathcal{Y}})_{\#}\pi$ are its marginals. We write $\tilde{\nu} \ll \nu$ if $\nu(A) = 0 \Rightarrow \tilde{\nu}(A) = 0$. When absolutely continuous, we use the same symbol for a measure and its density; thus, if $d\tilde{\nu} = \tilde{\nu}(\mathbf{x})d\mathbf{x}$ and $d\nu = \nu(\mathbf{x})d\mathbf{x}$, then $d\tilde{\nu}/d\nu = \tilde{\nu}/\nu$ on $\{\nu > 0\}$. We use ρ_t, $\mathbf{v}_t$, and g_t for time-varying measures, velocity fields, and growth fields, respectively, and $\mathbb{E}$ for expectations under probability measures.

Unbalanced optimal transport (UOT) generalizes balanced OT (Santambrogio, 2015) by allowing transport between finite nonnegative measures $\mu \in \mathcal{M}_+(\mathcal{X})$ and $\nu \in \mathcal{M}_+(\mathcal{Y})$ for which $\mu(\mathcal{X})$ and $\nu(\mathcal{Y})$ need not be equal. Let $f_1, f_2 : \mathbb{R}_+ \to \mathbb{R} \cup \{+\infty\}$ be proper, convex, lower semicontinuous functions satisfying $f_1(1) = f_2(1) = 0$. Given $\pi \in \mathcal{M}_+(\mathcal{X} \times \mathcal{Y})$, define $\tilde{\mu} = (\mathrm{pr}_{\mathcal{X}})_{\#}\pi$ and $\tilde{\nu} = (\mathrm{pr}_{\mathcal{Y}})_{\#}\pi$. The UOT problem is then stated as (Frogner et al., 2015; Séjourné et al., 2023)

$$\mathcal{W}_{\mathrm{UOT}}^{\{f_1,f_2\}}(\mu,\nu) = \inf_{\pi \in \mathcal{M}_+(\mathcal{X}\times\mathcal{Y})} \int_{\mathcal{X}\times\mathcal{Y}} \mathtt{c}(\mathbf{x},\mathbf{y})\, d\pi(\mathbf{x},\mathbf{y}) + \gamma_1 \mathcal{D}_{f_1}(\tilde{\mu}\|\mu) + \gamma_2 \mathcal{D}_{f_2}(\tilde{\nu}\|\nu), \quad (2)$$

where $\gamma_1, \gamma_2 > 0$, and $\mathcal{D}_{f_1}$ and $\mathcal{D}_{f_2}$ denote the f-divergences corresponding to f_1 and f_2, respectively. Even when μ and ν are probability measures, the optimal solution of equation 2 is generally not a coupling of μ and ν. Thus, UOT relaxes the strict marginal equality constraints of OT by replacing them with f-divergence penalties, making it applicable to general finite nonnegative measures. UOT has also been used to estimate transport maps T using neural network parameterizations (Yang & Uhler, 2019; Choi et al., 2023; 2024a; Gazdieva et al., 2023; 2024).

Dynamic UOT handles general positive measures by explicitly modeling mass creation and destruction through a scalar growth field g_t (Chizat et al., 2018b;a), and has been applied to problems in generative modeling and trajectory inference (Neklyudov et al., 2023a;b; 2024). Dynamic UOT between $\mu, \nu \in \mathcal{M}_+$ is

$$\inf_{\rho_t, \mathbf{v}_t, g_t} \int_0^1 \int_\Omega \left[\frac{\|\mathbf{v}_t(\mathbf{x})\|^2 + g_t^2(\mathbf{x})}{2}\right] d\rho_t(\mathbf{x})dt, \text{ s.t. } \frac{\partial}{\partial t}\rho_t(\mathbf{x}) + \mathrm{div}(\rho_t(\mathbf{x})\mathbf{v}_t(\mathbf{x})) = g_t(\mathbf{x})\rho_t(\mathbf{x}),\ \rho_0 = \mu,\ \rho_1 = \nu. \quad (3)$$

Thus, UOT penalizes deviations from prescribed marginals, whereas dynamic UOT modifies the continuity equation through a mass growth term. When μ and ν are probability measures and mass creation/destruction is not needed, one may set $g_t = 0$, recovering balanced dynamic OT.

Positioning and contributions. Unlike UOT, which allows positive measures and relaxes marginal constraints, and dynamic UOT, which models mass creation/destruction but preserves marginals, our dynamic subset alignment preserves the source mass and transports it to a terminal distribution supported on a subset of the target support and within a density-ratio bound of the target. This is achieved by replacing the terminal marginal equality constraint with an f-divergence-based penalty, which recovers balanced OT as a special case. We derive a dual dynamic formulation whose potential identifies the selected target subset and whose gradient defines the transport velocity. We use neural network-based parameterizations of the velocity field and path distribution to achieve the alignment and subset selection. Furthermore, we demonstrate the framework on unpaired domain translation, PU-learning, and trajectory inference problems.

2 Related Work

Our work on dynamic subset alignment is most directly related to Lagrangian neural optimal transport (Pooladian et al., 2024), action matching (Neklyudov et al., 2023a), and Wasserstein Lagrangian flows (Neklyudov et al., 2024). The neural optimal transport framework with Lagrangian costs (Pooladian et al., 2024) studies optimal transport with general Lagrangian potentials in Euclidean space.

Wasserstein Lagrangian flows (Neklyudov et al., 2024) are primarily developed for applications in cellular trajectory inference and quantum many-body problems and extend to more general settings such as the Wasserstein-Fisher-Rao metric (Chizat et al., 2018a;b; Séjourné et al., 2023), with the ability to model mass growth/destruction and different types of dynamics.

These approaches, as well as standard flow-based models, are developed for settings in which the endpoint marginals are prescribed, which prevents their direct application to the subset-alignment problem. A more extensive review of recent work on dynamic neural optimal transport is included in Appendix C.2. In contrast, our proposed dynamic support subset-selection framework preserves one marginal while dynamically transporting its mass to a subset of the support of the other marginal while minimizing the transport cost. Therefore, our approach can be viewed as a novel extension of prior work (Neklyudov et al., 2024). Although we focus on the ℓ_2^2 cost, our method is compatible with other Lagrangian costs (Pooladian et al., 2024), which could be useful for incorporating side information, as in semi-supervised domain adaptation.

Additionally, we review advances in static neural optimal transport in Appendix C.1. In Appendix A.2, we also discuss the static formulation of subset alignment. The static formulation of our framework can be related to incomplete optimal transport (IOT) (Gazdieva et al., 2023) and unbalanced optimal transport mapping (UOTM) (Choi et al., 2023). IOT (Gazdieva et al., 2023) can be interpreted as a proximal form of static subset alignment. UOTM can also be reduced to static subset alignment between two probability measures by choosing, in equation 2, f_1 as the f-divergence corresponding to the indicator function $\imath_{\{1\}}(x) = \begin{cases} 0, & x = 1, \\ +\infty, & x \neq 1, \end{cases}$ and $f_2 = \imath_{[0,c]}$.

3 METHODOLOGY

In our semi-relaxed dynamic OT formulation equation 1, we can replace the inequality $\rho_1 \leq c\nu$ constraint by adding the equivalent f-divergence penalty term $\mathcal{D}_{\imath_{[0,c]}}$ to the objective, where $\mathcal{D}_{\imath_{[0,c]}}$ is a special case of the range divergence $\mathcal{D}_{\imath_{[a,b]}}$ corresponding to the convex indicator function $\imath_{[a,b]}$, which, along with its Legendre–Fenchel conjugate $\imath^*_{[a,b]}$, is given as

$$\imath_{[a,b]}(r) = \begin{cases} 0, & r \in [a,b] \\ +\infty, & \text{o.w.} \end{cases}, \qquad \imath^*_{[a,b]}(t) = \sup_{u \in [a,b]} (u \cdot t) = \max(-at, bt). \tag{4}$$

Setting $a = 0$ and $b = c$, one obtains $\mathcal{D}_{\imath_{[0,c]}}(\rho_1 \| \nu)$ with $\imath^*_{[0,c]}(u) = c \cdot \max(0, u)$. Since $\imath_{[0,c]}$ is convex and lower semicontinuous, $\imath_{[0,c]} = \imath^{**}_{[0,c]}$. The variational form of f-divergences (Nguyen et al., 2010; Nowozin et al., 2016), (Polyanskiy & Wu, 2024, Theorem 7.26) $\mathcal{D}_f(\tilde{\nu} \| \nu) = \sup_{\eta:\Omega\to\mathbb{R}} \int_\Omega \eta(\mathbf{x}) d\tilde{\nu}(\mathbf{x}) - \int_\Omega f^*(\eta(\mathbf{x})) d\nu(\mathbf{x})$, yields

$$\mathcal{D}_{\imath_{[0,c]}}(\rho_1 \| \nu) = \sup_{\eta:\Omega\to\mathbb{R}} \mathop{\mathbb{E}}_{\mathbf{x}\sim\rho_1} [\eta(\mathbf{x})] - \mathop{\mathbb{E}}_{\mathbf{x}\sim\nu} [c \cdot \max(0, \eta(\mathbf{x}))]\,. \tag{5}$$

To solve the relaxed dynamic optimal transport problem in equation 1, we replace the terminal inequality constraint with the additive penalty $\mathcal{D}_{\imath_{[0,c]}}(\rho_1 \| \nu)$ in the objective. We introduce the Lagrange multiplier φ_t for the continuity equation and the Lagrange multiplier ψ_0 for the initial marginal constraint $\rho_0 = \mu$. After further simplification (see Appendix A.3 for details), one obtains the conditions.

$$\mathbf{v}_t(\mathbf{x}) = \nabla\varphi_t(\mathbf{x}),\ \psi_0(\mathbf{x}) = \varphi_0(\mathbf{x}),\ \eta(\mathbf{x}) = -\varphi_1(\mathbf{x}). \tag{6}$$

These conditions can then be used to eliminate $\mathbf{v}_t(\mathbf{x})$, $\psi_0(\mathbf{x})$, and $\eta(\mathbf{x})$, yielding the optimization problem $\sup_{\rho_t} \inf_{\varphi_t} \mathcal{L}(\rho_t, \varphi_t)$ with the Lagrangian

$$\mathcal{L}(\rho_t, \varphi_t) = \overbrace{\mathop{\mathbb{E}}_{\mathbf{x}\sim\mu}[\varphi_0(\mathbf{x})] + \mathop{\mathbb{E}}_{\mathbf{x}\sim\nu}[c \cdot \max(0, -\varphi_1(\mathbf{x}))]}^{\text{boundary part}} + \overbrace{\int_0^1 \mathop{\mathbb{E}}_{\mathbf{x}_t\sim\rho_t} \left[\frac{\partial}{\partial t}\varphi_t(\mathbf{x}_t) + \frac{\|\nabla\varphi_t(\mathbf{x}_t)\|^2}{2}\right] dt}^{\text{intermediate part}}\,. \tag{7}$$

To approximate the solution, we directly parameterize the dual potential φ_t with a neural network and parameterize the probability path ρ_t using a neural network that acts on samples from the terminal marginals μ and ν. A similar parameterization is used in the computational framework for learning Wasserstein–Lagrangian flows (WLF) (Neklyudov et al., 2024). In contrast, flow-based models

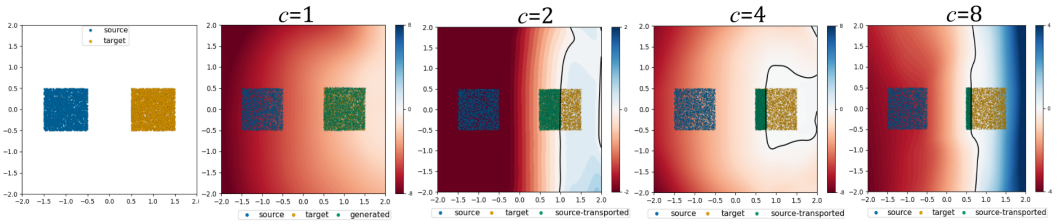


Figure 1: Subset alignment between two uniform distributions in $\mathbb{R}^2$ at $t = 1$, obtained using Algorithm 1 with $c \in \{1, 2, 4, 8\}$. The black contour line indicates the zero-level set of $\varphi_1(\mathbf{x})$.

typically do not learn ρ_t directly, but instead use analytically defined interpolants between two fixed marginals during training (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023). Our problem differs from both flow matching and WLF because we aim to map the source distribution μ to an unknown distribution ρ_1^* supported on a subset of $\mathrm{supp}(\nu)$. To this end, for $t \in [0, 1]$, we construct an interpolant $\mathbf{x}_t \sim \rho_t$ from independently sampled points $\mathbf{x}_0 \sim \mu$ and $\mathbf{x}_1 \sim \nu$ as

$$\mathbf{x}_t = (1 - t)\mathbf{x}_0 + t\left(\mathbf{x}_1 + Q_t(\mathbf{x}_0, \mathbf{x}_1)\right), \tag{8}$$

where Q_t is a time-dependent neural network. At $t = 1$, we have $\mathbf{x}_1 + Q_1(\mathbf{x}_0, \mathbf{x}_1) \sim \rho_1$. Since in general, $\rho_1 \neq \nu$, the network must learn a new terminal marginal. Let $\Phi_t : \Omega \to \Omega$ denote the flow induced by the velocity field $\mathbf{v}_t$, so that $(\Phi_t)_{\#}\mu = \rho_t$ and $\Phi_t(\mathbf{x}_0) = \mathbf{x}_0 + \int_0^t \mathbf{v}_\tau(\mathbf{x}_\tau)\, d\tau, \qquad \mathbf{x}_0 \in \mathrm{supp}(\mu)$. The optimal correction $Q_t^\star$ corresponding to this flow is then given by $Q_1^\star(\mathbf{x}_0, \mathbf{x}_1) = \Phi_1(\mathbf{x}_0) - \mathbf{x}_1$ and $Q_t^\star(\mathbf{x}_0, \mathbf{x}_1) = \frac{1}{t}\Big(\Phi_t(\mathbf{x}_0) - (1 - t)\mathbf{x}_0\Big) - \mathbf{x}_1$. By the standard neural-network universal approximation theorem (Hornik, 1991), bounds on the ϵ-approximation error $\|Q_t - Q_t^\star\|$ translate into bounds on $\|\Phi_t - \Phi_t^\star\|$. Consequently, for any admissible velocity field, one obtains $W_2((\Phi_t)_{\#}\mu, \rho_t) \leq C\epsilon$ for some constant $C > 0$.

Range-based penalty 5 also allows us to distinguish between data points sampled from the selected and rejected subsets of the target support $\mathrm{supp}(\nu)$. Let $\Phi_t^\star$ denote the optimal flow operator corresponding to the optimal dual potential $\varphi_t^\star$. Then points sampled from the selected subset $\mathrm{supp}((\Phi_1^\star)_{\#}\mu)$ and from the rejected subset $\mathrm{supp}(\nu) \setminus \mathrm{supp}((\Phi_1^\star)_{\#}\mu)$ can be separated by the following test

$$\begin{aligned} &\varphi_1^\star(\mathbf{x}) \leq 0, \quad \text{almost surely on } \mathrm{supp}((\Phi_1^\star)_{\#}\mu), \\ &\varphi_1^\star(\mathbf{x}) > 0, \quad \text{almost surely on } \mathrm{supp}(\nu) \setminus \mathrm{supp}((\Phi_1^\star)_{\#}\mu). \end{aligned} \tag{9}$$

Test 9 follows directly from the optimality conditions for the range penalty and dual φ_t (see Appendix B) and is used in our PU-learning and melanoma experiments.

4 Experiments and Results

We present experimental results for subset alignment on an easily interpretable image-to-image task, where the MNIST dataset (Deng, 2012) is the source and the EMNIST dataset (Cohen et al., 2017) is the target. In this setting, digit images form a subset of the characters in EMNIST. We also applied dynamic susbet alignment to the PU-Learning task. We then apply our proposed approach to domain translation on the FFHQ dataset (Karras et al., 2019) in the 512-dimensional latent space of ALAE (Pidhorskyi et al., 2020). We use Algorithm 1 to solve all the above discussed subset alignment problems. We also extended dynamic subset alignment to a multi-marginal setting and applied it melanoma dataset from Bunne et al. (2023).

4.1 Toy Examples

First, we apply subset alignment to 1D and 2D point distributions. For the 1D toy example, we align a uniform distribution with a 1D semicircular distribution. Results for $c \in \{1, 2, 4, 8\}$ are shown in Figure 2. As the parameter c increases, the source distribution μ is transported to a subset of the support of the target distribution ν, for which the potential satisfies $\varphi_1 \leq 0$. The transport between two 2D uniform distributions for $c \in \{1, 2, 4, 8\}$ is shown in Figure 1, where one can observe that $\varphi_1(\boldsymbol{x})$ is non-positive in selected subsets. Additional toy examples are provided in Appendix D, including subset alignment between two 1D uniform distributions and between two bimodal 2D distributions.

Algorithm 1: `Dynamic-Neural-SS`: Learning Algorithm for Dynamic Subset Selection

Require: Samples from distributions μ and ν, time-dependent neural network $\varphi_t(\cdot;\theta_\varphi)$, interpolant network $Q_t(\cdot,\cdot;\theta_\rho)$, batch size N, number of updates n_φ and n_ρ, gradient-based optimizers optim_φ and optim_ρ.

Ensure: Sample-based neural estimate for $\varphi_t(\cdot;\theta_\varphi)$.

for learning iteration $k = 0, 1, \ldots$ **do**
  for $u = 1, \ldots, n_\varphi$ **do**
    Sample $\{\mathbf{x}_0^i\}_{i=1}^N \sim \mu$, $\{\mathbf{x}_1^i\}_{i=1}^N \sim \nu$, and $\{t^i\}_{i=1}^N \sim \mathrm{Uniform}([0,1])$.
    $\{\mathbf{x}_t^i \leftarrow (1-t^i)\mathbf{x}_0^i + t^i(\mathbf{x}_1^i + Q_{t^i}(\mathbf{x}_0^i, \mathbf{x}_1^i;\theta_\rho))\}_{i=1}^N$.
    $\mathrm{grad}_{\theta_\varphi} \leftarrow$

$$\nabla_{\theta_\varphi} \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial}{\partial t}\varphi_{t^i}(\mathbf{x}_t^i;\theta_\varphi) + \frac{\|\nabla\varphi_{t^i}(\mathbf{x}_t^i;\theta_\varphi)\|^2}{2} + \varphi_0(\mathbf{x}_0^i;\theta_\varphi) + c\max\Big(0, -\varphi_1(\mathbf{x}_1^i;\theta_\varphi)\Big)\right].$$

    Using optim_φ and $\mathrm{grad}_{\theta_\varphi}$ to update θ_φ via gradient descent.
  end for
  for $u = 1, \ldots, n_\rho$ **do**
    Sample $\{\mathbf{x}_0^i\}_{i=1}^N \sim \mu$, $\{\mathbf{x}_1^i\}_{i=1}^N \sim \nu$, and $\{t^i\}_{i=1}^N \sim \mathrm{Uniform}([0,1])$.
    $\{\mathbf{x}_t^i \leftarrow (1-t^i)\mathbf{x}_0^i + t^i(\mathbf{x}_1^i + Q_{t^i}(\mathbf{x}_0^i, \mathbf{x}_1^i;\theta_\rho))\}_{i=1}^N$.

$$\mathrm{grad}_{\theta_\rho} \leftarrow \nabla_{\theta_\rho} \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial}{\partial t}\varphi_{t^i}(\mathbf{x}_t^i(\theta_\rho);\theta_\varphi) + \frac{\|\nabla\varphi_{t^i}(\mathbf{x}_t^i(\theta_\rho);\theta_\varphi)\|^2}{2}\right].$$

    Update θ_ρ using optim_ρ and $\mathrm{grad}_{\theta_\rho}$ via gradient ascent.
  end for
end for

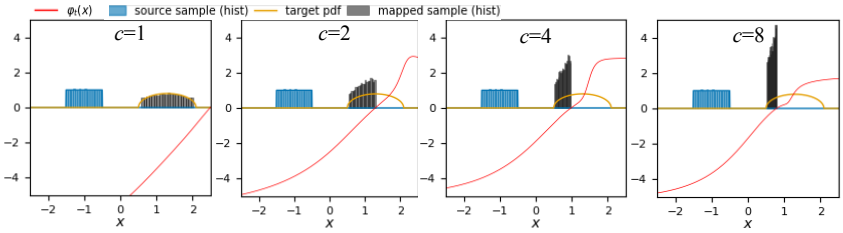


Figure 2: Source, target, and transported source histograms with sampling densities for 1D subset alignment between a uniform distribution and a semicircular density. φ_1 is shown in red.

4.2 MNIST to EMNIST Domain Translation

The MNIST dataset contains 60,000 images of digits from 0 to 9 in the training set and 10,000 images in the test set. The MNIST dataset is roughly balanced, in the sense that the proportions of the classes are approximately the same. The EMNIST ByClass (EMNIST) dataset contains English letters and digits. EMNIST consists of 62 imbalanced classes, of which 10 classes correspond to digits between 0–9, while the remaining 52 classes correspond to uppercase and lowercase letters of the English alphabet. Roughly, 16% of EMNIST consists of digits, while the remaining 84% consists of letters.

Since our goal is to transfer MNIST images to EMNIST images in such a way that MNIST digits are mapped to EMNIST digits while ignoring letters, we trained a neural network classifier to distinguish between digits and letters and used it to evaluate the learned mapping; see Appendix E.2 for implementation details. After training the classifier, we used both static and dynamic subset-selection approaches for domain translation between MNIST and EMNIST. Implementation details of the underlying models and their training are provided in Appendix E.3. We evaluated classification

Table 1: Classification accuracies of translated images for MNIST→EMNIST.

Method	$c=1$	$c=2$	$c=4$	$c=8$
static (Gazdieva et al., 2023)	46.93	75.33	82.44	87.32
dynamic (one step)	64.85	75.16	93.57	95.84
dynamic (ODE)	58.80	70.47	92.68	95.00

accuracy on translations of the entire MNIST dataset; the corresponding confusion matrices are provided in Appendix F. Accuracies of the translated outputs are in Table 1.

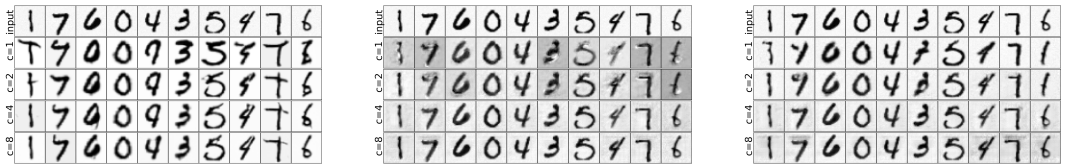
In our experiments, we trained and evaluated both static and dynamic models using the corresponding subset-selection frameworks for $c \in \{1, 2, 4, 8\}$. Similar to flow-based generative models, dynamic subset alignment requires numerical integration; in our experiments, we use ODE integration with an

Table 2: Comparison of average balanced accuracies of 20 models trained using dynamic subset alignment methods with PUSB and NTC-MI reported by Teisseyre et al. (2025), and our formulation of incomplete extremal optimal transport from Gazdieva et al. (2023) (labeled static). The best methods are colored red and second best blue.

Dataset	PUSB	NTC-MI	static		dynamic	
			sorted	sign	sorted	sign
Abalone	0.544 ± 0.060	0.575 ± 0.025	0.561 ± 0.029	0.503 ± 0.008	0.578 ± 0.025	0.540 ± 0.054
Banknote	0.829 ± 0.050	0.922 ± 0.019	0.883 ± 0.037	0.882 ± 0.039	0.952 ± 0.022	0.939 ± 0.018
Breast-w	0.766 ± 0.145	0.870 ± 0.028	0.930 ± 0.028	0.941 ± 0.027	0.927 ± 0.024	0.931 ± 0.026
Diabetes	0.546 ± 0.042	0.700 ± 0.039	0.635 ± 0.044	0.629 ± 0.044	0.629 ± 0.043	0.630 ± 0.046
Haberman	0.513 ± 0.023	0.532 ± 0.066	0.539 ± 0.066	0.540 ± 0.067	0.544 ± 0.074	0.540 ± 0.075
Heart	0.527 ± 0.033	0.757 ± 0.053	0.637 ± 0.093	0.623 ± 0.089	0.656 ± 0.080	0.653 ± 0.094
Ionosphere	0.440 ± 0.085	0.755 ± 0.059	0.773 ± 0.091	0.762 ± 0.088	0.827 ± 0.066	0.807 ± 0.070
Isolet	0.793 ± 0.072	0.725 ± 0.006	0.881 ± 0.028	0.923 ± 0.030	0.731 ± 0.140	0.707 ± 0.182
Jm1	0.628 ± 0.016	0.628 ± 0.013	0.576 ± 0.015	0.575 ± 0.010	0.593 ± 0.023	0.558 ± 0.029
Kc1	0.645 ± 0.075	0.679 ± 0.030	0.604 ± 0.036	0.607 ± 0.035	0.649 ± 0.030	0.573 ± 0.032
Madelon	0.496 ± 0.030	0.519 ± 0.028	0.533 ± 0.025	0.523 ± 0.015	0.535 ± 0.018	0.517 ± 0.019
Musk	0.712 ± 0.036	0.767 ± 0.012	0.841 ± 0.018	0.847 ± 0.018	0.857 ± 0.021	0.847 ± 0.046
Segment	0.848 ± 0.074	0.803 ± 0.014	0.898 ± 0.038	0.927 ± 0.031	0.911 ± 0.038	0.936 ± 0.024
Semeion	0.569 ± 0.055	0.755 ± 0.022	0.824 ± 0.044	0.850 ± 0.067	0.603 ± 0.096	0.544 ± 0.067
Sonar	0.497 ± 0.041	0.573 ± 0.057	0.561 ± 0.091	0.524 ± 0.074	0.602 ± 0.089	0.590 ± 0.101
Spambase	0.821 ± 0.031	0.887 ± 0.014	0.786 ± 0.011	0.775 ± 0.010	0.799 ± 0.017	0.776 ± 0.018
Vehicle	0.549 ± 0.067	0.804 ± 0.042	0.806 ± 0.037	0.823 ± 0.032	0.622 ± 0.090	0.556 ± 0.077
Waveform	0.860 ± 0.012	0.829 ± 0.012	0.795 ± 0.015	0.743 ± 0.013	0.772 ± 0.0129	0.729 ± 0.016
Wdbc	0.798 ± 0.155	0.801 ± 0.043	0.861 ± 0.068	0.845 ± 0.063	0.884 ± 0.050	0.872 ± 0.039
Yeast	0.517 ± 0.051	0.657 ± 0.024	0.630 ± 0.040	0.612 ± 0.049	0.627 ± 0.029	0.533 ± 0.046

Euler-type numerical integrator with 100 integration steps. However, one-step integration can also be used (Liu et al., 2023; 2024b). Figure 3 shows that, perceptually, one-step integration performs worse than both static and ODE-based generation, but classification accuracies are better.

The classifier-based evaluation should be interpreted together with the qualitative results. As discussed above, the classifier is trained on the target data, where the digit classes are assigned separate labels (0-9), while all alphabet classes are grouped into a single additional label. Therefore, this evaluation primarily measures whether translated MNIST samples are classified as digit-like EMNIST samples rather than alphabet-like samples. For small c, one-step translation can achieve higher accuracy because it only slightly perturbs the source image and often stays within the same classifier decision region, even if the resulting image is visually less aligned with EMNIST. In contrast, the ODE solution follows the learned transport more faithfully, which can cause more label changes for small c and lower classifier accuracy. As c increases, the relevant target subset becomes easier to identify, and the gap between one-step and ODE results becomes small both quantitatively and qualitatively.



(a) static (Gazdieva et al., 2023) (b) dynamic (one step) (c) dynamic (ODE integration)

Figure 3: Image translation outputs for MNIST → EMNIST.

4.3 Positive-Unlabeled Learning

Positive-unlabeled (PU) learning is a binary classification problem in which only a subset of positive samples is labeled. This labeled subset is then used to train a model that distinguishes positive samples from negative samples within an unlabeled dataset containing both positive and negative examples (Bekker & Davis, 2020; Kato et al., 2019; Chapel et al., 2020; Riaz et al., 2023). Since the sign of the optimal potential function φ_t in our framework differs between the selected and unselected subsets, it can be used to distinguish positive samples from unlabeled samples; see Appendix B.

We applied both static and dynamic optimal transport methods to PU learning on the 20 UCI datasets (Kelly et al.), following the setting of Teisseyre et al. (2025). We used the same 75/25 train-test split for each dataset and the sampled-completely-at-random (SCAR) mechanism to select and label positive points. Dynamic subset alignment employs Algorithm 1, while static subset alignment employs Algorithm 2, described in Appendix A.1.

Neural networks consisting of five MLP layers with appropriate input and output dimensions and Swish activation functions are used to train both the static and dynamic models. The learning rates are held constant during training for both models. Architecture and parameter details for each model are

provided in Appendix E.4. We trained 20 different models for each dataset using different train-test splits. In total, we trained 400 models for static subset alignment and 400 models for dynamic subset alignment. We adopted alternative sign-based and value-based label-assignment strategies for the unlabeled dataset. The performance of our approaches, measured in terms of balanced accuracy, along with the top-performing baselines PUSB (Kato et al., 2019) and NTC-MI (Teisseyre et al., 2025), is reported in Table 2. In all PU learning experiments, c is fixed as $c = \frac{1}{\pi}$, where π denotes the known prior proportion of positively labeled points in the dataset. Details are discussed in Appendix E.4.

Since OT methods rely on the assumption that data points in a dataset can be geometrically related through a metric, we used t-SNE (van der Maaten & Hinton, 2008) embeddings with perplexity 30 for all datasets in Table 2 to evaluate the spatial separability of the positive and negative classes. We observed that our methods perform better or comparably to competing methods when the positive class (1) and negative class (0) have spatially separable t-SNE embeddings as shown by the t-SNE embeddings are provided in Appendix G.

4.4 FFHQ Image Translation

We also apply our proposed approaches to the unpaired image translation problem. We followed the experimental setup of Gazdieva et al. (2024), where the FFHQ dataset, embedded in the latent space of the Adversarial Latent Autoencoder (ALAE) (Pidhorskyi et al., 2020), is divided either by gender (man or woman) or by age, as two orthogonal labels. Table 3, adapted from Gazdieva et al. (2024), shows the number of images for each class; images with ages less than 16 are ignored, ages between 16 and 43 are labeled young, and the remainder are labeled old. Given these classes, the task is to learn to map a source distribution to a target distribution. There are four cases: young to old, old to young, man to woman, and woman to man. To evaluate the translation process, two classifiers pretrained in the ALAE latent space are used; one classifier is trained to classify young versus old, and the other to distinguish man versus woman. The target accuracy quantifies the proportion of translated images that lie within the target-class boundary. The source accuracy quantifies whether the translated images retain the orthogonal label. For example, with young→old, the source accuracy is whether the "aged" image of a young source image retains the same gender.

Table 3: Division of FFHQ train images.

Class	Man	Woman
Young	15K	23K
Old	7K	3.5K

Implementation details for both static and dynamic subset selection to the FFHQ dataset are given in Appendix E.5. Between young→old, old→young, man→woman and woman→man, it was observed that larger values of c tend to preserve the source accuracy but often have lower target accuracy. This can be related to the fact that for larger values of c, it takes more training steps to achieve the optimal subset selection. Example translated images for old→young translation with dynamic subset

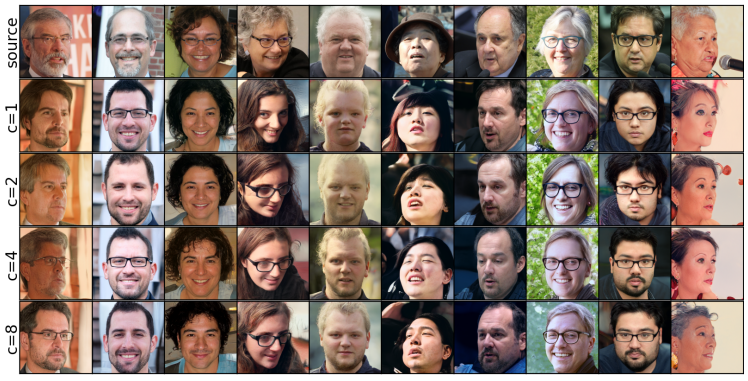


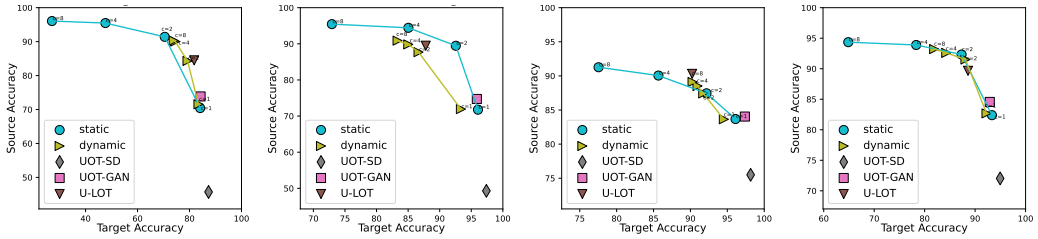
Figure 4: FFHQ old→young using dynamic subset alignment with Euler integration (100 steps).

alignment are shown in Figure 4, with other cases provided in Appendix H.

We compare with LOT Gazdieva et al. (2024), UOT-GAN Yang & Uhler (2019), UOTM, and UOT-SD Choi et al. (2023; 2024a). Results in Table 4 and Figure 5, using accuracy values for UOT-SD, UOT-GAN, and U-LOT reported by Gazdieva et al. (2024), whereas the implementation for UOTM is discussed in Appendix A.1. Methods with better target accuracy are worse for source/class accuracy.

Table 4: Target and source accuracy (%) for different domain translations on the FFHQ dataset. Dynamic subset selection is evaluated using Euler integration with 100 steps.

		c=1		c=2		c=4		c=8		UOT	UOT	U	UOTM	UOTM	UOTM
Task	Accuracy	static	dynamic	static	dynamic	static	dynamic	static	dynamic	-SD	-GAN	-LOT	KL-KL	χ^2-KL	lin.-χ^2
Young→Old	Target	84.09	83.45	70.47	79.23	47.63	74.51	27.07	73.93	87.33	84.25	81.78	71.06	72.68	80.78
	Class	70.43	71.55	91.41	84.31	95.47	90.03	96.06	90.30	45.71	73.85	84.49	90.17	88.68	78.77
Old→Young	Target	96.06	93.36	92.55	86.65	85.04	85.03	72.93	83.33	97.39	95.88	87.79	88.67	89.70	94.19
	Class	71.77	71.92	89.46	87.69	94.43	89.84	95.45	90.88	49.30	74.74	89.48	89.64	88.19	80.68
Man→Woman	Target	96.11	94.53	92.18	91.74	85.66	90.96	77.55	90.29	98.16	97.38	90.23	90.63	91.19	94.51
	Class	83.68	83.64	87.45	87.43	90.05	88.52	91.27	89.11	75.50	84.04	90.30	89.52	88.83	85.31
Woman→Man	Target	93.34	92.26	87.32	88.09	78.32	84.28	64.86	81.89	94.96	92.91	88.59	83.85	84.10	91.41
	Class	82.39	82.68	92.33	91.51	93.89	92.59	94.35	93.22	72.03	84.56	89.66	91.79	90.85	87.17



(a) Young→Old (b) Old→Young (c) Man→Woman (d) Woman→Man

Figure 5: Accuracy curves for $c \in \{1, 2, 4, 8\}$, in comparison to results from LOT (Gazdieva et al., 2024), UOT-GAN (Yang & Uhler, 2019), UOT-SD and UOTM (Choi et al., 2023; 2024a).

4.5 Melanoma Dataset

The subset-alignment framework is primarily developed for aligning a source distribution μ to an unknown distribution $\tilde{\nu}$ whose support is contained within the support of a target distribution ν. Following Neklyudov et al. (2024) and Pariset et al. (2023), we adapt our dynamic subset-alignment method to the melanoma treatment data of Bunne et al. (2023), using the preprocessed 50-dimensional representation. Unlike our other experiments, the melanoma setting involves three observed marginals at $t \in \{8\text{hr}, 24\text{hr}, 48\text{hr}\}$ and we must match the intermediate marginal exactly. We therefore order the populations in reverse chronological order, μ_0 (48h), μ_1 (24h), and μ_2 (8h), and train a multi-marginal interpolant that preserves μ_0 and μ_1 while leaving the terminal marginal (here, the 8h population) unconstrained so that sub-setting acts on μ_2. After training, we select cells $\mathbf{x} \sim \mu_2$ with $\varphi_{t_2}(\mathbf{x}) \leq 0$ and transport the selected subset back to t_0 using the learned velocity field $\nabla\varphi_t$. (We use normalized times $t_0 = 0$, $t_1 = 0.6$, $t_2 = 1$.) For evaluation, we integrate the ODE with explicit Euler (step size 10^{-3}) and report the best checkpoint based on validation MMD. The training procedure, neural parameterizations and experimental parameters for the Melanoma experiment are given in the Appendix E.6. As shown in Table 5, subset alignment improves over previous neural Schrödinger-bridge baselines (Chen et al., 2022; Pariset et al., 2023), while remaining competitive with WLF (Neklyudov et al., 2024).

Table 5: Results for the Melanoma dataset (Pariset et al., 2023, Table 1) with three marginals and train/test splits. MMD and W_2 are measured between $\nu_{t=48}$ and $\rho_{t=48}$ obtained by running dynamics from $\mu_{t=8}$. For subset alignment, dynamics are run from the selected subset of $\mu_{t=8}$ (cells with $\varphi_{t=8} \leq 0$; number of selected test cells in parentheses), and results are reported at the iteration with the lowest MMD on a validation set; note that the validation cells are a subset of the training cells.

Model	MMD	W_2	Model	MMD	W_2
SB-FBSDE (Chen et al., 2022)	1.86e−2	6.23	Subset Alignment ($c = 1$, 476/490)	5.21e−3	4.83
uDSB (no growth) (Pariset et al., 2023)	1.86e−2	6.27	Subset Alignment ($c = 2.0$, 247/490)	6.10e−3	**4.68**
uDSB (w/growth) (Pariset et al., 2023)	1.75e−2	6.11	Subset Alignment ($c = 4.0$, 148/490)	9.24e−3	5.40
WLF-OT no growth (Neklyudov et al., 2024)	**5.04e−3**	5.20	Subset Alignment ($c = 8.0$, 103/490)	1.48e−2	6.60
WLF-UOT $\lambda = 0.1$ (Neklyudov et al., 2024)	9.16e−3	5.01			

4.6 Celeb-A to Anime Domain Translation

For domain translation between the CelebA (Liu et al., 2015) and Anime (Chao, 2019) datasets, we use highly trimmed variants of the DDPM++ architecture (Song et al., 2021b) to parameterize both Q_t and φ_t so that the models fit within GPU memory. Because our subset-alignment framework

maps the source distribution to a *subset* of the target distribution, FID is not an appropriate evaluation metric for this setting. We therefore rely on qualitative visual inspection. In our experiments, dynamic subset alignment successfully translates the source distribution toward the target domain; however, larger values of c generally require longer training to achieve comparable visual quality and stability. A sample translation is shown in Figure 6.

5 LIMITATIONS

Practically, one point of concern is that the sample estimators of Wasserstein distances suffer from the curse of dimensionality (Weed & Bach, 2019; Fournier & Guillin, 2015). This issue can be alleviated to some extent by employing entropic regularization (Genevay et al., 2019; Feydy et al., 2019), which, in the dynamic setting, is intimately connected to Schrödinger bridges (Peluchetti, 2023; Bortoli et al., 2024).



Figure 6: CelebA→Anime.

Recently, entropically regularized UOT has been studied for modeling birth or death processes in population dynamics (Pariset et al., 2023; Neklyudov et al., 2023a). Our approach can also be applied to model death processes in settings where there is a canonical relationship between temporally ordered populations. In such cases, we treat μ as the final population of survivors and ν as the initial population. The learned mapping then relates the survivors to a subset of the initial population. Moreover, the sign of the potential can be used to distinguish members of the initial population that are likely to survive from those that are not. We note that, in this and other applications, the choice of c is critical. While c has a clear interpretation as a constraint on the density ratio, automatic selection of c based on the resulting transport cost may be possible, as previously studied for partial optimal transport in the discrete setting (Phatak et al., 2023). One consolidated approach would be to sample c from a prescribed range and use multi-task learning to optimize networks across varying values of c. From an implementation perspective, this can be achieved by using scalar embeddings of c, analogous to the time-variable embeddings used in dynamic networks.

Another key limitation of dynamic models, including frameworks similar to ours, is that their iterative generation process requires more computational resources and longer inference time (Liu et al., 2023). For example, Table 6 reports the average per-sample generation time for FFHQ image translation. These results show that, in this setting, one-step generation is at least an order of magnitude faster than ODE integration. Considerable recent effort has been devoted to improving the generation efficiency of dynamic models (Liu et al., 2024b; Geng et al., 2026).

Table 6: Average evaluation time per sample for FFHQ domain translation tasks from Table 4. Evaluation is performed with batch size 128.

Metric	One-Step	ODE
Time/sample (s)	1.44×10^{-4}	2.84×10^{-3}

We note that although this work focuses on relaxing the target distribution, the range-divergence framework could also be used to relax the source distribution. A fully relaxed version, in which both source and target marginals are relaxed, may be applicable to other classes of problems.

6 CONCLUSION

Our neural optimal transport approach with subset selection achieves subset alignment by translating between two distributions while reweighting and selecting a subset of the target support. Our results on image-translation tasks across two datasets and 20 tabular PU-learning tasks show that a meaningful subset can be learned simultaneously with a Monge map. Unlike previous work, our dynamic formulation allows the terminal distribution to differ from the original target marginal, thereby inducing flows toward the nearest target subset while maintaining the prescribed density-ratio constraint.

AI USE STATEMENT

We used generative AI tools primarily for copy-editing (improving grammar, clarity, and flow). We did not use generative AI for code development, except in the Melanoma dataset section, where an AI tool helped identify a bug. All AI-assisted outputs were reviewed and verified by the authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

To facilitate full reproducibility, we provide detailed descriptions of all experiments in the appendix. The main body summarizes the experimental setup and procedures for each experiment, while the appendix includes the complete specifications needed to replicate our results.

REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. Building Normalizing Flows with Stochastic Interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. [paper].

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025. [paper].

Fabian Altekrüger, Johannes Hertrich, and Gabriele Steidl. Neural Wasserstein Gradient Flows for Discrepancies with Riesz Kernels. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 664–690. PMLR, 23–29 Jul 2023. [paper].

David Alvarez-Melis, Yair Schiff, and Youssef Mroueh. Optimizing Functionals on the Space of Probabilities with Input Convex Neural Networks. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. [paper].

Brandon Amos. On amortizing convex conjugates for optimal transport. In *The Eleventh International Conference on Learning Representations*, 2023. [paper].

Brandon Amos, Lei Xu, and J Zico Kolter. Input Convex Neural Networks. In *International Conference on Machine Learning*, pp. 146–155. PMLR, PMLR, 2017. [paper].

Brandon Amos, Giulia Luise, Samuel Cohen, and Ievgen Redko. Meta Optimal Transport. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 791–813. PMLR, 23–29 Jul 2023. [paper].

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*, pp. 214–223. PMLR, 2017. [paper].

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. [paper].

Lazar Atanackovic, Xi Zhang, Brandon Amos, Mathieu Blanchette, Leo J Lee, Yoshua Bengio, Alexander Tong, and Kirill Neklyudov. Meta Flow Matching: Integrating Vector Fields on the Wasserstein Manifold. In *The Thirteenth International Conference on Learning Representations*, 2025. [paper].

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. [paper].

Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: a survey. *Machine Learning*, 109:719–760, 2020. [paper].

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000. [paper].

Valentin De Bortoli, Iryna Korshunova, Andriy Mnih, and Arnaud Doucet. Schrödinger Bridge Flow for Unpaired Data Translation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [paper].

Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on Pure and Applied Mathematics*, 44(4):375–417, 1991. [paper].

Charlotte Bunne, Andreas Krause, and marco cuturi. Supervised Training of Conditional Monge Maps. In *Advances in Neural Information Processing Systems*, 2022a. [paper].

Charlotte Bunne, Laetitia Papaxanthos, Andreas Krause, and Marco Cuturi. Proximal Optimal Transport Modeling of Population Dynamics. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 6511–6528. PMLR, 28–30 Mar 2022b. [paper].

Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20(11):1759–1768, 2023. [paper].

Brian Chao. Anime face dataset: a collection of high-quality anime faces., 2019. [dataset].

Laetitia Chapel, Mokhtar Z Alaya, and Gilles Gasso. Partial Optimal Tranport with applications on Positive-Unlabeled Learning. *Advances in Neural Information Processing Systems*, 33:2903–2913, 2020. [paper].

Ricky TQ Chen and Yaron Lipman. Flow Matching on General Geometries. In *The Twelfth International Conference on Learning Representations*, 2024. [paper].

Tianrong Chen, Guan-Horng Liu, and Evangelos Theodorou. Likelihood training of schrödinger bridge using forward-backward SDEs theory. In *International Conference on Learning Representations*, 2022. [paper].

Xiuyuan Cheng, Jianfeng Lu, Yixin Tan, and Yao Xie. Convergence of Flow-Based Generative Models via Proximal Gradient Descent in Wasserstein Space. *IEEE Transactions on Information Theory*, 70(11):8087–8106, 2024. [paper].

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. An Interpolating Distance Between Optimal Transport and Fisher–Rao Metrics. *Foundations of Computational Mathematics*, 18:1–44, 2018a. [paper].

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and Kantorovich formulations. *Journal of Functional Analysis*, 274(11): 3090–3123, 2018b. [paper].

Jaemoo Choi, Jaewoong Choi, and Myungjoo Kang. Generative Modeling through the Semi-dual Formulation of Unbalanced Optimal Transport. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [paper].

Jaemoo Choi, Jaewoong Choi, and Myungjoo Kang. Analyzing and Improving Optimal-Transport-based Adversarial Networks. In *The Twelfth International Conference on Learning Representations*, 2024a. [paper].

Jaemoo Choi, Jaewoong Choi, and Myungjoo Kang. Scalable Wasserstein Gradient Flow for Generative Modeling through Unbalanced Optimal Transport. In *Forty-first International Conference on Machine Learning*, 2024b. [paper].

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926, 2017. [paper].

Li Deng. The MNIST Database of Handwritten Digit Images for Machine Learning Research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. [paper].

Luca Eyring, Dominik Klein, Théo Uscidda, Giovanni Palla, Niki Kilbertus, Zeynep Akata, and Fabian J Theis. Unbalancedness in Neural Monge Maps Improves Unpaired Domain Translation. In *The Twelfth International Conference on Learning Representations*, 2024. [paper].

Jiaojiao Fan, Shu Liu, Shaojun Ma, Yongxin Chen, and Hao-Min Zhou. Scalable Computation of Monge Maps with General Costs. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022a. [paper].

Jiaojiao Fan, Qinsheng Zhang, Amirhossein Taghvaei, and Yongxin Chen. Variational Wasserstein gradient flow. In *proceedings of International Conference on Machine Learning*, 2022b. [paper].

Jiaojiao Fan, Shu Liu, Shaojun Ma, Hao-Min Zhou, and Yongxin Chen. Neural Monge Map estimation and its applications. *Transactions on Machine Learning Research*, 2023. [paper].

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between Optimal Transport and MMD using Sinkhorn Divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2681–2690. PMLR, 2019. [paper].

Alessio Figalli and Federico Glaudo. *An invitation to optimal transport, Wasserstein distances, and gradient flows: Second Edition*. European Mathematical Society, 2023. [book].

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3):707–738, 2015. [paper].

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso Poggio. Learning with a wasserstein loss. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. [paper].

Wilfrid Gangbo and Robert J McCann. The geometry of optimal transportation. *Acta Mathematica*, 177:113–161, 1996. [paper].

Milena Gazdieva, Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Extremal Domain Translation with Neural Optimal Transport. *Advances in Neural Information Processing Systems*, 36:40381–40413, 2023. [paper].

Milena Gazdieva, Arip Asadulaev, Evgeny Burnaev, and Alexander Korotin. Light Unbalanced Optimal Transport. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [paper].

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Advances in Neural Information Processing Systems*, volume 29, 2016. [paper].

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning Generative Models with Sinkhorn Divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617. PMLR, 2018. [paper].

Aude Genevay, Lénaic Chizat, Francis Bach, Marco Cuturi, and Gabriel Peyré. Sample Complexity of Sinkhorn Divergences. In *The 22nd iInternational Conference on Artificial Intelligence and Statistics*, pp. 1574–1583. PMLR, 2019. [paper].

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. [paper].

Jonathan Geuter, Gregor Kornhardt, Ingimar Tomasson, and Vaios Laschos. Universal Neural Optimal Transport. In *Forty-second International Conference on Machine Learning*, 2025. [paper].

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. [paper].

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved Training of Wasserstein GANs. *Advances in Neural Information Processing Systems*, 30, 2017. [paper].

Nikita Gushchin, Alexander Kolesov, Alexander Korotin, Dmitry P. Vetrov, and Evgeny Burnaev. Entropic Neural Optimal Transport via Diffusion Processes. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. [paper].

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. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023b. [paper].

Doron Haviv, Aram-Alexandre Pooladian, Dana Pe'er, and Brandon Amos. Wasserstein Flow Matching: Generative Modeling Over Families of Distributions. In *Forty-second International Conference on Machine Learning*, 2025. [paper].

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. [paper].

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2): 251–257, 1991. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(91)90009-T. [paper].

Bamdad Hosseini, Alexander W. Hsu, and Amirhossein Taghvaei. Conditional Optimal Transport on Function Spaces. *SIAM/ASA Journal on Uncertainty Quantification*, 13(1):304–338, 2025. [paper].

Samuel Howard, George Deligiannidis, Patrick Rebeschini, and James Thornton. Differentiable Cost-Parameterized Monge Map Estimators. In *ICML 2024 Workshop on Differentiable Almost Everything: Differentiable Relaxations, Algorithms, Operators, and Simulators*, 2024. [paper].

Chin-Wei Huang, Ricky T. Q. Chen, Christos Tsirigotis, and Aaron Courville. Convex Potential Flows: Universal Probability Distributions with Optimal Transport and Convex Optimization. In *International Conference on Learning Representations*, 2021. [paper].

Guillaume Huguet, Daniel Sumner Magruder, Alexander Tong, Oluwadamilola Fasina, Manik Kuchroo, Guy Wolf, and Smita Krishnaswamy. Manifold Interpolating Optimal-Transport Flows for Trajectory Inference. In *Advances in Neural Information Processing Systems*, 2022. [paper].

Richard Jordan, David Kinderlehrer, and Felix Otto. The Variational Formulation of the Fokker–Planck Equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998. [paper].

Leonid V Kantorovich. On the translocation of masses. *Doklady Akademii Nauk SSSR (New Series)*, 37:199–201, 1942. [paper: english translation].

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. [paper].

Masahiro Kato, Takeshi Teshima, and Junya Honda. Learning from Positive and Unlabeled Data with a Selection Bias. In *International Conference on Learning Representations*, 2019. [paper].

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The UCI Machine Learning Repository. [repository].

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Dynamic Conditional Optimal Transport through Simulation-Free Flows. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [paper].

Valentin Khrulkov, Gleb Ryzhakov, Andrei Chertkov, and Ivan Oseledets. Understanding DDPM Latent Codes Through Optimal Transport. In *The Eleventh International Conference on Learning Representations*, 2023. [paper].

Boah Kim, Yan Zhuang, Tejas Sudharshan Mathai, and Ronald M Summers. OTMorph: Unsupervised Multi-domain Abdominal Medical Image Registration Using Neural Optimal Transport. *IEEE Transactions on Medical Imaging*, 2024. [paper].

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, 2017. [paper].

Nikita Maksimovich Kornilov, Petr Mokrov, Alexander Gasnikov, and Alexander Korotin. Optimal Flow Matching: Learning Straight Trajectories in Just One Step. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [paper].

Alexander Korotin, Vage Egiazarian, Arip Asadulaev, Alexander Safin, and Evgeny Burnaev. Wasserstein-2 Generative Networks. In *International Conference on Learning Representations*, 2021a. [paper].

Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, Alexander Filippov, and Evgeny Burnaev. Do Neural Optimal Transport Solvers Work? a Continuous Wasserstein-2 benchmark. *Advances in Neural Information Processing Systems*, 34:14593–14605, 2021b. [paper].

Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Kernel Neural Optimal Transport. In *The Eleventh International Conference on Learning Representations*, 2023a. [paper].

Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural Optimal Transport. In *The Eleventh International Conference on Learning Representations*, 2023b. [paper].

Dohyun Kwon, Ying Fan, and Kangwook Lee. Score-based generative modeling secretly minimizes the wasserstein distance. In *Advances in Neural Information Processing Systems*, 2022. [paper].

Marc Lambert, Sinho Chewi, Francis Bach, Silvère Bonnabel, and Philippe Rigollet. Variational inference via wasserstein gradient flows. In *Advances in Neural Information Processing Systems*, 2022. [paper].

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*, 2023. [paper].

Guan-Horng Liu, Tianrong Chen, Oswin So, and Evangelos Theodorou. Deep Generalized Schrödinger Bridge. In *Advances in Neural Information Processing Systems*, 2022. [paper].

Guan-Horng Liu, Yaron Lipman, Maximilian Nickel, Brian Karrer, Evangelos Theodorou, and Ricky T. Q. Chen. Generalized Schrödinger Bridge Matching. In *The Twelfth International Conference on Learning Representations*, 2024a. [paper].

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *The Eleventh International Conference on Learning Representations*, 2023. [paper].

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and qiang liu. InstaFlow: One Step is Enough for High-Quality Diffusion-Based Text-to-Image Generation. In *The Twelfth International Conference on Learning Representations*, 2024b. [paper].

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. [dataset].

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations*, 2017. [paper].

Frederike Lübeck, Charlotte Bunne, Gabriele Gut, Jacobo Sarabia del Castillo, Lucas Pelkmans, and David Alvarez-Melis. Neural Unbalanced Optimal Transport via Cycle-Consistent Semi-Couplings. In *NeurIPS 2022 AI for Science: Progress and Promises*, 2022. [paper].

Shaojun Ma, Shu Liu, Hongyuan Zha, and Haomin Zhou. Learning Stochastic Behaviour from Aggregate Data. In *International Conference on Machine Learning*, pp. 7258–7267. PMLR, 2021. [paper].

Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, pp. 6672–6681. PMLR, 2020. [paper].

Robert J McCann. A Convexity Principle for Interacting Gases. *Advances in Mathematics*, 128(1): 153–179, 1997. [paper].

Petr Mokrov, Alexander Korotin, Lingxiao Li, Aude Genevay, Justin Solomon, and Evgeny Burnaev. Large-Scale Wasserstein Gradient Flows. In *Advances in Neural Information Processing Systems*, 2021. [paper].

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. [paper].

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Mémoires de Mathématique et de Physique, présentés à l'Académie Royale des Sciences*, pp. 666–704, 1781. [paper].

Kirill Neklyudov, Rob Brekelmans, Daniel Severo, and Alireza Makhzani. Action Matching: Learning Stochastic Dynamics from Samples. In *International Conference on Machine Learning*, pp. 25858–25889. PMLR, 2023a. [paper].

Kirill Neklyudov, Jannes Nys, Luca Thiede, Juan Carrasquilla, Qiang Liu, Max Welling, and Alireza Makhzani. Wasserstein Quantum Monte Carlo: A Novel Approach for Solving the Quantum Many-Body Schrödinger Equation. *Advances in Neural Information Processing Systems*, 36, 2023b. [paper].

Kirill Neklyudov, Rob Brekelmans, Alexander Tong, Lazar Atanackovic, Qiang Liu, and Alireza Makhzani. A Computational Framework for Solving Wasserstein Lagrangian Flows. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37461–37485. PMLR, 21–27 Jul 2024. [paper].

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010. [paper].

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. *Advances in Neural Information Processing Systems*, 29, 2016. [paper].

Matteo Pariset, Ya-Ping Hsieh, Charlotte Bunne, Andreas Krause, and Valentin De Bortoli. Unbalanced diffusion schrödinger bridge. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023. [paper].

Stefano Peluchetti. Diffusion Bridge Mixture Transports, Schrödinger Bridge Problems and Generative Modeling. *Journal of Machine Learning Research*, 24(374):1–51, 2023. [paper].

Abhijeet Phatak, Sharath Raghvendra, Chittaranjan Tripathy, and Kaiyi Zhang. Computing all Optimal Partial Transports. In *The Eleventh International Conference on Learning Representations*, 2023. [paper].

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. [paper].

Yury Polyanskiy and Yihong Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2024. [book].

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T Chen. Multisample Flow Matching: Straightening Flows with Minibatch Couplings. In *International Conference on Machine Learning*, 2023. [paper].

Aram-Alexandre Pooladian, Carles Domingo-Enrich, Ricky T. Q. Chen, and Brandon Amos. Neural Optimal Transport with Lagrangian Costs. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024. [paper].

Bilal Riaz, Yuksel Karahan, and Austin J. Brockmeier. Partial Optimal Transport for Support Subset Selection. *Transactions on Machine Learning Research*, 2023. [paper].

R Tyrrell Rockafellar. Integral functionals, normal integrands and measurable selections. *Lecture Notes in Mathematics*, pp. 157–207, 1976. [paper].

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Iimage 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. [paper].

Litu Rout, Alexander Korotin, and Evgeny Burnaev. Generative Modeling with Optimal Transport Maps. In *International Conference on Learning Representations*, 2022. [paper].

Filippo Santambrogio. Optimal Transport For Applied Mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015. [book].

Filippo Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7:87–154, 2017. [paper].

Christopher Scarvelis and Justin Solomon. Riemannian Metric Learning via Optimal Transport. In *The Eleventh International Conference on Learning Representations*, 2023. [paper].

Vivien Seguy, Bharath Bhushan Damodaran, Remi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large Scale Optimal Transport and Mapping Estimation. In *International Conference on Learning Representations*, 2018. [paper].

Thibault Séjourné, Gabriel Peyréa, and François-Xavier Vialard. Unbalanced Optimal Transport, from theory to numerics. *Numerical Control: Part B*, pp. 407, 2023. paper.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion Schrödinger Bridge Matching. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [paper].

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. [paper].

Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. Aligned Diffusion Schrödinger Bridges. In *The 39th Conference on Uncertainty in Artificial Intelligence*, 2023. [paper].

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2021a. [paper].

Yang Song and Stefano Ermon. Improved Techniques for Training Score-Based Generative Models. *Advances in Neural Information Processing Systems*, 33:12438–12448, 2020. [paper].

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021b. [paper].

Paweł Teisseyre, Timo Martens, Jessa Bekker, and Jesse Davis. Learning from biased positive-unlabeled data via threshold calibration. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025. [paper].

Alexander Tong, Jessie Huang, Guy Wolf, David Van Dijk, and Smita Krishnaswamy. TrajectoryNet: A Dynamic Optimal Transport Network for Modeling Cellular Dynamics. In *International conference on machine learning*, pp. 9526–9536. PMLR, 2020. [paper].

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856. [paper].

Alexander Y. Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-Free Schrödinger Bridges via Score and Flow Matching. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 1279–1287. PMLR, 02–04 May 2024b. [paper].

Théo Uscidda and Marco Cuturi. The Monge Gap: A Regularizer to Learn All Transport Maps. In *International Conference on Machine Learning*, pp. 34709–34733. PMLR, 2023. [paper].

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. [paper].

Nina Vesseron and Marco Cuturi. On a Neural Implementation of Brenier's Polar Factorization. In *Forty-first International Conference on Machine Learning*, 2024. [paper].

Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021. ISBN 978-1-4704-6726-5.

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

Wei Wan, Yuejin Zhang, Chenglong Bao, Bin Dong, and Zuoqiang Shi. A Scalable Deep Learning Approach for Solving high-Dimensional Dynamic Dptimal Transport. *SIAM Journal on Scientific Computing*, 45(4):B544–B563, 2023. [paper].

Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep Generative Learning via Schrödinger Bridge. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10794–10804. PMLR, 18–24 Jul 2021. [paper].

Jonathan Weed and Francis Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *Bernoulli*, 25(4A):pp. 2620–2648, 2019. ISSN 13507265, 15739759. [paper].

Chen Xu, Xiuyuan Cheng, and Yao Xie. Normalizing flow neural networks by JKO scheme. In *Advances in Neural Information Processing Systems*, volume 36, pp. 47379–47405, 2023. [paper].

Karren D. Yang and Caroline Uhler. Scalable Unbalanced Optimal Transport using Generative Adversarial Networks. In *International Conference on Learning Representations*, 2019. [paper].

A Preliminaries and Problem Formulation

Gaspard Monge proposed the original idea of optimal transport as mathematical model for the problem of minimum-cost transportation of dirt from a source location to a destination (Monge, 1781). In more modern parlance, given probability measures, μ defined on compact set $\mathcal{X} \subseteq \mathbb{R}^d$, ν defined on compact set $\mathcal{Y} \subseteq \mathbb{R}^d$, and the bounded uniformly continuous cost $\mathsf{c}(\cdot,\cdot) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the Monge formulation of optimal transport is stated as

$$\mathcal{D}_{\text{Monge}}(\mu,\nu) = \inf_{T \in \mathcal{J}(\mathcal{X},\mathcal{Y})} \int_{\mathcal{X}} \mathsf{c}\left(\mathbf{x}, T(\mathbf{x})\right) \mu(\mathbf{x}) d\mathbf{x} \qquad \text{s.t.} \quad T_{\#}\mu = \nu \tag{10}$$

where the set $\mathcal{J}\left(\mathcal{X},\mathcal{Y}\right)$ denotes the set of measurable maps between $\mathcal{X}$ and $\mathcal{Y}$. Monge formulation of the optimal transport problem requires that the transport map of T to be a deterministic function. To satisfy the constraint in the Monge problem 10, the transport map T must cover ν up to some ν-null sets. Usually, the cost c is non-linearly dependent on the transportation map T, making the problem 10 very cumbersome and very difficult to solve (Santambrogio, 2015; Villani et al., 2009). Kantorovich reformulated the Monge problem by relaxing the constraint that supports of μ and ν should be related to each other by a functional relation T (Kantorovich, 1942). Instead, he allowed

the marginals μ and ν to be related to each other by a joint measure. Kantorovich's reformulation of the problem is a linear program, and its solution exists for all convex lower-semi-continuous costs (Santambrogio, 2015; Figalli & Glaudo, 2023). The Kantorovich problem is

$$\mathcal{W}(\mu,\nu) = \inf_{\pi} \int_{\mathcal{X}\times\mathcal{Y}} \mathsf{c}\left(\mathbf{x},\mathbf{y}\right)\pi(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y} \quad \text{s.t.} \int_{\mathcal{Y}} \pi(\mathbf{x},\mathbf{y})d\mathbf{y} = \mu(\mathbf{x}), \quad \int_{\mathcal{X}} \pi(\mathbf{x},\mathbf{y})d\mathbf{x} = \nu(\mathbf{y}), \tag{11}$$

where it can be observed that integrals $\int_{\mathcal{X}}$ and $\int_{\mathcal{Y}}$ marginalize with respect to spaces $\mathcal{Y}$ and $\mathcal{X}$ respectively. Therefore, π must be the joint measure between μ and ν defined on the product space $\mathcal{X}\times\mathcal{Y}$, i.e. $\pi \in \mathcal{P}\left(\mathcal{X}\times\mathcal{Y}\right)$. In other words, constraints in the Kantorovich problem ensure that every feasible solution π must be a joint distribution of μ and ν. While in general, the Kantorovich problem is much easier to solve in comparison to the Monge problem, there are conditions of practical importance where one can employ the solutions of the Kantorovich problem to obtain the solution of the Monge problem. Those conditions are more clearly discussed in terms of the dual form of the Kantorovich problem (Santambrogio, 2015) given as,

$$\mathcal{W}(\mu,\nu) = \sup_{f,g} \int_{\mathcal{X}} f(\mathbf{x})\mu(\mathbf{x})d\mathbf{x} + \int_{\mathcal{Y}} g(\mathbf{y})\mu(\mathbf{y})d\mathbf{y} \quad \text{s.t.} \quad f(\mathbf{x}) + g(\mathbf{y}) \leq \mathsf{c}(\mathbf{x},\mathbf{y}). \tag{12}$$

The functions $f(\mathbf{x})$ and $g(\mathbf{y})$ are called Kantorovich potentials. By defining c-conjugate (also called c-transform) of $f(\mathbf{x})$, and $\bar{\mathsf{c}}$-conjugate (also called $\bar{\mathsf{c}}$-transform) of $g(\mathbf{y})$ as

$$f^{\mathsf{c}}(\mathbf{y}) = \inf_{\mathbf{x}\in\mathcal{X}} \mathsf{c}(\mathbf{x},\mathbf{y}) - f(\mathbf{x}), \tag{13}$$

$$g^{\bar{\mathsf{c}}}(\mathbf{x}) = \inf_{\mathbf{x}\in\mathcal{X}} \mathsf{c}(\mathbf{x},\mathbf{y}) - g(\mathbf{y}), \tag{14}$$

Using c and $\bar{\mathsf{c}}$ conjugates, Kantorovich problem is expressed as

$$\mathcal{W}(\mu,\nu) = \sup_{f} \int_{\mathcal{X}} f(\mathbf{x})\mu(\mathbf{x})d\mathbf{x} + \int_{\mathcal{Y}} f^{\mathsf{c}}(\mathbf{y})\nu(\mathbf{y})d\mathbf{y} \tag{15}$$

$$= \sup_{g} \int_{\mathcal{X}} g^{\bar{\mathsf{c}}}(\mathbf{x})\mu(\mathbf{x})d\mathbf{x} + \int_{\mathcal{Y}} g(\mathbf{y})\nu(\mathbf{y})d\mathbf{y} \tag{16}$$

Under very general conditions, one can relate the cost c with the support of optimal coupling solution $\pi^\star$ and optimal Kantorovich potentials $f^\star(\mathbf{x})$ and $f^{\mathsf{c}\star}(\mathbf{y})$ (Santambrogio, 2015, Theorem 1.37) by

$$\text{supp}(\pi^\star) \subset \{(\mathbf{x},\mathbf{y}) \in \mathcal{X}\times\mathcal{Y} : f^\star(x) + f^{\star\mathsf{c}}(\mathbf{y}) = \mathsf{c}(\mathbf{x},\mathbf{y})\} \tag{17}$$

In discrete domains, above result is equivalent to Karush-Kuhn-Tucker (KKT) conditions for optimality. One can further relate the optimal solutions of Monge and Kantorovich problems using a landmark result by Gangbo & McCann (1996) (Figalli & Glaudo, 2023, Theorem 2.7.1), which states that there exists an optimal Kantorovich coupling of the form $\pi^\star = (\text{Id}\times T^\star)_{\#}\mu$, where $T^\star$ is Monge map satisfying

$$\nabla_{\mathbf{x}}\mathsf{c}(\mathbf{x}, T^\star(\mathbf{x})) + \nabla f^\star(\mathbf{x}) = 0, \tag{18}$$

if the following conditions are satisfied

- μ is absolutely continuous,
- $\forall\ \mathbf{y}\in\mathcal{Y}$ map $x \mapsto \mathsf{c}(\mathbf{x},\mathbf{y})$ is differentiable $\forall\ \mathbf{x}\in\mathcal{X}$,
- $\forall\ \mathbf{x}\in\mathcal{X}$ the gradient map $y \mapsto \nabla_{\mathbf{x}}\mathsf{c}(\mathbf{x},\mathbf{y})$ is injective $\forall\ \mathbf{y}\in\mathcal{Y}$,
- and, the gradient $\nabla_{\mathbf{x}}\mathsf{c}(\mathbf{x},\mathbf{y})$ satisfies the local Lipschitz condition $\|\nabla_{\mathbf{x}}\mathsf{c}(\mathbf{x},\mathbf{y})\| \leq C_r$ for all $\mathbf{x}\in\mathcal{B}_r$ where $\mathcal{B}_r$ is a ball of radius r around $\mathbf{x}$.

When the cost can be written as $\mathsf{c}(\mathbf{x},\mathbf{y}) = h(\mathbf{x}-\mathbf{y})$, where h is strictly convex and translation-invariant function, one can further relate the Monge mapping with optimal dual potential by (Santambrogio, 2015, Theorem 1.17)

$$T^\star(\mathbf{x}) = \mathbf{x} - \nabla h^* \circ \nabla f^\star(\mathbf{x}), \tag{19}$$

where h^* is the Legendre-Fenchel conjugate of h given by $h^*(\mathbf{y}) = \sup_{x\in\mathcal{X}}\{\langle\mathbf{y},\mathbf{x}\rangle - h(\mathbf{x})\}$. In the result above, when $h(\mathbf{x}-\mathbf{y}) = \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2$, one obtains the result of the celebrated Brenier theorem of optimal transport for squared-Euclidean costs with $T^\star(\mathbf{x}) = \nabla f^\star(\mathbf{x})$, where $f^\star(\mathbf{x})$ is

convex (Brenier, 1991) (Figalli & Glaudo, 2023, Theorem 2.5.10). The Brenier theorem on optimal transport differs from another important theorem on polar factorization (Brenier, 1991) stating that under very general conditions a square-integrable vector field $\mathbf{v}$ can be decomposed into the composition of the gradient of a unique convex function ξ and a unique measure-preserving map $\mathbf{u}$, i.e. $\mathbf{v}(\mathbf{x}) = \nabla\xi \circ \mathbf{u}(\mathbf{x})$. Before the discussion on dynamic formulation of the problem, we would like to point out that much of the recent work on static neural optimal transport relies on the above results.

Benamou and Brenier formulated the Wasserstein distance with squared Euclidean cost as the expected kinetic energy minimization problem under the assumption that both the source μ and target ν distributions have finite second moments (Benamou & Brenier, 2000; Santambrogio, 2015; Figalli & Glaudo, 2023). Assuming that supports of both source and target distributions lie in a convex set $\Omega \subseteq \mathbb{R}^d$, whose normal at the boundary is given by $\mathbf{n} : \partial\Omega \to \mathbb{R}^d$. For a bounded and smooth velocity-field $\mathbf{v}_t(\mathbf{x}) : [0,1] \times \Omega \to \mathbb{R}^d$ such that $\langle \mathbf{v}_t(\mathbf{x}), \mathbf{n}\rangle|_{\partial\Omega} = 0$, the flow corresponding to the velocity-field $\mathbf{v}_t$ is given by

$$\frac{d}{dt}\Phi_t(\mathbf{x}_t) = \mathbf{v}_t(\Phi_t(\mathbf{x}_t)), \;\; \Phi_0(\mathbf{x}_0) = \mathbf{x}_0. \tag{20}$$

Considering that there also exists a probability path $\rho_t(\mathbf{x}) : [0,1] \times \Omega \to \mathbb{R}_+$ corresponding to the flow $\Phi_t(\mathbf{x})$ such that $\rho_t(\mathbf{x}) = \Phi_{t\#}\rho_0(\mathbf{x})$, Benamou-Brenier formulation of optimal transport is

$$\begin{aligned} &\inf_{\rho_t, \mathbf{v}_t} \int_0^1 \int_\Omega \frac{\|\mathbf{v}_t(\mathbf{x})\|^2}{2} \rho_t(\mathbf{x}) d\mathbf{x}\, dt \\ &\text{s.t.} \quad \frac{\partial}{\partial_t}\rho_t(\mathbf{x}) + \operatorname{div}(\rho_t(\mathbf{x})\mathbf{v}_t(\mathbf{x})) = 0, \quad \rho_0(\mathbf{x}) = \mu(\mathbf{x}), \quad \rho_1(\mathbf{x}) = \nu(\mathbf{x}), \end{aligned} \tag{21}$$

where, $\operatorname{div}(\cdot)$ denotes the divergence operator mapping scalar or vector fields to scalar for the field $z_t(\mathbf{x})$ by $\operatorname{div}(z_t(\mathbf{x})) = \sum_i \frac{\partial}{\partial x_i} z_t(\mathbf{x})$. The optimal flow $\Phi_t^\star$ is related to Monge mapping $T^\star$ by displacement interpolation (McCann, 1997),

$$\Phi_t^\star = (1-t)\mathrm{Id} + tT^\star. \tag{22}$$

It is important to mention that the Benamou-Brenier formulation can be extended to Wasserstein-p distances, for $p > 1$, under the assumption that both source and target distributions have finite pth moments (Santambrogio, 2015, Chapters 5 & 6).

A.1 Static support subset alignment

Our formulation of static support subset alignment with optimal transport is derived from relaxed optimal transport problem where the constraint on the first marginal of the joint density π is maintained, while the second marginal $\int_{\mathcal{X}} \pi(\mathbf{x}, \mathbf{y}) d\mathbf{x} = \tilde{\nu}(\mathbf{y})$ is allowed to vary from within a range $[0, c]$ of the target density ν, such that $0 \leq \frac{\tilde{\nu}(\mathbf{y})}{\nu(\mathbf{y})} \leq c$. The density $\tilde{\nu}$ can be interpreted as a reweighted target density $\tilde{\nu}(\mathbf{y}) = \omega(\mathbf{y})\nu(\mathbf{y}), \quad 0 \leq \omega(\mathbf{y}) \leq c$, where portions of the support can be up-weighted while others are down-weighted or removed. The relaxed constraint is equivalent to a case of the partial optimal transport relaxations using f-divergences introduced by Séjourné et al. (2023),

$$\inf_{\pi} \int_{\mathcal{X}\times\mathcal{Y}} \mathsf{c}(\mathbf{x}, \mathbf{y})\pi(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} + \mathcal{D}_{\imath_{[a,b]}}(\tilde{\nu} \| \nu) \quad \text{s.t.} \int_{\mathcal{Y}} \pi(\mathbf{x}, \mathbf{y}) d\mathbf{y} = \mu(\mathbf{x}), \tag{23}$$

where $\mathcal{D}_{\imath_{[a,b]}}$ is the range divergence with $\imath_{[a,b]}$ being the convex indicator function,

$$\imath_{[a,b]}(r) = \begin{cases} 0, & r \in [a,b] \\ +\infty, & \text{o.w.} \end{cases}, \quad \imath^*_{[a,b]}(t) = \sup_{u \in [a,b]} (u \cdot t) = \max(-at, bt), \tag{24}$$

where $\imath^*_{[a,b]}$ denotes its Legendre-Fenchel conjugate. Since the function $\imath_{[a,b]}$ is convex lower semi-continuous, therefore $\imath_{[a,b]} = \imath^{**}_{[a,b]}$, we can apply the variational form of the f-divergence (Nguyen et al., 2010), (Polyanskiy & Wu, 2024, Theorem 7.26), exploited by f-GAN (Nowozin et al., 2016), leading to a form requiring only expected values

$$\mathcal{D}_\imath(\tilde{\nu} \| \nu) = \int_{\mathcal{Y}} \imath_{[a,b]}\left(\frac{\tilde{\nu}}{\nu}(\mathbf{y})\right) \nu(\mathbf{y}) d\mathbf{y} = \sup_{\eta} \underbrace{\int_{\mathcal{Y}} \eta(\mathbf{y})\tilde{\nu}(\mathbf{y}) d\mathbf{y}}_{\mathbb{E}_{\tilde{\mathbf{y}}\sim\tilde{\nu}}[\eta(\tilde{\mathbf{y}})]} - \underbrace{\int_{\mathcal{Y}} \imath^*_{[a,b]}(\eta(\mathbf{y}))\nu(\mathbf{y}) d\mathbf{y}}_{\mathbb{E}_{\mathbf{y}\sim\nu}[\imath^*_{[a,b]}(\eta(\mathbf{y}))]}. \tag{25}$$

To match 23, we focus on $a = 0$ and $b = c \geq 1$, such that $\imath^*_{[0,c]}(t) = c \cdot \max(0, t)$ and for notational compactness denote $\eta_+(\mathbf{y}) = \max(0, \eta(\mathbf{y}))$. Introducing ψ as a measurable function to act as a Lagrange multiplier to enforce the constraint in 23 and combining with 25 yields the problem

$$\inf_{\pi} \sup_{\psi,\eta} \int (\mathtt{c}(\mathbf{x}, \mathbf{y}) + \eta(\mathbf{y}) - \psi(\mathbf{x}))\pi(\mathbf{x}, \mathbf{y})d\mathbf{x}d\mathbf{y} + \int \psi(\mathbf{x})\mu(\mathbf{x})d\mathbf{x} - c \int \eta_+(\mathbf{y})\nu(\mathbf{y})d\mathbf{y}. \quad (26)$$

As described in Appendix A.2, since $\mathtt{c}$ is convex and lower semi-continuous, we interchange the $\inf_\pi$ and $\sup_\eta$ and apply what is known as the $\mathtt{c}$-transform of $-\eta(\mathbf{y})$ (Santambrogio, 2015; Villani et al., 2009) to obtain the dual problem with measurable map $T : \mathcal{X} \to \mathcal{Y}$

$$\sup_{\eta} \inf_{T} \underbrace{\int_{\mathcal{X}} \big(\mathtt{c}(\mathbf{x}, T(\mathbf{x})) + \eta(T(\mathbf{x}))\big)\mu(\mathbf{x})d\mathbf{x}}_{\mathbb{E}_{\mathbf{x}\sim\mu}[\mathtt{c}(\mathbf{x},T(\mathbf{x}))+\eta(T(\mathbf{x}))]} - \underbrace{c \int_{\mathcal{Y}} \eta_+(\mathbf{y})\nu(\mathbf{y})d\mathbf{y}}_{\mathbb{E}_{\mathbf{y}\sim\nu}[c\cdot\max(0,\eta(\mathbf{y}))]}, \quad (27)$$

For computational implementation T and η are parameterized using neural networks with associated parameters θ_T and θ_η and expectations are estimated using samples from μ and ν as described in the Algorithm 2.

Algorithm 2: `Static-Neural-SS`: Learning Algorithm for Static Subset Alignment

Require: Source distribution μ and target distribution ν, cost function $\mathtt{c}(\cdot, \cdot)$, reweighting bound c, neural networks $T(\cdot; \theta_T)$ and $\eta(\cdot; \theta_\eta)$, batch size N, number of updates n_T and n_η, optimizers optim_T and optim_η.
Ensure: Trained neural estimator for transport map T.

```
for ℓ = 1, 2, ..., L do
    for k = 1, 2, ..., n_T do
        Sample {x_i}_{i=1}^N ~ μ and {y_j}_{j=1}^N ~ ν
        grad_{θ_T} ← ∇_{θ_T} (1/N) Σ_{i=1}^N [c(x_i, T(x_i; θ_T)) + η(T(x_i; θ_T); θ_η)]
        Using optim_T update θ_T via gradient descent
    end for
    for k = 1, 2, ..., n_η do
        Sample {x_i}_{i=1}^N ~ μ and {y_j}_{j=1}^N ~ ν
        grad_{θ_η} ← ∇_{θ_η} [ (c/N) Σ_{j=1}^N max(0, η(y_j; θ_η)) − (1/N) Σ_{i=1}^N η(T(x_i; θ_T); θ_η) ]
        Using optim_η update θ_η via gradient ascent
    end for
end for
```

A.2 Derivation of Static Neural Subset Selection

We denote the problem expressed in 26 as $\inf_\pi \sup_\psi \sup_\eta \mathcal{L}(\pi, \psi, \eta)$, where the Lagrangian is

$$\mathcal{L}(\pi, \psi, \eta) = \int_{\mathcal{X}\times\mathcal{Y}} (\mathtt{c}(\mathbf{x}, \mathbf{y}) + \eta(\mathbf{y}) + \psi(\mathbf{x}))\, \pi(\mathbf{x}, \mathbf{y})d\mathbf{x}d\mathbf{y} + \int_{\mathcal{X}} \psi(\mathbf{x})\, \mu(\mathbf{x})d\mathbf{x} - c \int_{\mathcal{Y}} \eta_+(\mathbf{y})\nu(\mathbf{y})d\mathbf{y}. \quad (28)$$

We proceed to interchange the $\sup$ and $\inf$,[1] which is allowed due to the strong duality property associated with optimal transport when the cost $\mathtt{c}$ is convex and lower semi-continuous,

$$\inf_{\pi} \sup_{\psi,\eta} \mathcal{L}(\pi, \psi, \eta) = \sup_{\eta,\psi} \inf_{\pi} \mathcal{L}(\pi, \psi, \eta). \quad (29)$$

optimizing with respect to π for given η and ψ, the integrand in the first term of 28 is unbounded from below at any point $\mathtt{c}(\mathbf{x}, \mathbf{y}) + \eta(\mathbf{y}) - \psi(\mathbf{x}) < 0$. Thus, η and ψ need to ensure that $\mathtt{c}(\mathbf{x}, \mathbf{y}) + \eta(\mathbf{y}) -$

[1]This requires us to verify Slater's constraint qualifications, which are: (i) primal is convex wrt π ,(which is obvious), (ii) dual is concave wrt η, which is also obvious (iii) relative interior for inequality constraints set is non-empty, which can be verified by looking at the fact that for any $\tilde{\nu}$ the distribution $\pi(\mathbf{x}, \mathbf{y}) = \mu(\mathbf{x})\tilde{\nu}(\mathbf{y})$ is feasible and one can see that if one defines the feasible set of coupling $\Pi_c(\mu, \nu) = \{\pi \in \mathcal{P}(\mathcal{X}\times\mathcal{Y}) : \pi_\mathcal{X} = \mu, \pi_\mathcal{Y} \leq c\nu\}$, then for $\forall\, 1 \leq c_0 \leq c_1, \Pi_{c_0}(\mu, \nu) \subseteq \Pi_{c_1}(\mu, \nu)$, which in other words mean that $\Pi_{c=1}(\mu, \nu)$ is a subset of feasible solutions for all values of $c > 1$, therefore relative-interior in non-empty.

$\psi(\mathbf{x}) \geq 0$. This constraint requires that, for any $\mathbf{x} \in \text{supp}(\mu) \subseteq \mathcal{X}$, $\psi(\mathbf{x}) = \inf_{\mathbf{y}\in\mathcal{Y}}(\mathtt{c}(\mathbf{x},\mathbf{y}) + \eta(\mathbf{y}))$ (Villani et al., 2009, Theorem 5.10 and Remark 5.13) . This definition of ψ corresponds to the c-transform of $-\eta(\mathbf{y})$ in the optimal transport literature (Santambrogio, 2015; Villani et al., 2009). Then, the inner infimum with respect to π is attained with zero value if $\forall\,(\mathbf{x},\mathbf{y}) \in (\mathcal{X}\times\mathcal{Y})$: $\pi(\mathbf{x},\mathbf{y}) > 0 \implies \mathtt{c}(\mathbf{x},\mathbf{y}) + \eta(\mathbf{y}) - \psi(\mathbf{x}) = 0$. Therefore, the dual problem becomes

$$\sup_{\eta} \int_{\mathcal{X}} \left(\inf_{\mathbf{y}\in\mathcal{Y}} (\mathtt{c}(\mathbf{x},\mathbf{y}) + \eta(\mathbf{y}) \right) \mu(\mathbf{x}) d\mathbf{x} - c \int_{\mathcal{Y}} \eta_+(\mathbf{y})\, \nu(\mathbf{y}) d\mathbf{y}. \tag{30}$$

$$= \sup_{\eta} \inf_{T} \int_{\mathcal{X}} \big(\mathtt{c}(\mathbf{x}, T(\mathbf{x})) + \eta(T(\mathbf{x}))\big) \mu(\mathbf{x}) d\mathbf{x} - c \int_{\mathcal{Y}} \eta_+(\mathbf{y})\, \nu(\mathbf{y}) d\mathbf{y} \tag{31}$$

$$= \sup_{\eta} \inf_{T} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mu} \big[\mathtt{c}(\mathbf{x}, T(\mathbf{x})) + \eta(T(\mathbf{x}))\big] - c \mathop{\mathbb{E}}_{\mathbf{y}\sim\nu} [\eta_+(\mathbf{y})]. \tag{32}$$

$$= \inf_{T} \sup_{\eta} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mu} \big[\mathtt{c}(\mathbf{x}, T(\mathbf{x}))\big] + \mathop{\mathbb{E}}_{\mathbf{x}\sim\mu} \big[\eta(T(\mathbf{x})\big] - c \mathop{\mathbb{E}}_{\mathbf{y}\sim\nu} [\eta_+(\mathbf{y})]. \tag{33}$$

$$= \inf_{T} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mu} \big[\mathtt{c}(\mathbf{x}, T(\mathbf{x}))\big] + \mathcal{D}_{\imath_{[0,c]}}(T_\sharp\mu \| \nu) = \inf_{\substack{T \\ T_\sharp\mu \leq c\nu}} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mu} \big[\mathtt{c}(\mathbf{x}, T(\mathbf{x}))\big]. \tag{34}$$

Equation 30 and equation 31 are equal due to a theorem by Rockafellar (1976, Theorem 3A); equation 31 and equation 32 are equivalent by definition; equation 32 and equation 33 are equivalent since the function is convex with respect to T and concave with respect to η; and equation 33 and equation 34 are equivalent by the variational formula for the f-divergence. Thus, we obtain a relaxed Monge formulation.

A.3 Derivation of Dynamic Neural Subset Selection

The modified Benamou-Brenier problem takes the following minimax form

$$\begin{aligned} &\inf_{\rho_t, \mathbf{v}_t} \int_0^1 \mathop{\mathbb{E}}_{x\sim\rho_t} \left[\frac{\|\mathbf{v}_t(\mathbf{x})\|^2}{2}\right] dt + \sup_{\eta:\Omega\to\mathbb{R}} \mathbb{E}_{\mathbf{x}\sim\rho_1} [\eta(\mathbf{x})] - \mathop{\mathbb{E}}_{\mathbf{x}\sim\nu} \left[\imath^*_{[0,c]}(\eta(\mathbf{x}))\right] \\ &\text{s.t.} \quad \frac{\partial}{\partial_t}\rho_t(\mathbf{x}) + \text{div}(\rho_t(\mathbf{x})\mathbf{v}_t(\mathbf{x})) = 0, \quad \rho_0(\mathbf{x}) = \mu(\mathbf{x}) \end{aligned} \tag{35}$$

Combining the objective and constraints in equation 35 to obtain the Lagrangian

$$\begin{aligned} \mathcal{L}(\mathbf{v}_t, \rho_t, \psi_0, \varphi_t, \eta) = &\overbrace{\int_0^1\int_\Omega \frac{\|\mathbf{v}_t(\mathbf{x})\|^2}{2}\rho_t(\mathbf{x})d\mathbf{x}dt}^{\text{(I)}} + \overbrace{\int_\Omega \psi_0(\mathbf{x})\left(\rho_0(\mathbf{x}) - \mu(\mathbf{x})\right)d\mathbf{x}}^{\text{(II)}} \\ &+ \overbrace{\int_0^1\int_\Omega \varphi_t(\mathbf{x})\frac{\partial}{\partial_t}\rho_t(\mathbf{x})d\mathbf{x}dt}^{\text{(III)}} + \overbrace{\int_0^1\int_\Omega \varphi_t(\mathbf{x})\text{div}\left(\rho_t(\mathbf{x})\mathbf{v}_t(\mathbf{x})\right)d\mathbf{x}dt}^{\text{(IV)}} \\ &+ \overbrace{\sup_\eta\left(\int_\Omega \eta(\mathbf{x})\rho_1(\mathbf{x})d\mathbf{x} - c\int_\Omega \max(0,\eta(\mathbf{x}))\nu(\mathbf{x})d\mathbf{x}\right)}^{\text{(V)}}. \end{aligned} \tag{36}$$

Since $\rho_t(\mathbf{x})$ is supported on a bounded subset $\Omega \subset \mathbb{R}^d$, one can change the order of integration. Therefore, for term (III), by changing the order of integration and then computing integration by parts one obtains,

$$\begin{aligned} \text{(III)} = \int_\Omega\int_0^1 \varphi_t(\mathbf{x})\frac{\partial}{\partial_t}\rho_t(\mathbf{x})d\mathbf{x}dt &= \int_\Omega \varphi_1(\mathbf{x})\rho_1(\mathbf{x})d\mathbf{x} - \int_\Omega \varphi_0(\mathbf{x})\rho_0(\mathbf{x})d\mathbf{x} \\ &\quad - \int_0^1\int_\Omega \frac{\partial}{\partial_t}\varphi_t(\mathbf{x})\rho_t(\mathbf{x})d\mathbf{x}dt. \end{aligned} \tag{37}$$

To simplify (IV), we can use product rule of derivatives to write

$$\varphi_t(\mathbf{x})\text{div}(\rho_t(\mathbf{x})\mathbf{v}_t(\mathbf{x})) = \text{div}\left(\varphi_t(\mathbf{x})\rho_t(\mathbf{x})\mathbf{v}_t(\mathbf{x})\right) - \rho_t(\mathbf{x})\langle\nabla\varphi_t(\mathbf{x}), \mathbf{v}_t(\mathbf{x})\rangle.$$

Therefore by combining the above identity with Gauss's theorem, one obtains

$$\begin{aligned}(\text{IV}) &= \int_0^1\int_\Omega \operatorname{div}\left(\varphi_t(\mathbf{x})\rho_t(\mathbf{x})\mathbf{v}_t(\mathbf{x})\right) d\mathbf{x}dt - \int_0^1\int_\Omega \rho_t(\mathbf{x})\langle\nabla\varphi_t(\mathbf{x}), \mathbf{v}_t(\mathbf{x})\rangle d\mathbf{x}dt \\ &= \underbrace{\int_0^1\oint_{\partial\Omega} \varphi_t(\mathbf{x}_t)\rho_t(\mathbf{x}_t)\langle\mathbf{v}_t(\mathbf{x}), d\boldsymbol{n}\rangle dt}_{=0} - \int_0^1\int_\Omega \rho_t(\mathbf{x})\langle\nabla\varphi_t(\mathbf{x}), \mathbf{v}_t(\mathbf{x})\rangle d\mathbf{x}dt.\end{aligned} \tag{38}$$

From the boundary condition on optimal transport (see the discussion above equation 20, and also (Figalli & Glaudo, 2023, section 4.1)) , the first part of the right-hand side of equation 38 is zero; therefore,

$$(\text{IV}) = \int_0^1\int_\Omega \varphi_t(\mathbf{x})\cdot\operatorname{div}(\rho_t(\mathbf{x})\mathbf{v}_t(\mathbf{x}))d\mathbf{x}dt = -\int_0^1\int_\Omega \langle\nabla\varphi_t(\mathbf{x}), \mathbf{v}_t(\mathbf{x})\rangle\rho_t(\mathbf{x})d\mathbf{x}dt. \tag{39}$$

To eliminate the primal variable $\mathbf{v}_t(\mathbf{x})$, substitute equation 37 and equation 39 into equation 36 and compute the variational derivative to obtain the stationary condition. For that one can write the terms of Lagrangian depending on $\mathbf{v}_t(\mathbf{x})$ as

$$\tilde{\mathcal{L}}(\mathbf{v}_t) = \int_0^1\int_\Omega \left(\frac{\|\mathbf{v}_t(\mathbf{x})\|^2}{2} - \langle\nabla\varphi_t(\mathbf{x}), \mathbf{v}_t(\mathbf{x})\rangle\right)\rho_t(\mathbf{x})d\mathbf{x}dt. \tag{40}$$

With the additive perturbation function $\boldsymbol{\tau}_t$ vanishing at $t=0$ and $t=1$ and a scalar ε, the Lagrangian $\tilde{\mathcal{L}}(\mathbf{v}_t+\varepsilon\boldsymbol{\tau}_t)$ is

$$\begin{aligned}\tilde{\mathcal{L}}(\mathbf{v}_t+\varepsilon\boldsymbol{\tau}_t) &= \int_0^1\int_\Omega \left(\frac{\|\mathbf{v}_t(\mathbf{x})+\varepsilon\boldsymbol{\tau}_t(\mathbf{x})\|^2}{2} - \left\langle\nabla\varphi_t(\mathbf{x}),\ \mathbf{v}_t(\mathbf{x})+\varepsilon\boldsymbol{\tau}_t(\mathbf{x})\right\rangle\right)\rho_t(\mathbf{x})d\mathbf{x}dt \\ &= \int_0^1\int_\Omega \left(\frac{\|\mathbf{v}_t(\mathbf{x})\|^2}{2} - \left\langle\nabla\varphi_t(\mathbf{x}), \mathbf{v}_t(\mathbf{x})\right\rangle\right)\rho_t(\mathbf{x})d\mathbf{x}dt \\ &\quad + \int_0^1\int_\Omega \left(\varepsilon^2\frac{\|\boldsymbol{\tau}_t(\mathbf{x})\|^2}{2} + \varepsilon\left\langle\mathbf{v}_t(\mathbf{x}) - \nabla\varphi_t(\mathbf{x}),\ \boldsymbol{\tau}_t(\mathbf{x})\right\rangle\right)\rho_t(\mathbf{x})d\mathbf{x}dt,\end{aligned}$$

and variational derivative is

$$\delta\tilde{\mathcal{L}}(\mathbf{v}_t(\mathbf{x}))\Big|_{\mathbf{v}_t} = \frac{d}{d\varepsilon}\Big|_{\varepsilon=0}\tilde{\mathcal{L}}(\mathbf{v}_t+\varepsilon\boldsymbol{\tau}_t) = \int_0^1\int_\Omega \langle\mathbf{v}_t(\mathbf{x}) - \nabla\varphi_t(\mathbf{x}),\ \boldsymbol{\tau}_t(\mathbf{x})\rangle\rho_t(\mathbf{x})d\mathbf{x}dt. \tag{41}$$

The stationarity condition requires $\delta_{\mathbf{v}_t}\tilde{\mathcal{L}}(\mathbf{v}_t(\mathbf{x})) = 0$. For arbitrary perturbation $\boldsymbol{\tau}_t(\mathbf{x})$, the variational-derivative $\delta_{\mathbf{v}_t}\tilde{\mathcal{L}}(\mathbf{v}_t(\mathbf{x})) = 0$ if and only if

$$\mathbf{v}_t(\mathbf{x}) = \nabla\varphi_t(\mathbf{x}). \tag{42}$$

Therefore, one can write the Lagrangian as

$$\begin{aligned}\mathcal{L}(\rho_t,\psi_0,\varphi_t,\eta) &= \int_\Omega \psi_0(\mathbf{x})\cdot(\rho_0(\mathbf{x}) - \mu(\mathbf{x}))\,d\mathbf{x} + \int_\Omega \varphi_1(\mathbf{x})\rho_1(\mathbf{x})d\mathbf{x} - \int_\Omega \varphi_0(\mathbf{x})\rho_0(\mathbf{x})d\mathbf{x} \\ &\quad + \int_\Omega \eta(\mathbf{x})\rho_1(\mathbf{x})d\mathbf{x} - c\cdot\int_\Omega \max(0,\eta(\mathbf{x}))\nu(\mathbf{x})d\mathbf{x} \\ &\quad - \int_0^1\int_\Omega \left(\frac{\partial}{\partial t}\varphi_t(\mathbf{x}) + \frac{\|\nabla\varphi_t(\mathbf{x})\|^2}{2}\right)\rho_t(\mathbf{x})d\mathbf{x}dt.\end{aligned}$$

Similarly, by computing $\delta_{\psi_0}\mathcal{L}$, $\delta_{\rho_1}\mathcal{L}$, and using stationary conditions, one obtains the condition,

$$\psi_0(\mathbf{x}) = \varphi_0(\mathbf{x}), \tag{43}$$

$$\eta(\mathbf{x}) = -\varphi_1(\mathbf{x}). \tag{44}$$

Therefore, Lagrangian is simplified to

$$\begin{aligned}\mathcal{L}(\rho_t,\varphi_t,\eta) &= -\int_\Omega \varphi_0(\mathbf{x})\mu(\mathbf{x})d\mathbf{x} - c\int_\Omega \max(0,-\varphi_1(\mathbf{x}))\nu(\mathbf{x})d\mathbf{x} \\ &\quad - \int_0^1\int_\Omega \left(\frac{\partial}{\partial t}\varphi_t(\mathbf{x}) + \frac{\|\nabla\varphi_t(\mathbf{x})\|^2}{2}\right)\rho_t(\mathbf{x})d\mathbf{x}dt.\end{aligned} \tag{45}$$

The simplified problem (equation 7 in the main body) is

$$\sup_{\rho_t}\inf_{\varphi_t}\mathop{\mathbb{E}}_{\mathbf{x}\sim\mu}\left[\varphi_0(\mathbf{x})\right] + c\cdot\mathop{\mathbb{E}}_{\mathbf{x}\sim\nu}\left[\max(0,-\varphi_1(\mathbf{x}))\right] + \int_0^1\mathop{\mathbb{E}}_{\mathbf{x}_t\sim\rho_t}\left[\frac{\partial}{\partial t}\varphi_t(\mathbf{x}_t) + \frac{\|\nabla\varphi_t(\mathbf{x}_t)\|^2}{2}\right]dt.$$

B THRESHOLDING FOR PU-LEARNING AND REJECTION SAMPLING

Our idea of rejection sampling and thresholding for PU-learning relies on the fact that the dual form of range divergence is zero when the supremum in the dual is attained by the function $\eta^\star(\mathbf{x})$ with $\tilde{\nu}(\mathbf{x}) = \rho_1^\star(\mathbf{x})$ i.e.

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim\tilde{\nu}}[\eta^\star(\mathbf{x})] - c \mathop{\mathbb{E}}_{\mathbf{x}\sim\nu}[\texttt{ReLU}(\eta^\star(\mathbf{x}))] = 0 \tag{46}$$

By defining $\mathcal{A}{=}\text{supp}(\nu)$, $\tilde{\mathcal{A}}{=}\text{supp}(\tilde{\nu})$, and $\bar{\mathcal{A}}{=}\mathcal{A}/\tilde{\mathcal{A}}$, one can also see that $\tilde{\mathcal{A}} \cap \bar{\mathcal{A}} = \emptyset$, therefore one can write equation 46 as

$$\int_{\tilde{\mathcal{A}}} \eta^\star(\mathbf{x})\tilde{\nu}(\mathbf{x})d\mathbf{x} - c\int_{\tilde{\mathcal{A}}} \texttt{ReLU}\left(\eta^\star(\mathbf{x})\right)\nu(\mathbf{x})d\mathbf{x} - c\int_{\bar{\mathcal{A}}} \texttt{ReLU}(\eta^\star(\mathbf{x}))\nu(\mathbf{x})d\mathbf{x} = 0. \tag{47}$$

One can further write

$$\eta^\star(\mathbf{x}) = \texttt{ReLU}\left(\eta^\star(\mathbf{x})\right) - \texttt{ReLU}\left(-\eta^\star(\mathbf{x})\right). \tag{48}$$

After substituting equation 48 into equation 47 one obtains

$$\overbrace{\int_{\tilde{\mathcal{A}}} \texttt{ReLU}(\eta^\star(\mathbf{x}))(\tilde{\nu}(\mathbf{x}) - c\nu(\mathbf{x}))d\mathbf{x}}^{\text{LHS}} = \overbrace{\int_{\tilde{\mathcal{A}}} \texttt{ReLU}(-\eta^\star(\mathbf{x}))\tilde{\nu}(\mathbf{x})d\mathbf{x} + c\int_{\bar{\mathcal{A}}} \texttt{ReLU}(\eta^\star(\mathbf{x}))\nu(\mathbf{x})d\mathbf{x}}^{\text{RHS}} \tag{49}$$

The dual form equation 47 is optimal with zero duality gap, if the primal form satisfies $\forall \mathbf{x} \in \mathcal{A},\ \imath_{[0,c]}\left(\frac{\tilde{\nu}}{\nu}(\mathbf{x})\right) = 0$, which can also be restricted to $\forall\, \mathbf{x} \in \tilde{\mathcal{A}}\ \imath_{[0,c]}\left(\frac{\tilde{\nu}}{\nu}(\mathbf{x})\right) = 0$. This is equivalent to $\tilde{\nu}(\mathbf{x}) \leq c\nu(\mathbf{x})$ almost-everywhere in $\tilde{\mathcal{A}}$. Therefore, one can say that $0 \geq$ LHS and also

$$0 \geq \overbrace{\int_{\tilde{\mathcal{A}}} \texttt{ReLU}(-\eta^\star(\mathbf{x}))\nu(\mathbf{x})d\mathbf{x} + c\int_{\bar{\mathcal{A}}} \texttt{ReLU}(\eta^\star(\mathbf{x}))\nu(\mathbf{x})d\mathbf{x}}^{\text{RHS}} \tag{50}$$

We can now see that both integrands in equation 49 are nonnegative and sum to a value less than or equal to zero, which is only possible if both are equal to zero. Therefore, one can write

$$0 = \int_{\tilde{\mathcal{A}}} \texttt{ReLU}(-\eta^\star(\mathbf{x}))\nu(\mathbf{x})d\mathbf{x} + c\int_{\bar{\mathcal{A}}} \texttt{ReLU}(\eta^\star(\mathbf{x}))\nu(\mathbf{x})d\mathbf{x} \tag{51}$$

Further, two non-negative integrals are evaluated on two mutually exclusive sets; therefore, to have a sum equal to zero value, we can conclude that each integral is zero individually. Therefore, we can write

$$0 = \int_{\tilde{\mathcal{A}}} \texttt{ReLU}(-\eta^\star(\mathbf{x}))\nu(\mathbf{x})d\mathbf{x} = c\int_{\bar{\mathcal{A}}} \texttt{ReLU}(\eta^\star(\mathbf{x}))\nu(\mathbf{x})d\mathbf{x} \tag{52}$$

Equation 52 is therefore equivalent to the following element-wise test:

$$\begin{aligned} \eta^\star(\mathbf{x}) &\geq 0, \text{ almost surely in } \tilde{\mathcal{A}} \\ \eta^\star(\mathbf{x}) &< 0, \text{ almost surely in } \bar{\mathcal{A}} \end{aligned} \tag{53}$$

Additionally, from the equation 50, one can also conclude that

$$\overbrace{\int_{\tilde{\mathcal{A}}} \texttt{ReLU}(\eta^\star(\mathbf{x}))(\tilde{\nu}(\mathbf{x}) - c\nu(\mathbf{x}))d\mathbf{x}}^{\text{LHS}} = 0, \tag{54}$$

which is a complementary slackness condition in the sense that $\tilde{\nu}(\mathbf{x}) < c\nu(\mathbf{x}) \implies \eta^*(\mathbf{x}) = 0$ almost everywhere in $\mathcal{A}$. During the neural network training with finite data-points, the potential function η is usually sub-optimal, and its sign cannot be relied on; therefore, instead of directly using the sign, one can sort values of potential at data points and select a predetermined proportion (prior) of data-points. Therefore, for training PU-learning models, we applied both sign-based and sorting-based filtration of data. From Figures 7, 8 and 9, one can observe that the optimal potential for static problem exactly follows equation 53, whereas in the dynamic case the sign of φ_1 is inverted, which is due to the relation obtained in equation 44, which ensures that for the at optimal $\varphi_1^\star$ following relation holds

$$\begin{aligned} \varphi_1^\star(\mathbf{x}) &\leq 0, \text{ almost surely in } \tilde{\mathcal{A}} \\ \varphi_1^\star(\mathbf{x}) &> 0, \text{ almost surely in } \bar{\mathcal{A}}. \end{aligned} \tag{55}$$

C SURVEY OF RECENT WORK ON NEURAL OPTIMAL TRANSPORT

In this section, we discuss the recent related work on computational optimal transport and its applications. More specifically, we consider the works that are related to neural estimation of optimal transport maps with occasional reference to theoretical developments.

C.1 STATIC NEURAL OPTIMAL TRANSPORT

Seguy et al. (2018) employed stochastic gradient-based approaches in one of the earliest works to estimate the optimal transport map using neural networks. Notably, the work by Seguy et al. (2018) differed from early work by Genevay et al. (2016) in the sense that the later work employed stochastic gradient-based methods to estimate the transport plan for large-scale data, whereas earlier work by Genevay et al. (2016) only minimized the optimal transport loss using stochastic-gradient-based methods. This is also in contrast to the well-known Wasserstein-GAN (Arjovsky et al., 2017; Gulrajani et al., 2017) that employs the Kantorovich-Rubinstein duality to minimize the Wasserstein-1 loss function for generative modeling, where neural networks are employed as parameterizations for both dual-potential and data generator, but do not provide transport plans. Finally, the Sinkhorn-GAN employs an approximation of the discrete Wasserstein distance between latent representations of data and that of samples from non-informative prior (Genevay et al., 2018) for generative modeling. Now, we can see clear distinction between two different classes of approaches employing Wasserstein distances in generative modeling, the first class of works concerns with employing Wasserstein distance as a loss for generative modeling, without any explicit concern for obtaining the underlying transport plan across the distributions (Arjovsky et al., 2017; Gulrajani et al., 2017). The second class seeks to learn a transport plan to realize the generative model.

Efforts to learn Monge maps were motivated by a theorem by Brenier (1991), which essentially states that, for continuous distributions with squared-Euclidean transportation cost, the optimal solution of the Monge problem is the gradient of a convex function (Figalli & Glaudo, 2023, Theorem 2.5.10). Therefore, initially, gradients of input-convex neural networks (ICNN) (Amos et al., 2017) were employed to estimate the transport plan for the Wasserstein-2 distance (Makkuva et al., 2020; Korotin et al., 2021a;b). This approach has also been employed to supervised conditional neural Monge maps (Bunne et al., 2022a) and unbalanced optimal transport (Lübeck et al., 2022). The study by Amos et al. (2023) focuses on the development of an efficient neural optimal solution that could be implemented quickly in more practical scenarios. This approach to solve Wasserstein-2 distances employing convex potentials involves computationally challenging evaluation of the Fenchel conjugate of an ICNN-parameterized convex function. More recent work in this direction focuses on improved optimization strategies and better ICNN architectures to bypass problems related to Fenchel conjugate evaluations and ICNN training (Amos, 2023; Vesseron & Cuturi, 2024). Recent work also focuses on some batch-based methods, which have been devised to improve the regularity of learned neural Monge maps (Uscidda & Cuturi, 2023; Eyring et al., 2024).

Another recent direction of work is based on the idea that ICNNs can be overly restrictive; therefore, more general neural network architectures should be employed to directly parameterize the transport maps (Rout et al., 2022; Korotin et al., 2023b). The work by Fan et al. (2022a; 2023) focuses on employing neural networks to approximate the solution for Monge's transport problem and also draws inspiration from the recent developments in neural-network-based parametric realizations for approximating Kantorovich plans. Recently, neural optimal transport has also been extended to unbalanced transportation settings (Yang & Uhler, 2019; Choi et al., 2023). Another work directly related to the static subset selection problem is the inexact optimal transport Gazdieva et al. (2023).

Unless there is a corresponding Monge mapping (Choi et al., 2024a; Mokrov et al., 2024; Geuter et al., 2025), optimal transport requires a stochastic transport plan. A recent body of work (Korotin et al., 2023b;a; Asadulaev et al., 2024) deals with learning transportation plans using a weaker formulation of optimal transport (Gozlan et al., 2017; Backhoff-Veraguas et al., 2019) along with noise outsourcing techniques, which have also been extended to more general costs. Apart from the applications in image translation (Korotin et al., 2023b), neural optimal transport has been applied for biomedical image registration (Kim et al., 2024) and to study single-cell perturbations (Bunne et al., 2023). Neural optimal transport has also been employed for metric learning (Howard et al., 2024; Scarvelis & Solomon, 2023).

C.2 Dynamic Neural Optimal Transport

The potential applications of dynamic optimal transport in the cellular trajectory inference (Tong et al., 2020) and its connections with flow-based models for generative modeling (Huang et al., 2021; Huguet et al., 2022) has been instrumental in the recent research developments in this direction. Jordan-Kinderlehrer-Otto flow (JKO) is a time discretization scheme to solve Wasserstein gradient flows for different energy functionals (Jordan et al., 1998; Santambrogio, 2017). Therefore, a lot of effort done in that regard is focused on neural network-parameterized schemes to solve the JKO-flow problem for both cellular trajectory inference and generative modeling (Ma et al., 2021; Fan et al., 2022b; Lambert et al., 2022; Bunne et al., 2022b; Xu et al., 2023; Choi et al., 2023; 2024b; Altekrüger et al., 2023; Mokrov et al., 2021; Alvarez-Melis et al., 2022). The JKO scheme has also been studied for the applications related to molecular discovery (Alvarez-Melis et al., 2022). A recent study deals with convergence properties of JKO-based generative models (Cheng et al., 2024).

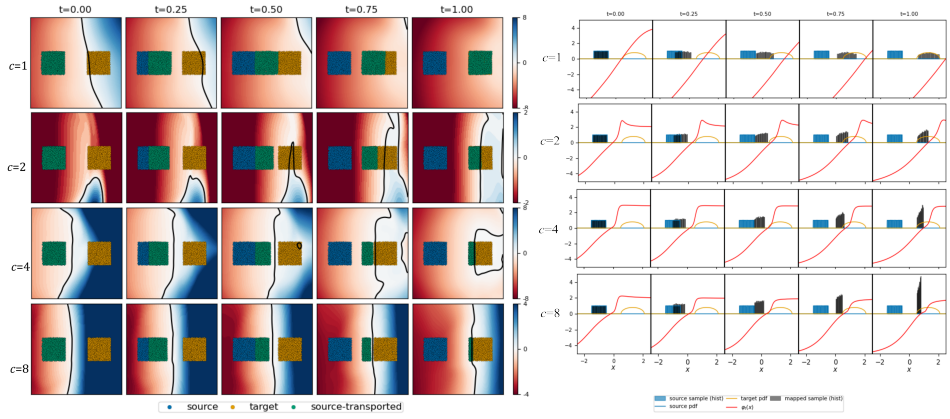
Recent developments in flow-matching models based on flow matching (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2023) for generative modeling led to even more interest in the development of algorithms to solve dynamic optimal transportation problems. The action-matching-based framework led to the development of a more general framework to solve both trajectory inference and generative modeling problems (Neklyudov et al., 2023a) for the cases where one could also sample from the trajectory between two terminal marginals. Rectified flow matching (Liu et al., 2023; 2024b) uses the neural optimal transport in an additional rectification step to improve the linearity of flows so that after training the model, images could be generated efficiently with only a single-step integration along straight-line paths. For generative modeling, in contrast to target-conditional flow matching (Lipman et al., 2023), where during training, flows are conditioned on target samples, discrete optimal transport-conditioned flow matching employs the mini-batch optimal transport to create the conditionals (Pooladian et al., 2023; 2024; Tong et al., 2024b). Another recent work (Kornilov et al., 2024) attempts to alleviate the error accumulation problems associated with mini-batch optimal transport by learning straight paths between source and target distributions in a single step. Flow-matching (Albergo & Vanden-Eijnden, 2023; Albergo et al., 2025), diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2020; Song et al., 2021a), and Schrödinger bridges (Wang et al., 2021; Liu et al., 2022; 2024a; Shi et al., 2023; Gushchin et al., 2023b;a; Somnath et al., 2023) are deeply interconnected under the framework of generalized bridge matching (Tong et al., 2024b; Albergo et al., 2025; Tong et al., 2024a; Shi et al., 2023). Recently, there have also been attempts to understand diffusion models as approaches to minimize the dynamic Wasserstein distances (Kwon et al., 2022; Khrulkov et al., 2023). Recent works generalize flow matching from different perspectives, Chen & Lipman (2024) generalize the flow matching to the flows on Riemannian manifolds, Atanackovic et al. (2025) attempt to extend the flow models to return meaningful flows for the data beyond training distributions, and Haviv et al. (2025) generalize the flow matching to the cases where data can be treated as distributions of distributions.

Additionally, there has been a recent dynamic extension to the conditional neural optimal transport (Hosseini et al., 2025; Kerrigan et al., 2024). There have also been efforts to study neural network-based scalable approaches to solve high-dimensional partial differential equations (Wan et al., 2023).

D Additional Toy Results

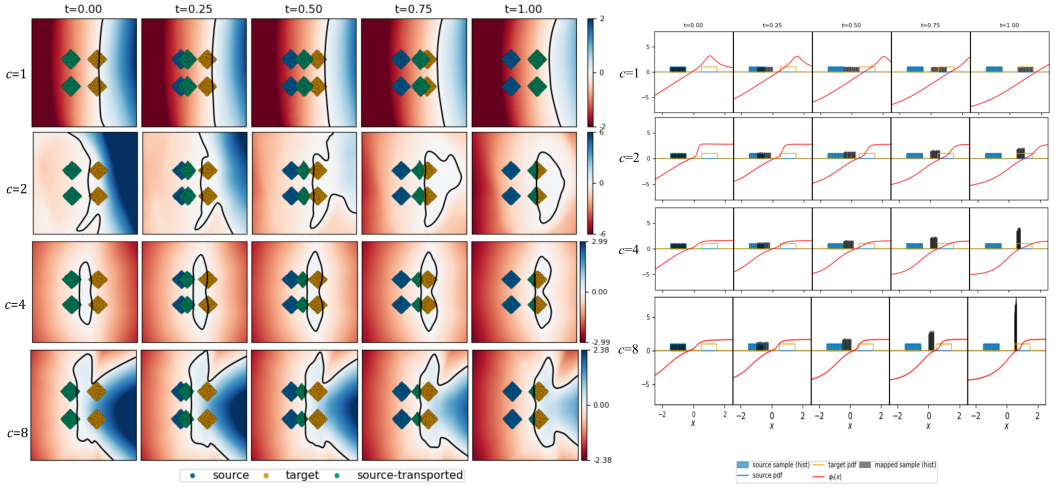
For the toy results given in Figures 1 and 2 the temporal snapshots of the evolution of points sampled from source along with the potential φ_t are given in Figures 7a and 7b, respectively. Similar temporal snapshots between two 2D bimodal uniform squares and two 1D uniform distributions are shown in Figure 8a and 8b respectively.

The static framework for subset alignment is equivalent to the relaxations of the inexact optimal transport problem discussed by Gazdieva et al. (2023). Our approach for static subset alignment is discussed in Algorithm 2 and Appendix A.1.



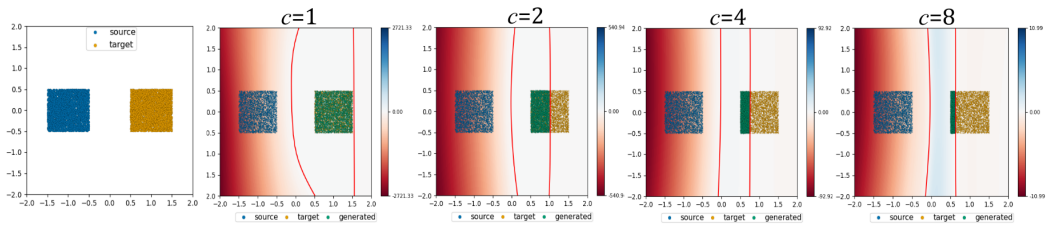
(a) (b)

Figure 7: Temporal snapshots at $t \in \{0.0, 0.25, 0.50, 0.75, 1.0\}$ for $c \in 1, 2, 4, 8$. (a) Evolution between two 2D uniform distributions shown in Figure 1; the black contour line indicates the zero-level set. (b) Evolution from 1D uniform to semicircular density is shown in Figure 2 with support subset alignment. $\varphi_t(\mathbf{x})$ is denoted by red curve.

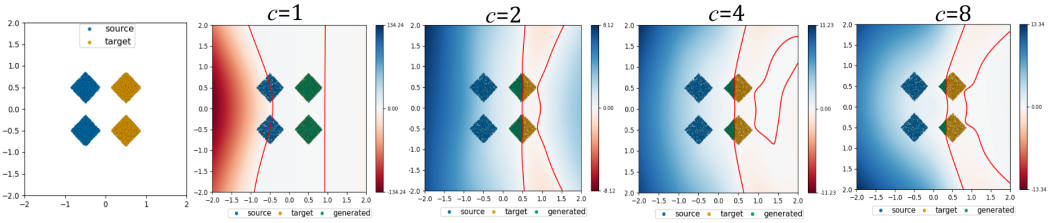


(a) (b)

Figure 8: Temporal snapshots at $t \in \{0.0, 0.25, 0.50, 0.75, 1.0\}$ for $c \in 1, 2, 4, 8$. (a) Evolution between two 2D bimodal distributions; the black contour line indicates the zero-level set of $\varphi_t(\mathbf{x})$. (b) Evolution from 1D uniform to another 1D uniform density, with dynamic subset subset alignment. $\varphi_t(\mathbf{x})$ is denoted by red curve.

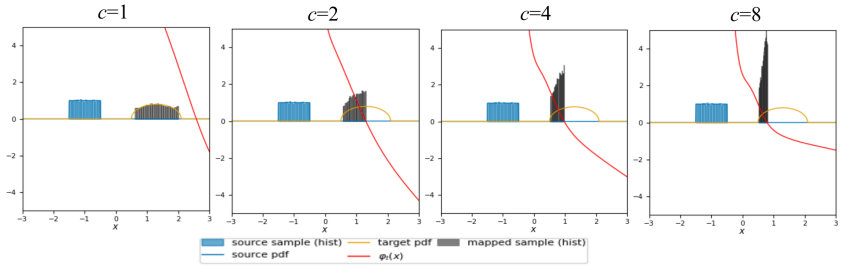


(a)

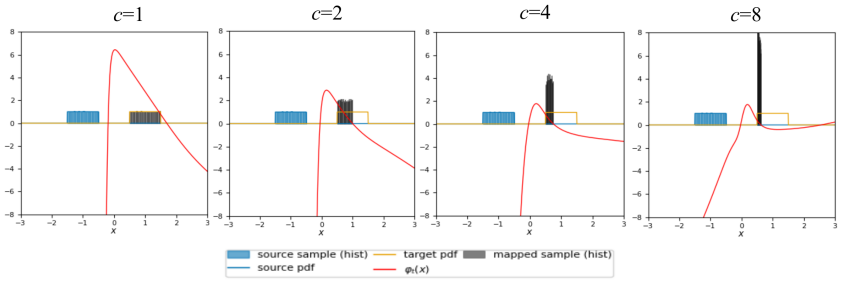


(b)

Figure 9: Static subset alignment results for $c \in \{1, 2, 4, 8\}$. (a) Translation between two 2D uniform distributions shown in Figure 1. (b) 2D bimodal distributions shown in Figure 8a. Red curves indicate the zero-level sets of $\eta(\mathbf{x})$.



(a)



(b)

Figure 10: Static subset alignment results for $c \in \{1, 2, 4, 8\}$. (a) Transport from 1D uniform distribution to semicircular density shown in Figure 7b. (b) Transport from 1D uniform distribution to a uniform distribution shown in Figure 8b. Red curves indicate the $\eta(\mathbf{x})$.

E Implementation Details

E.1 Toy Results

For dynamic subset alignment, we use the model architectures in Listings 1 and 2, where `Smodel` and `Qmodel` parameterize θ_φ and θ_ρ, respectively. Listing 1 is used for the 1D toy experiments, while Listing 2 is used for the 2D toy experiments. We set `num_hid`$= 512$ for all dynamic architectures. For both settings, we use Algorithm 1 with $n_\varphi = n_\rho = 1$. We train for 50,000 iterations in 1D and 100,000 iterations in 2D. In all dynamic toy experiments, we use the Adam optimizer (Kingma & Ba, 2017) with a learning rate of 1×10^{-4} and a batch size of 512.

```python
import jax
from jax import numpy as jnp
from flax import linen as nn
import math
from ml_collections import config_dict

class Smodel(nn.Module):
    num_hid: int
    num_out: int
    use_layernorm: bool = True
    boundary_zero: bool = False

    @nn.compact
    def __call__(self, t, x):
        if jnp.ndim(t) == 0:
            t = jnp.broadcast_to(t, x.shape[:-1] + (1,))
        elif t.shape[-1] != 1:
            t = t.reshape(x.shape[:-1] + (1,))

        h = jnp.concatenate([t, x], axis=-1)
        for _ in range(4):
            h = nn.Dense(self.num_hid)(h)
            if self.use_layernorm:
                h = nn.LayerNorm()(h)
            h = nn.swish(h)

        v = nn.Dense(self.num_out)(h)
        return t * (1.0 - t) * v if self.boundary_zero else v

class Qmodel(nn.Module):
    num_hid: int
    num_out: int

    @nn.compact
    def __call__(self, t, x_0, x_1):
        h = jnp.concatenate([t, x_0, x_1, t < 0.5], axis=-1)
        for _ in range(3):
            h = nn.swish(nn.Dense(self.num_hid)(h))

        h = nn.Dense(self.num_out)(h)
        return (1.0 - t) * x_0 + t * (x_1 + h)
```

Listing 1: Model architectures used for toy 1D dynamic subset alignment experiments.

```python
import jax
from jax import numpy as jnp
from flax import linen as nn
import math
from ml_collections import config_dict

class Smodel(nn.Module):
    num_hid: int
    num_out: int

    @nn.compact
    def __call__(self, t, x):
        if jnp.ndim(t) == 0:
            t = jnp.broadcast_to(t, x.shape[:-1] + (1,))
        h = jnp.concatenate([t, x], axis=-1)
        for _ in range(3):
            h = nn.swish(nn.Dense(self.num_hid)(h))
        return nn.Dense(self.num_out)(h)

class Qmodel(nn.Module):
    num_hid: int
    num_out: int

    @nn.compact
    def __call__(self, t, x_0, x_1):
        h = jnp.concatenate([t, x_0, x_1, t < 0.5], axis=-1)
        for _ in range(3):
            h = nn.swish(nn.Dense(self.num_hid)(h))
        h = nn.Dense(self.num_out)(h)
        return (1.0 - t) * x_0 + t * (x_1 + h)
```

Listing 2: Model architectures used for toy 2D dynamic subset alignment experiments

For static subset alignment, we use the model architecture in Listing 3 for both the 1D and 2D toy experiments. The `Tmodel` parameterizes the transport map T via a residual connection, while `DualNET` parameterizes the dual function η. The same architecture is used in both settings, with the input and output dimensions adjusted accordingly. We set `num_hid`$= 512$ for all static architectures. For both settings, we use Algorithm 2 with $n_T = 5$ and $n_\eta = 1$. We train for 20,000 iterations in 1D and 50,000 iterations in 2D. In all static toy experiments, we use the Adam optimizer (Kingma & Ba, 2017) with a learning rate of 1×10^{-4} and a batch size of 512.

E.2 EMNIST CLASSIFIER

We merge all letters into a single class and treat each digit as its own class (digits 0–9 keep their numeric label, and all letters are labeled 10). To mitigate class imbalance during training, we use a

```python
import jax
from jax import numpy as jnp
from flax import linen as nn
import math
from ml_collections import config_dict

class Tmodel(nn.Module):
    num_hid: int
    num_out: int

    @nn.compact
    def __call__(self, inputs):
        h = inputs
        for _ in range(4):
            h = nn.swish(nn.Dense(self.num_hid)(h))
        h = nn.Dense(self.num_out)(h)
        return inputs + h

class DualNET(nn.Module):
    num_hid: int
    num_out: int

    @nn.compact
    def __call__(self, x):
        h = x
        for _ in range(4):
            h = nn.swish(nn.Dense(self.num_hid)(h))
        return nn.Dense(self.num_out)(h)
```

Listing 3: Model architectures used for 1D and 2D static subset alignment toy experiments.

class-reweighted SoftMax loss. For k-class classification, let $\boldsymbol{z} \in \mathbb{R}^k$ denote the per-class counts in the training set; we define reweighting coefficients $\boldsymbol{\omega} \in \mathbb{R}^k$ by

$$\omega_i = \left(\sum_{j=1}^{N} \frac{z_i}{z_j} \right)^{-1}, \ \forall\, i \in [k]. \tag{56}$$

Given a one-hot label vector $\mathbf{y}$ and the network SoftMax output $\hat{\mathbf{y}}$, the reweighted loss (risk) is

$$\ell(\mathbf{y}, \hat{\mathbf{y}}) = \mathbf{1}_k^\top (\boldsymbol{\omega} \odot \mathbf{y} \odot \hat{\mathbf{y}}) \tag{57}$$

We train the EMNIST classifier using the train/validation split provided with the EMNIST dataset (Cohen et al., 2017). The model is ResNet-18 (He et al., 2016) and is optimized with the class-reweighted SoftMax loss in equation 57. We use Adam (Kingma & Ba, 2017) with a warmup–cosine learning-rate schedule (Loshchilov & Hutter, 2017), a peak learning rate of 1×10^{-3}, and 500 warmup steps. The cosine schedule decays over 20,000 steps to an end learning rate of 1×10^{-5}. Training is stopped after 20,000 steps, once the classifier exceeds 90% overall validation accuracy and 99% accuracy on digits. Confusion matrices are reported in Appendix F.

E.3 MNIST-EMNIST Translation Models

For dynamic subset selection, following Neklyudov et al. (2023a), we parameterize the vector field φ_t with a U-Net equipped with DDPM-style time embeddings (Song & Ermon, 2020). As in action matching (Neklyudov et al., 2023a), φ_t returns a scalar via

$$\varphi_t(\mathbf{x}) = \langle \mathrm{UNet}(x), x \rangle. \tag{58}$$

Likewise, Q_t, which parameterizes ρ_t, is implemented as a U-Net with time embeddings. For both U-Nets, we use the same architectural settings: `nonlinearity = swish`, `nf = 64`, `ch_mult = (1, 2, 2, 2)`, `num_res_blocks = 2`, and `attn_resolutions = (16, 8)`. We use the AdamW optimizer with learning-rate scheduling for 50,000 training iterations. The optimizer parameters are $\beta = (0, 0.999)$, `weight decay`$= 0.1$, and `dropout`$= 0.1$. We additionally employ exponential moving averages (EMA) during training with an EMA rate of 0.999. These settings are very similar to those used in rectified flow matching and action matching (Liu et al., 2023; Neklyudov et al., 2023a). All input images in neural subset alignment are centered at $[0.5, 0.5]$ and resized to $32 \times 32 \times 1$. The learning rate is linearly increased from zero to its maximum value during the first 5,000 iterations and is then kept constant. The maximum learning rates are 2×10^{-4} and 1×10^{-4} for φ_t and Q_t, respectively. Additionally, gradients are clipped element-wise to the interval $[-1, 1]$. Algorithm 1 is employed for 50,000 training iterations with $n_\varphi = 2$ and $n_\rho = 1$. A batch size of 128 is used to train the dynamic models for MNIST-EMNIST translation.

For static domain translation, the transport network T is a U-Net Ronneberger et al. (2015) with base factor 64, and the critic network η is a ResNet; we use the same choices as Gazdieva et al. (2023) for inexact extremal optimal transport. All inputs to the static models are centered at $[0.5, 0.5]$ and resized to $32 \times 32 \times 1$. We train T and η with Adam Kingma & Ba (2017) using an initial learning rate of 1×10^{-4}, which is halved after $10{,}000 + 5{,}000c$, $20{,}000 + 5{,}000c$, $30{,}000 + 5{,}000c$, $40{,}000 + 5{,}000c$, and $70{,}000 + 5{,}000c$ steps. We run Algorithm 2 for $50{,}000$ learning iterations with $n_T = 10$ and $n_\eta = 1$. A batch size of 128 is used for static MNIST–EMNIST translation.

E.4 Models for PU-Learning Using Subset Alignment

The architectures used for PU learning with dynamic subset alignment are given in Listing 4, where `Smodel` and `Qmodel` parameterize φ_t and ρ_t, respectively. For both models, we use `num_hid` $= 1024$ and `embedding_dim` $= 64$. The output dimension is `num_out` $= 1$ for `Smodel`, while for `Qmodel` it equals the dataset dimension (Table 7). We run Algorithm 1 for $50{,}000$ training iterations with $n_\varphi = 2$ and $n_\rho = 1$. Both models are optimized with AdamW using learning rate 5×10^{-6}, $\beta_1 = 0.9$, and $\epsilon = 10^{-8}$. We use $10{,}000$ warmup steps and gradient clipping with maximum norm 0.1, and we maintain an exponential moving average (EMA) with rate 0.999. Batch sizes are dataset-dependent and are reported in Table 7.

The architectures used for PU learning with static subset alignment are given in Listing 5, where `Tmodel` and `DualNET` parameterize T and η, respectively. For both models, we use `num_hid` $= 1024$ and `embedding_dim` $= 128$. The output dimension for `Tmodel` matches the dataset dimension, while `DualNET` uses `num_out` $= 1$. We run Algorithm 2 for $20{,}000$ training iterations with $n_T = 10$ and $n_\eta = 1$. Both models are optimized with learning rate 10^{-4} and use EMA with rate 0.999. Batch sizes are dataset-dependent and are reported in Table 7. For all PU-learning experiments, we fix $c = \frac{1}{\pi_+}$.

Table 7: UCI datasets for PU learning, along with the total number of data points (n), dimension (dim), positive prior (π) and batch sizes employed in training the corresponding dynamic models.

Dataset	data-set size	dimension	π	batch size
Abalone	4177	8	0.16	20
Banknote	1372	4	0.44	10
Breast-w	699	9	0.34	10
Diabetes	768	8	0.35	6
Haberman	306	3	0.26	6
Heart	270	13	0.44	6
Ionosphere	351	34	0.64	6
Isolet	7797	617	0.04	4
Jm1	10885	21	0.19	20
Kc1	2109	21	0.15	20
Madelon	2600	500	0.5	20
Musk	6598	166	0.15	20
Segment	2310	19	0.14	20
Semeion	1593	256	0.1	4
Sonar	208	60	0.53	4
Spambase	4601	57	0.39	20
Vehicle	846	18	0.26	6
Waveform	5000	40	0.34	20
Wdbc	569	30	0.37	6
Yeast	1484	8	0.31	10

E.5 Image-to-Image Translation on FFHQ

For dynamic subset alignment, the architectures for the networks parameterizing φ_t and ρ_t are given in Listing 6, where `Smodel` and `Qmodel` parameterize φ_t and ρ_t respectively . Both networks use `t_embed_dim`$= 128$ and `num_hid`$= 1024$, with `num_out`$= 1$ for the φ_t network and `num_out`$= 512$ for the ρ_t network (matching the FFHQ latent dimensionality). We use AdamW with a fixed learning rate of 1×10^{-5}, gradient clipping with maximum norm 1.0, EMA rate 0.999, and batch size 128. We run Algorithm 1 for $500{,}000$ learning iterations with $n_\varphi = 2$ and $n_\rho = 1$.

```python
import jax
from jax import numpy as jnp
from flax import linen as nn
import math
from ml_collections import config_dict

def get_timestep_embedding(timesteps, embedding_dim: int,  max_positions: float = 10_000.0):
  t = jnp.asarray(timesteps)
  t = t.reshape((-1,)) if t.ndim == 1 else t.reshape((t.shape[0], -1))[:, 0]
  t = t.astype(jnp.float32)

  half = embedding_dim // 2
  if half < 1:
    return jnp.zeros((t.shape[0], embedding_dim), dtype=t.dtype)

  freqs = jnp.exp(-jnp.log(jnp.array(max_positions,dtype=t.dtype))
      * jnp.arange(half, dtype=t.dtype) / (half - 1 + 1e-8))

  args = t[:, None] * freqs[None, :]          # (B, half)
  emb = jnp.concatenate([jnp.cos(args), jnp.sin(args)], axis=-1)  # (B, 2*half)

  if embedding_dim % 2 == 1:
    emb = jnp.pad(emb, ((0, 0), (0, 1)))

  return emb[:, :embedding_dim]

class Smodel(nn.Module):
  config: config_dict.ConfigDict
  @nn.compact
  def __call__(self, t, x, *, train: bool = True):
    config = self.config
    time_emb_dim = config.embedding_dim
    num_hid = config.num_hid
    num_out = config.num_out
    x = jnp.asarray(x)
    if x.ndim == 1:                # (D,) -> (1,D)
      x = x[None, :]

    # make t broadcastable to x batch: (B,1)
    t = jnp.asarray(t)
    t = jnp.broadcast_to(t, x.shape[:-1] + (1,))

    temb = get_timestep_embedding(t, time_emb_dim)  # should return (B, time_emb_dim)
    temb = nn.swish(nn.Dense(num_hid)(temb))
    temb = nn.Dense(num_hid)(temb)          # (B, num_hid)

    h = jnp.concatenate([t, x], axis=-1)
    h = nn.swish(nn.Dense(num_hid)(h))
    h = h + temb

    for _ in range(3):
      h = nn.swish(nn.Dense(num_hid)(h))

    return nn.Dense(num_out)(h)

class Qmodel(nn.Module):
  config: config_dict.ConfigDict

  @nn.compact
  def __call__(self, t, x_0, x_1, *, train: bool = True):
    config = self.config
    time_emb_dim = config.embedding_dim
    num_hid = config.num_hid
    num_out = config.num_out

    x_0 = jnp.asarray(x_0)
    x_1 = jnp.asarray(x_1)
    if x_0.ndim == 1:                # (D,) -> (1,D)
      x_0 = x_0[None, :]
      x_1 = x_1[None, :]

    # broadcast t to (B,1) to match x batch
    t = jnp.asarray(t)
    t = jnp.broadcast_to(t, x_0.shape[:-1] + (1,))     # (B,1)

    # time embedding -> project to hidden width
    temb = get_timestep_embedding(t, time_emb_dim)     # (B, time_emb_dim)
    temb = nn.swish(nn.Dense(num_hid)(temb))
    temb = nn.Dense(num_hid)(temb)                     # (B, num_hid)

    heaviside_step = (t < 0.5).astype(x_0.dtype)       # (B,1)
    h = jnp.concatenate([t, x_0, x_1, heaviside_step], axis=-1)

    h = nn.swish(nn.Dense(num_hid)(h))
    h = h + temb

    for _ in range(3):
      h = nn.swish(nn.Dense(num_hid)(h))

    h = nn.Dense(num_out)(h)

    x_t = (1.0 - t) * x_0 + t * (x_1 + h)
    return x_t
```

Listing 4: Model architectures for PU-Learning for dynamic subset alignment

```python
import jax
from jax import numpy as jnp
from flax import linen as nn
import math
from ml_collections import config_dict

class etamodel(nn.Module):
    num_hid: int
    num_out: int
    @nn.compact
    def __call__(self, x):
        h = x
        for _ in range(3):
            h = nn.swish(nn.Dense(self.num_hid)(h))
        return nn.Dense(self.num_out)(h)

class Tmodel(nn.Module):
    num_hid: int
    num_out: int
    @nn.compact
    def __call__(self, x):
        h = x
        for _ in range(4):
            h = nn.swish(nn.Dense(self.num_hid)(h))
        h = nn.Dense(self.num_out)(h)
        residual = nn.Dense(self.num_out)(x)
        return h + residual
```

Listing 5: Model architectures for PU learning with static subset alignment

```python
import jax
from jax import numpy as jnp
from flax import linen as nn
import math

def get_timestep_embedding(t, dim, max_positions=10000):
    t = jnp.reshape(t, (-1,))
    half = dim // 2
    scale = jnp.exp(
        jnp.arange(half, dtype=jnp.float32)
        * (-math.log(max_positions) / (half - 1))
    )
    emb = t[:, None] * scale[None, :]
    emb = jnp.concatenate([jnp.sin(emb), jnp.cos(emb)], axis=1)
    return jnp.pad(emb, [(0, 0), (0, dim % 2)])

class Smodel(nn.Module):
    num_hid: int
    num_out: int
    t_embed_dim: int

    @nn.compact
    def __call__(self, t, x):
        if t.ndim == 0:
            t = jnp.broadcast_to(t, (x.shape[0], 1, 1))

        e = get_timestep_embedding(t, self.t_embed_dim)
        e = jnp.reshape(e, x.shape[:-1] + (self.t_embed_dim,))

        h = jnp.concatenate([e, x], axis=-1)
        for i in range(4):
            h = nn.swish(nn.Dense(self.num_hid)(h))
            if i < 3:
                h = jnp.concatenate([e, h], axis=-1)

        return nn.Dense(self.num_out)(h)

class Qmodel(nn.Module):
    num_hid: int
    num_out: int
    t_embed_dim: int

    @nn.compact
    def __call__(self, t, x_0, x_1):
        e = get_timestep_embedding(t.squeeze(), self.t_embed_dim)
        h = jnp.concatenate([e, x_0, x_1, t < 0.5], axis=-1)

        for i in range(4):
            h = nn.swish(nn.Dense(self.num_hid)(h))
            if i < 3:
                h = jnp.concatenate([e, h], axis=-1)

        h = nn.Dense(self.num_out)(h)
        return (1.0 - t) * x_0 + t * (x_1 + h)
```

Listing 6: Model architectures for FFHQ dynamic subset alignment.

For static subset alignment, the neural network architectures `TransportNET` and `DualNET` given in Listing 7 parameterize T and η, respectively. Both networks use `hidden_dim`$= 1024$. `TransportNET` maps 512-dimensional FFHQ latents to 512-dimensional outputs, while `DualNET` returns a scalar. We use AdamW with a fixed learning rate of 1×10^{-5}, weight decay 10^{-8}, EMA rate 0.999, and batch size $1{,}024$. We run Algorithm 2 for $50{,}000$ learning iterations with $n_T = 5$ and $n_\eta = 1$. The same architectures are used for the UOTM experiments in Table 4.

```python
import jax
from jax import numpy as jnp
from flax import linen as nn
import math

class TransportNET(nn.Module):
    input_dim: int
    hidden_dim: int
    @nn.compact
    def __call__(self, x):
        h = x
        for _ in range(3):
            h = nn.swish(nn.Dense(self.hidden_dim)(h))
        return nn.Dense(self.input_dim)(h) + nn.Dense(self.input_dim)(x)

class DualNET(nn.Module):
    hidden_dim: int
    @nn.compact
    def __call__(self, x):
        h = x
        for _ in range(3):
            h = nn.swish(nn.Dense(self.hidden_dim)(h))
        return nn.Dense(1)(h)
```

Listing 7: Model architectures for FFHQ static subset alignment.

E.6 Melanoma Experiment

Unlike the other experiments in this paper, the melanoma setting involves multiple intermediate marginals that must be matched. Because we want to preserve all marginals except the terminal one, we generalize equation 8 to the multi-marginal setting. Assuming we can sample from the marginals at times t_k and t_{k+1}, for $t \in [t_k, t_{k+1}]$ we define $\tau = \frac{t-t_k}{t_{k+1}-t_k}$ and use the interpolant

$$\mathbf{x}_t = (1-\tau)\,\mathbf{x}_{t_k} + \tau\,\mathbf{x}_{t_{k+1}} + w_k(\tau)\,Q\big(t, \mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}; \theta_\rho\big), \tag{59}$$

where

$$w_k(\tau) = \begin{cases} 1-\tau^2-(1-\tau)^2, & k < K-1, \\ 1-(1-\tau)^2, & k = K-1, \end{cases} \tag{60}$$

where Q is a neural network shared across segments. Since w_k vanishes at $\tau = 0$ for all k and at $\tau = 1$ for $k < K-1$, the interpolant matches $\mu_0, \ldots, \mu_{K-1}$ exactly, while $w_{K-1}(1) = 1$ leaves the terminal marginal unconstrained: $\mathbf{x}_{t_K} \mapsto \mathbf{x}_{t_K} + Q(t_K, \mathbf{x}_{t_{K-1}}, \mathbf{x}_{t_K}; \theta_\rho)$. For $K = 1$, this reduces to $(1-t)\mathbf{x}_0 + t\mathbf{x}_1 + t(2-t)\,Q$, which, like $(1-t)\mathbf{x}_0 + t(\mathbf{x}_1 + Q)$, fixes μ_0 while relaxing the terminal marginal. In addition, we use the Wasserstein gradient-flow updates of Neklyudov et al. (2024) to improve stability and convergence.

The Algorithm 3 summarizes the training procedure for the melanoma experiment. The Neural Network Architectures employed in Melanoma experiments are provided in Listing 8. We should emphasize that the interpolation equation 59 has been implemented as part of interpolant neural network architecture, but for clear description of learning procedure, we included interpolation equation a part of learning algorithm.

We use the first 50 principal components of the melanoma data (Bunne et al., 2023), centered and divided by the largest per-dimension standard deviation of the training set. The three observed populations are ordered in reverse time, μ_0 (48h), μ_1 (24h) and μ_2 (8h), at normalised times $t_0 = 0$, $t_1 = 0.6$ and $t_2 = 1$ ($K = 2$), so that the sub-setting constraint acts on the $\mu_{K=2}$. The potential $\varphi_t(\mathbf{x}; \theta_\varphi) = \langle f(t, \mathbf{x}; \theta_\varphi), \mathbf{x} \rangle$ is parameterized by a `Smodel` in Listing 8 with `nf` $= 512$ and activations `act` $=$ `Swish`; the time t is mapped by a two-layer embedding network of width 512 and added to the input of each of the three inner hidden layers. The interpolant network $Q_t(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}; \theta_\rho)$ is parameterized by `Qmodel` in Listing 8, which takes concatenation $(t, \mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}})$ as input and has a single hidden layer of width 512 with swish activation and a linear output in $\mathbb{R}^{50}$. Both networks are trained with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 10^{-4}) at learning rate 2×10^{-4} with a linear warm-up over the first $5{,}000$ iterations and element-wise gradient clipping at 1, using batch size $N = 512$, $n_\varphi = n_\rho = 1$ update per iteration, $M = 10$ Wasserstein gradient steps with step size $\eta = 10^{-2}$, and $200{,}000$ iterations; the times t^i are drawn from a low-discrepancy sequence on $[0, 1]$. We report results for sub-setting constants $c \in \{1.0, 2.0, 4.0, 8.0\}$ in table 5. For evaluation we use an exponential moving average of the parameters with rate 0.999, and transport cells with the learned velocity field $\nabla\varphi_t$ by integrating the ODE from t_K back to t_0 with the explicit Euler scheme and step size 1×10^{-3}.

```python
import jax
from jax import numpy as jnp
from flax import linen as nn
import math

class Smodel(nn.Module):
  config: ml_collections.ConfigDict

  @nn.compact
  def __call__(self, t: jnp.ndarray, x: jnp.ndarray, train: bool):
    config = self.config
    act = get_act(config)
    nf = config.nf

    if config.embed_time:
      temb = t
      temb = nn.Dense(nf)(temb)
      temb = nn.Dense(nf)(act(temb))
      h = x
    else:
      h = jnp.hstack([x, t])

    h = act(nn.Dense(nf)(h))
    for _ in range(config.n_layers):
      if config.embed_time:
        h += temb
      h = nn.Dropout(config.dropout)(h, deterministic=not train)
      if config.skip:
        h = act(nn.Dense(nf)(h)) + h
      else:
        h = act(nn.Dense(nf)(h))
    h = act(nn.Dense(nf)(h))
    h = nn.Dense(config.input_dim)(h)
    return (h*x).sum(1, keepdims=True)

class Qmodel(nn.Module):
  config: ml_collections.ConfigDict

  @nn.compact
  def __call__(self,t:jnp.ndarray,batch:jnp.ndarray,train:bool):
    config=self.config
    act=get_act(config)
    nf=config.nf

    timesteps,x=batch
    B=x.shape[0]
    n_marginals=timesteps.shape[1]

    t_reshaped=t.reshape(-1,1,1)
    t_right=(timesteps<t_reshaped).sum(1)
    t_right=jnp.fmax(t_right,jnp.ones_like(t_right).astype(int))
    t_left=t_right-1

    batch_idx=jnp.arange(B)
    x_left=x[batch_idx,t_left.ravel(),:]
    x_right=x[batch_idx,t_right.ravel(),:]
    t_0=timesteps[batch_idx,t_left.ravel(),:]
    t_1=timesteps[batch_idx,t_right.ravel(),:]

    temb=layers.get_timestep_embedding(t.ravel(),nf)
    temb=nn.Dense(nf)(temb)
    temb=nn.Dense(nf)(act(temb))

    h=jnp.concatenate([x_left,x_right],axis=1)
    h=jnp.hstack([t,h])

    if config.indicator:
      h=jnp.hstack([h,t<0.5*(t_0+t_1)])

    h=act(nn.Dense(nf)(h))

    for _ in range(config.n_layers):
      h=nn.Dropout(config.dropout)(h,deterministic=not train)
      h=act(nn.Dense(nf)(h))+h if config.skip else act(nn.Dense(nf)(h))

    h=nn.Dense(config.input_dim)(h)

    alpha=(t-t_0)/(t_1-t_0)
    beta=(t_1-t)/(t_1-t_0)

    w_inner=1.0-alpha**2-beta**2
    w_last=1.0-beta**2

    w=jnp.where(t_right==(n_marginals-1),w_last,w_inner)

    if config.get('pin_last',False):
      w=w_inner

    return beta*x_left+alpha*x_right+w*h
```

Listing 8: Model architectures for Melanoma Experiment.

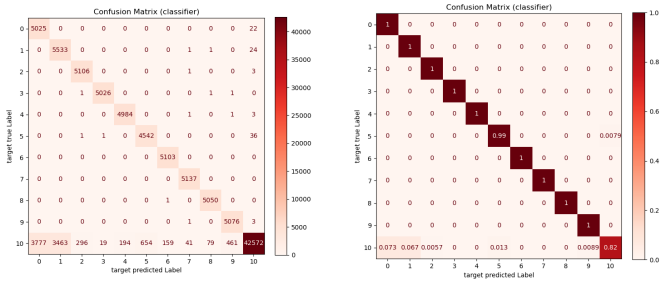
Algorithm 3: `Modified-Dynamic-Neural-SS`: Modified Learning Algorithm for Dynamic Subset Alignment on Melanoma

Require: Samples from distributions $\{\mu_k\}_{k=0}^K$ observed at times $\{t_k\}_{k=0}^K$, time-dependent neural network $\varphi_t(\cdot;\theta_\varphi)$, interpolant network $Q_t(\cdot,\cdot;\theta_\rho)$, sub-setting constant $c \geq 1$, batch size N, number of updates n_φ and n_ρ, gradient clipping parameter ε, number of Wasserstein gradient steps M with step size η, gradient based optimizers optim_φ and optim_ρ.

Ensure: Sample-based neural estimate for $\varphi_t(\cdot;\theta_\varphi)$

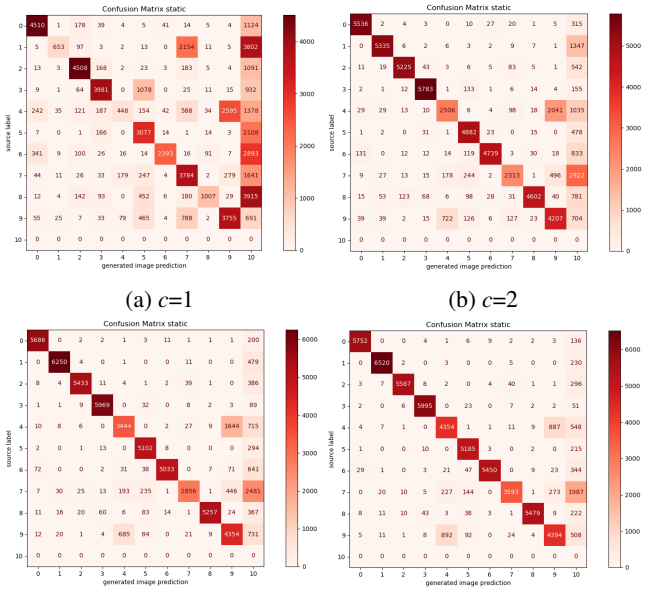
for learning iteration $\ell = 0, 1, \ldots$ **do**
  for $u = 1, \ldots, n_\varphi$ **do**
    Sample $\{\mathbf{x}_{t_k}^i\}_{i=1}^N \sim \mu_k$ for $k = 0, \ldots, K$, and $\{t^i\}_{i=1}^N \sim \text{Uniform}([0,1])$.
    $$\left\{\mathbf{x}_{t^i}^i \leftarrow (1-\tau^i)\,\mathbf{x}_{t_{k^i}}^i + \tau^i\,\mathbf{x}_{t_{k^i+1}}^i + w_{k^i}(\tau^i)\,Q_{t^i}\left(\mathbf{x}_{t_{k^i}}^i, \mathbf{x}_{t_{k^i+1}}^i;\theta_\rho\right)\right\}_{i=1}^N,$$
    where $t^i \in [t_{k^i}, t_{k^i+1}]$, $\tau^i = \frac{t^i - t_{k^i}}{t_{k^i+1} - t_{k^i}}$,
    $w_k(\tau) = 1 - \tau^2 - (1-\tau)^2$ for $k < K-1$, and $w_{K-1}(\tau) = 1 - (1-\tau)^2$
    for $j = 1, \ldots, M$ **do**
      $$\left\{\mathbf{x}_{t^i}^i \leftarrow \mathbf{x}_{t^i}^i + \eta\, w_{k^i}(\tau^i)\,\text{clip}\Big(\nabla_{\mathbf{x}}\Big[\frac{\partial}{\partial t}\varphi_{t^i}(\mathbf{x}_{t^i}^i;\theta_\varphi) + \frac{\|\nabla\varphi_{t^i}(\mathbf{x}_{t^i}^i;\theta_\varphi)\|^2}{2}\Big], -\varepsilon, \varepsilon\Big)\right\}_{i=1}^N$$
    end for
    $$\text{grad}_{\theta_\varphi} \leftarrow \nabla_{\theta_\varphi}\frac{1}{N}\sum_{i=1}^N\Big[\frac{\partial}{\partial t}\varphi_{t^i}(\mathbf{x}_{t^i}^i;\theta_\varphi) + \frac{\|\nabla\varphi_{t^i}(\mathbf{x}_{t^i}^i;\theta_\varphi)\|^2}{2} + \varphi_{t_0}(\mathbf{x}_{t_0}^i;\theta_\varphi) + c\cdot\max\Big(0, -\varphi_{t_K}(\mathbf{x}_{t_K}^i;\theta_\varphi)\Big)\Big]$$
    Using optim_φ and $\text{grad}_{\theta_\varphi}$ to update θ_φ via gradient descent.
  end for
  for $u = 1, \ldots, n_\rho$ **do**
    Sample $\{\mathbf{x}_{t_k}^i\}_{i=1}^N \sim \mu_k$ for $k = 0, \ldots, K$, and $\{t^i\}_{i=1}^N \sim \text{Uniform}([0,1])$.
    $$\left\{\mathbf{x}_{t^i}^i(\theta_\rho) \leftarrow (1-\tau^i)\,\mathbf{x}_{t_{k^i}}^i + \tau^i\,\mathbf{x}_{t_{k^i+1}}^i + w_{k^i}(\tau^i)\,Q_{t^i}\left(\mathbf{x}_{t_{k^i}}^i, \mathbf{x}_{t_{k^i+1}}^i;\theta_\rho\right)\right\}_{i=1}^N$$
    $$\text{grad}_{\theta_\rho} \leftarrow \nabla_{\theta_\rho}\frac{1}{N}\sum_{i=1}^N\Big[\frac{\partial}{\partial t}\varphi_{t^i}(\mathbf{x}_{t^i}^i(\theta_\rho);\theta_\varphi) + \frac{\|\nabla\varphi_{t^i}(\mathbf{x}_{t^i}^i(\theta_\rho);\theta_\varphi)\|^2}{2}\Big]$$
    Update θ_ρ using optim_ρ and $\text{grad}_{\theta_\rho}$ via gradient ascent.
  end for
end for

F CONFUSION MATRICES FOR MNIST TO EMNIST DOMAIN TRANSLATION



(a) (b)

Figure 11: (a) Normalized and (b) unnormalized confusion matrices for the EMNIST classifier discussed in Section 4.2.



(a) c=1 (b) c=2

(c) c=4 (d) c=8

Figure 12: Confusion matrices for MNIST→EMNIST domain translation using static subset selection. Accuracy is computed as the ratio between the trace and the sum of all entries in the confusion matrices.

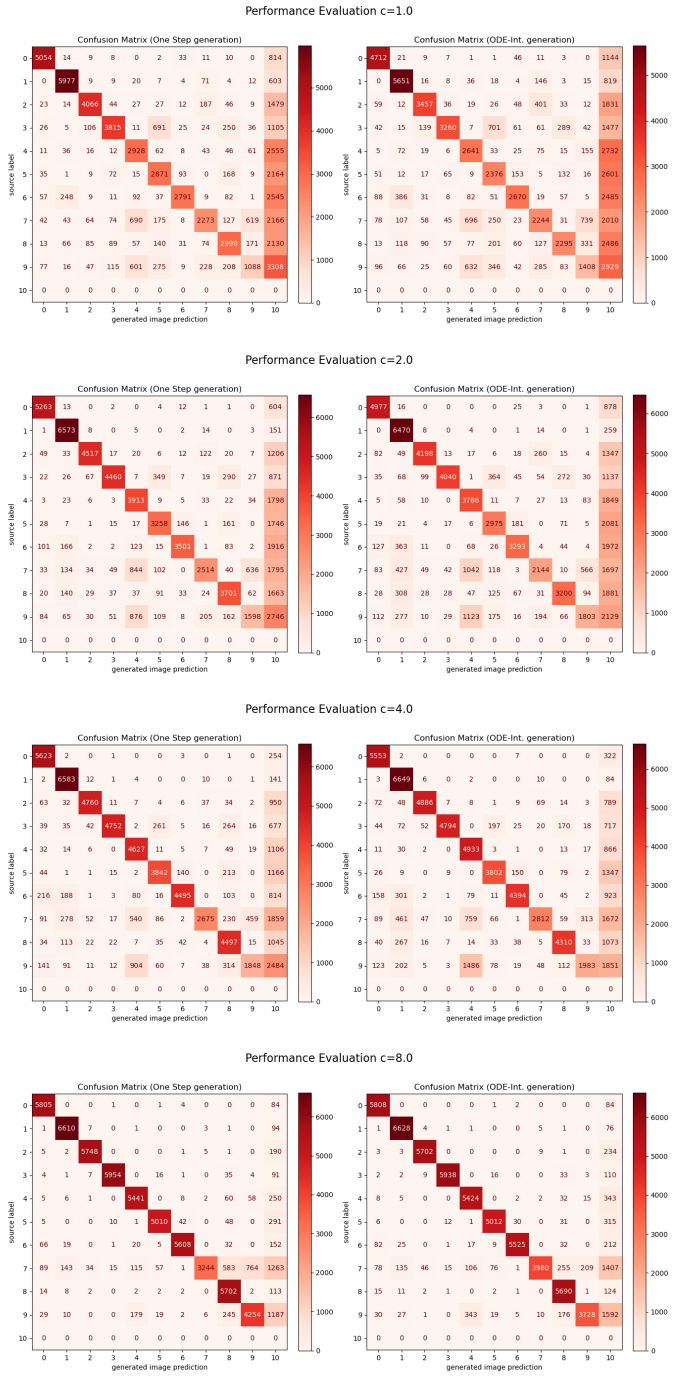


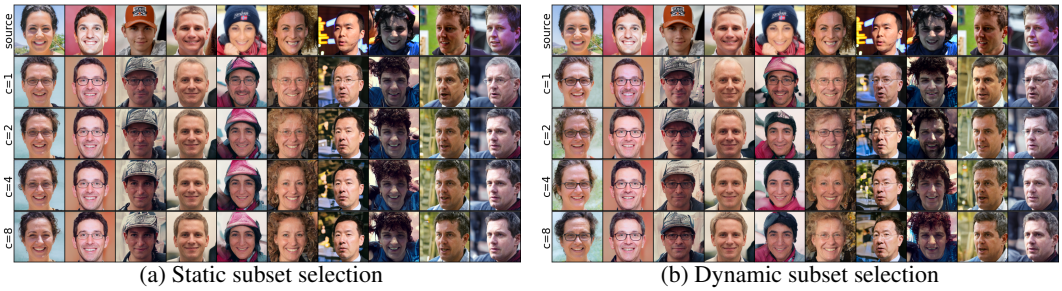
Figure 13: Confusion matrices for MNIST→EMNIST domain translation using dynamic subset selection.

G T-SNE EMBEDDINGS OF UCI DATASETS



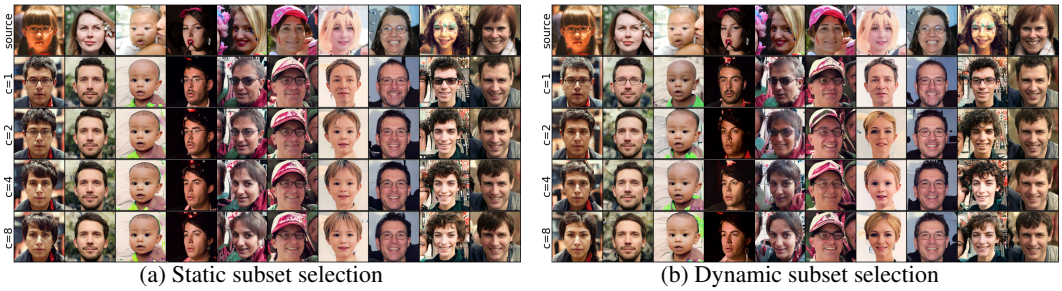
Figure 14: The t-SNE embeddings for (a) Abalone, (b) Banknote, (c) Breast-w, (d) Diabetes, (e) Haberman, (f) Heart, (g) Ionosphere, (h) Isolet, (i) Kc1, (j) Madelon, (k) Musk, (l) Semeion, (m) Sonar, (n) Spambase, (o) Vehicle, (p) Waveform, (q) Wdbc, (r) Yeast, (s) Jm1, and (t) Segment datasets.

H RESULTS FROM FFHQ



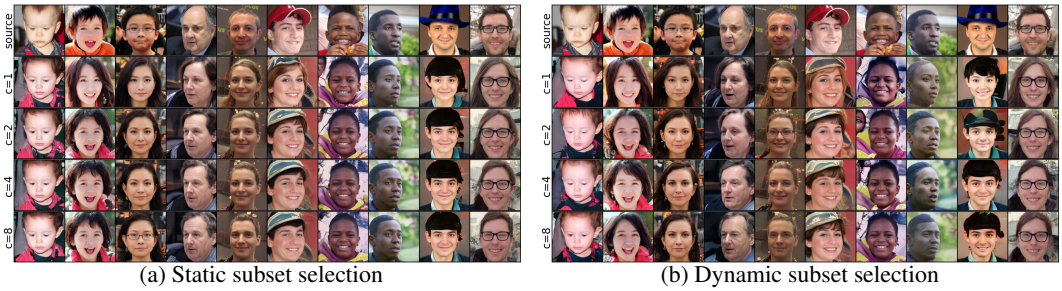
(a) Static subset selection (b) Dynamic subset selection

Figure 15: FFHQ young→old translation using (a) static and (b) dynamic subset selection. Dynamic subset selection is evaluated using Euler integration with 100 steps.



(a) Static subset selection (b) Dynamic subset selection

Figure 16: FFHQ woman→man translation using (a) static and (b) dynamic subset selection. Dynamic subset selection is evaluated using Euler integration with 100 steps.



(a) Static subset selection (b) Dynamic subset selection

Figure 17: FFHQ man→woman translation using (a) static and (b) dynamic subset selection. Dynamic subset selection is evaluated using Euler integration with 100 steps.