
\mathbb{R}^d	d -dimensional Euclidean space
$\ \cdot\ _2$	Euclidean norm
$\langle\cdot,\cdot\rangle$	standard dot product
\mathbb{S}^{d-1}	$(d-1)$ -dimensional hypersphere
θ	unit vector
\sqcup	disjoint union
$L^1(X)$	space of Lebesgue integrable functions on X
$\mathcal{P}(X)$	space of probability measures on X
$\mathcal{M}(X)$	space of measures on X
μ, ν	measures
$\delta(\cdot)$	1-dimensional Dirac delta function
$\mathcal{U}(\mathbb{S}^{d-1})$	uniform distribution on \mathbb{S}^{d-1}
$\#$	pushforward (measure)
$\mathcal{C}(X, Y)$	space of continuous maps from X to Y
$\mathbf{d}(\cdot, \cdot)$	metric in metric space
$d_{\mathcal{T}}(\cdot, \cdot)$	tree metric
$T(d)$	translation group of order d
$O(d)$	orthogonal group of order d
$E(d)$	Euclidean group of order d
g	element of group
\mathbf{W}_p	p -Wasserstein distance
\mathbf{SW}_p	Sliced p -Wasserstein distance
Λ	(rooted) subtree
e	edge in graph
w_e	weight of edge in graph
l	line, index of line
\mathcal{L}	system of lines, tree system
$\bar{\mathcal{L}}$	ground set of system of lines, tree system
\mathbb{L}_k^d	space of systems of k lines in \mathbb{R}^d
\mathcal{T}	tree structure in system of lines
L	number of tree systems
k	number of lines in a system of lines or a tree system
\mathcal{R}^α	Radon Transform on Systems of Lines
Δ_{k-1}	$(k-1)$ -dimensional standard simplex
α	splitting map
ξ	tuning parameter in splitting maps
\mathbb{T}	space of tree systems
σ	distribution on space of tree systems

524	Appendix of “Tree-Sliced Entropy Partial Transport”	
525	Table of Contents	
526	A Background on Optimal Transport on Metric Spaces with Tree Metrics	17
527	B Background on Entropy Partial Transport on Metric Spaces with Tree Metrics	17
528	C Background on Tree-Sliced Wasserstein Distance on Euclidean Spaces	20
529	C.1 Tree System	20
530	C.2 A Variant of Radon Transform for Systems of Lines	21
531	C.3 Tree-Sliced Wasserstein Distance for Probability Measures on Euclidean Spaces . .	22
532	D Theoretical Proofs	22
533	D.1 Proof for Proposition B.1	22
534	D.2 Proof for Theorem B.2	23
535	D.3 Proof for Proposition B.3	28
536	D.4 Proof for Proposition B.4	28
537	D.5 Proof for Proposition B.5	29
538	D.6 Proof for Equation (13)	30
539	D.7 Proof for Theorem 3.3	31
540	E Experimental Details	32
541	E.1 Algorithm for Partial Tree-Sliced Wasserstein Distance	32
542	E.2 Computational and Memory Complexity Analysis	33
543	E.3 Empirical Runtime and Memory Performance of Partial TSW	33
544	E.4 Discussion on hyper-parameters of evaluated methods	34
545	E.5 Comparing Computational Efficiency	34
546	E.6 Noisy Point Cloud Gradient Flow	35
547	E.7 Robust Generative Model	35
548	E.7.1 Implementation detail	35
549	E.7.2 Ablation result for baselines	37
550	E.8 Imbalance Image to Image Translation	46
551	F Boarder Impacts	47

552 A Background on Optimal Transport on Metric Spaces with Tree Metrics

553 Let $\mathcal{T} = (V, E)$ be a tree rooted at a node r , where each edge $e \in E$ is assigned a nonnegative length
 554 w_e . Here, V denotes the set of nodes and E the set of edges. For notational convenience, we use \mathcal{T}
 555 to also refer to the union of all nodes and the continuous points along the edges. We now recall the
 556 formal definition of a tree metric:

557 **Definition A.1** (Tree metric [69, Section 7, p.145–182]). A metric $d : \Omega \times \Omega \rightarrow [0, \infty)$ is said to be
 558 a *tree metric* on a set Ω if there exists a tree \mathcal{T} such that $\Omega \subset \mathcal{T}$ and, for all $x, y \in \Omega$, the distance
 559 $d(x, y)$ equals the length of the unique path in \mathcal{T} connecting x and y .

560 Suppose V is a subset of a vector space, and let $d_{\mathcal{T}}(\cdot, \cdot)$ denote the tree metric defined on \mathcal{T} . We
 561 denote by $[x, y]$ the unique shortest path in \mathcal{T} between any two points x and y . Let ω be the unique
 562 Borel (length) measure on \mathcal{T} satisfying $\omega([x, y]) = d_{\mathcal{T}}(x, y)$ for all $x, y \in \mathcal{T}$. For any $x \in \mathcal{T}$, we
 563 define the subtree rooted at x by

$$\Lambda(x) := \{y \in \mathcal{T} : x \in [r, y]\}. \quad (18)$$

564 Let $\mathcal{P}(\mathcal{T})$ denote the set of all probability measures on \mathcal{T} , i.e., Borel measures with total mass equal
 565 to one. The following result provides a closed-form expression for the 1-Wasserstein distance on the
 566 tree metric space \mathcal{T} .

567 **Theorem A.2** (Optimal Transport on Tree Metric Spaces [40, Section 3, Proposition 1]). *For any*
 568 *$\mu, \nu \in \mathcal{P}(\mathcal{T})$, the 1-Wasserstein distance with respect to the tree metric $d_{\mathcal{T}}$ is given by*

$$W_{1,d_{\mathcal{T}}}(\mu, \nu) = \int_{\mathcal{T}} |\mu(\Lambda(x)) - \nu(\Lambda(x))| \omega(dx). \quad (19)$$

569 B Background on Entropy Partial Transport on Metric Spaces with Tree 570 Metrics

571 In this section, we revisit the Entropy Partial Transport (EPT) formulation introduced in [38] for
 572 completeness. All theoretical proofs are outlined in Appendix D.

573 We denote by $\mathcal{M}(\mathcal{T})$ the collection of all nonnegative Borel measures on \mathcal{T} with finite total mass.
 574 Let $C(\mathcal{T})$ denote the space of continuous functions defined on \mathcal{T} , and let $L^\infty(\mathcal{T})$ denote the space of
 575 Borel measurable functions on \mathcal{T} that are essentially bounded with respect to the measure ω . The
 576 space $L^\infty(\mathcal{T})$ forms a Banach space when equipped with the norm

$$\|f\|_{L^\infty(\mathcal{T})} := \inf \{\bar{a} \in \mathbb{R} : |f(x)| \leq \bar{a} \text{ for } \omega\text{-almost every } x \in \mathcal{T}\}. \quad (20)$$

577 Let $\mathcal{M}(\mathcal{T} \times \mathcal{T})$ denote the space of all nonnegative Borel measures on $\mathcal{T} \times \mathcal{T}$ with finite total mass.
 578 Given $\mu, \nu \in \mathcal{M}(\mathcal{T})$, define the set of admissible partial couplings as

$$\Pi_{\leq}(\mu, \nu) := \{\gamma \in \mathcal{M}(\mathcal{T} \times \mathcal{T}) : \gamma_1 \leq \mu, \gamma_2 \leq \nu\}, \quad (21)$$

579 where γ_1 and γ_2 represent the marginals of γ on the first and second coordinates, respectively.

580 For any $\gamma \in \Pi_{\leq}(\mu, \nu)$, let f_1 and f_2 be the Radon–Nikodym derivatives of γ_1 with respect to μ and
 581 γ_2 with respect to ν , respectively. That is, $\gamma_1 = f_1\mu$ and $\gamma_2 = f_2\nu$, with the constraints $0 \leq f_1 \leq 1$
 582 μ -a.e. and $0 \leq f_2 \leq 1$ ν -a.e.

583 Let $w : \mathcal{T} \rightarrow [0, \infty)$ be a b -Lipschitz continuous and nonnegative weight function, defined by

$$w(x) = a_1 d_{\mathcal{T}}(x, x_0) + a_0, \quad (22)$$

584 where $x_0 \in \mathcal{T}$, $a_1 \in [0, b]$, and $a_0 \in [0, \infty)$. Here, $d_{\mathcal{T}}(\cdot, \cdot)$ denotes the tree metric over \mathcal{T} . We use
 585 the entropy function $F : [0, \infty) \rightarrow (0, \infty)$ given by $F(s) = |s - 1|$.

586 Letting $\bar{m} := \min\{\mu(\mathcal{T}), \nu(\mathcal{T})\}$, and fixing $m \in [0, \bar{m}]$, the EPT problem is formulated as

$$\mathcal{W}_m(\mu, \nu) := \inf_{\substack{\gamma \in \Pi_{\leq}(\mu, \nu) \\ \gamma(\mathcal{T} \times \mathcal{T}) = m}} \left[\mathcal{F}_1(\gamma_1 | \mu) + \mathcal{F}_2(\gamma_2 | \nu) + b \int_{\mathcal{T} \times \mathcal{T}} d_{\mathcal{T}}(x, y) \gamma(dx, dy) \right], \quad (23)$$

587 where the regularization terms are defined as the weighted relative entropies

$$\mathcal{F}_1(\gamma_1 | \mu) := \int_{\mathcal{T}} w(x) F(f_1(x)) \mu(dx), \quad \mathcal{F}_2(\gamma_2 | \nu) := \int_{\mathcal{T}} w(x) F(f_2(x)) \nu(dx). \quad (24)$$

588 To handle the mass constraint $\gamma(\mathcal{T} \times \mathcal{T}) = m$, we introduce a Lagrange multiplier $\lambda \in \mathbb{R}$ and instead
589 consider the relaxed objective

$$\text{ET}_{\lambda}(\mu, \nu) := \inf_{\gamma \in \Pi_{\leq}(\mu, \nu)} \left[\mathcal{F}_1(\gamma_1 | \mu) + \mathcal{F}_2(\gamma_2 | \nu) + b \int_{\mathcal{T} \times \mathcal{T}} (d_{\mathcal{T}}(x, y) - \lambda) \gamma(dx, dy) \right]. \quad (25)$$

590 We now expand the entropic terms and define

$$\begin{aligned} \mathcal{C}_{\lambda}(\gamma) := & \int_{\mathcal{T}} w(x) \mu(dx) + \int_{\mathcal{T}} w(x) \nu(dx) - \int_{\mathcal{T}} w(x) \gamma_1(dx) - \int_{\mathcal{T}} w(x) \gamma_2(dx) \\ & + b \int_{\mathcal{T} \times \mathcal{T}} (d_{\mathcal{T}}(x, y) - \lambda) \gamma(dx, dy), \end{aligned} \quad (26)$$

591 so that Equation (25) is equivalent to

$$\text{ET}_{\lambda}(\mu, \nu) = \inf_{\gamma \in \Pi_{\leq}(\mu, \nu)} \mathcal{C}_{\lambda}(\gamma). \quad (27)$$

592 As established in [38, Theorem 3.1, part i)], the solutions to Equation (23) and Equation (27) are
593 related via the identity

$$\mathcal{W}_m(\mu, \nu) = \text{ET}_{\lambda}(\mu, \nu) + \lambda b m. \quad (28)$$

594 Inspired by the construction proposed by Caffarelli and McCann [12], we recast the entropy-
595 regularized partial transport problem in Equation (27) as a classical optimal transport (OT) problem
596 between balanced measures. To achieve this, we augment the original domain \mathcal{T} by introducing an
597 auxiliary point $\hat{s} \notin \mathcal{T}$, and define the extended space $\hat{\mathcal{T}} := \mathcal{T} \cup \{\hat{s}\}$.

598 We then lift the unbalanced measures $\mu, \nu \in \mathcal{M}(\mathcal{T})$ to balanced counterparts supported on $\hat{\mathcal{T}}$:

$$\hat{\mu} := \mu + \nu(\mathcal{T}) \delta_{\hat{s}}, \quad \hat{\nu} := \nu + \mu(\mathcal{T}) \delta_{\hat{s}}, \quad (29)$$

599 where $\delta_{\hat{s}}$ denotes the Dirac measure at point \hat{s} . Next, we define a cost function $\hat{c} : \hat{\mathcal{T}} \times \hat{\mathcal{T}} \rightarrow \mathbb{R}$ that
600 extends the original transport cost:

$$\hat{c}(x, y) := \begin{cases} b [d_{\mathcal{T}}(x, y) - \lambda] & \text{if } x, y \in \mathcal{T}, \\ w(x) & \text{if } x \in \mathcal{T} \text{ and } y = \hat{s}, \\ w(y) & \text{if } y \in \mathcal{T} \text{ and } x = \hat{s}, \\ 0 & \text{if } x = y = \hat{s}. \end{cases} \quad (30)$$

601 Using this extended cost, we formulate the balanced OT problem over $\hat{\mu}$ and $\hat{\nu}$:

$$\text{KT}(\hat{\mu}, \hat{\nu}) := \inf_{\hat{\gamma} \in \Gamma(\hat{\mu}, \hat{\nu})} \int_{\hat{\mathcal{T}} \times \hat{\mathcal{T}}} \hat{c}(x, y) \hat{\gamma}(dx, dy), \quad (31)$$

602 where the set of admissible transport plans $\Gamma(\hat{\mu}, \hat{\nu})$ is given by:

$$\Gamma(\hat{\mu}, \hat{\nu}) := \left\{ \hat{\gamma} \in \mathcal{M}(\hat{\mathcal{T}} \times \hat{\mathcal{T}}) : \hat{\gamma}(U \times \hat{\mathcal{T}}) = \hat{\mu}(U), \hat{\gamma}(\hat{\mathcal{T}} \times U) = \hat{\nu}(U), \forall \text{ Borel set } U \subset \hat{\mathcal{T}} \right\}. \quad (32)$$

603 The connection between the entropy-regularized partial transport formulation ET_{λ} in Equation (27)
604 and the balanced optimal transport problem KT in Equation (31) is established by the following
605 result.

606 **Proposition B.1** (Equivalence of ET_{λ} and KT). *Let $\mu, \nu \in \mathcal{M}(\mathcal{T})$. Then the two formulations
607 coincide:*

$$\text{ET}_{\lambda}(\mu, \nu) = \text{KT}(\hat{\mu}, \hat{\nu}). \quad (33)$$

608 Furthermore, the optimal plans γ for the partial transport problem and $\hat{\gamma}$ for the balanced transport
609 problem are related by:

$$\hat{\gamma} = \gamma + (1 - f_1) \mu \otimes \delta_{\hat{s}} + \delta_{\hat{s}} \otimes (1 - f_2) \nu + \gamma(\mathcal{T} \times \mathcal{T}) \delta_{(\hat{s}, \hat{s})}, \quad (34)$$

610 where f_1 and f_2 are the Radon–Nikodym derivatives of the marginals of γ with respect to μ and ν ,
611 respectively.

The detailed proof is provided in Appendix D.1. Note that KT corresponds to a classical optimal transport problem defined between two balanced measures over the extended space $\hat{\mathcal{T}}$ and governed by the cost function \hat{c} . This allows us to invoke standard OT duality theory, such as [12, Corollary 2.6], to obtain a variational dual formulation for ET_λ , as described below.

Theorem B.2 (Dual Representation of ET_λ). *The dual problem associated with the entropy-regularized partial transport functional $\text{ET}_\lambda(\mu, \nu)$ is given by:*

$$\text{ET}_\lambda(\mu, \nu) = \sup \left\{ \int_{\mathcal{T}} f (d\mu - d\nu) : f \in \mathbb{L} \right\} - \frac{b\lambda}{2} [\mu(\mathcal{T}) + \nu(\mathcal{T})], \quad (35)$$

where the admissible function class \mathbb{L} is defined as

$$\mathbb{L} := \left\{ f \in C(\mathcal{T}) : -w - \frac{b\lambda}{2} \leq f \leq w + \frac{b\lambda}{2}, \quad |f(x) - f(y)| \leq b d_{\mathcal{T}}(x, y) \text{ for all } x, y \in \mathcal{T} \right\}.$$

The proof of Theorem B.2 is deferred to Appendix D.2. To obtain a tractable approximation of the dual problem, we introduce a regularization based on a restricted class of test functions. Let r denote the root of the tree \mathcal{T} , and let ω be the associated length measure on \mathcal{T} . For a fixed parameter $a \in [0, \frac{b\lambda}{2} + w(r)]$, define the function class \mathbb{L}_a to consist of all functions $f : \mathcal{T} \rightarrow \mathbb{R}$ of the form:

$$f(x) = s + \int_{[r, x]} g(y) \omega(dy), \quad (36)$$

where s is a constant satisfying

$$s \in \left[-w(r) - \frac{b\lambda}{2} + a, w(r) + \frac{b\lambda}{2} - a \right], \quad (37)$$

and $g \in L^\infty(\mathcal{T})$ is a bounded function with $\|g\|_{L^\infty(\mathcal{T})} \leq b$.

The a -regularized entropy partial transport is then defined as:

$$\widetilde{\text{ET}}_\lambda^a(\mu, \nu) := \sup_{f \in \mathbb{L}_a} \left\{ \int_{\mathcal{T}} f (d\mu - d\nu) \right\} - \frac{b\lambda}{2} [\mu(\mathcal{T}) + \nu(\mathcal{T})]. \quad (38)$$

This regularized formulation admits an explicit closed-form expression:

Proposition B.3 (Closed-Form Solution for $\widetilde{\text{ET}}_\lambda^a$). *For $\mu, \nu \in \mathcal{M}(\mathcal{T})$, we have:*

$$\begin{aligned} \widetilde{\text{ET}}_\lambda^a(\mu, \nu) &= \int_{\mathcal{T}} |\mu(\Lambda(x)) - \nu(\Lambda(x))| \omega(dx) \\ &\quad - \frac{b\lambda}{2} [\mu(\mathcal{T}) + \nu(\mathcal{T})] + \left(w(r) + \frac{b\lambda}{2} - a \right) |\mu(\mathcal{T}) - \nu(\mathcal{T})|. \end{aligned} \quad (39)$$

The proof is provided in Appendix D.3. We now compare the original entropy transport value ET_λ with its regularized approximation $\widetilde{\text{ET}}_\lambda^a$:

Proposition B.4 (Comparison Bounds between ET_λ and $\widetilde{\text{ET}}_\lambda^a$). *The following inequalities hold:*

$$\text{ET}_\lambda(\mu, \nu) \leq \widetilde{\text{ET}}_\lambda^0(\mu, \nu), \quad (40)$$

and if the condition

$$[4L_{\mathcal{T}} - \lambda]b \leq 2w(r), \quad \text{where } L_{\mathcal{T}} := \max_{x \in \mathcal{T}} \omega([r, x]), \quad (41)$$

is satisfied, then

$$\widetilde{\text{ET}}_\lambda^a(\mu, \nu) \leq \text{ET}_\lambda(\mu, \nu) \quad (42)$$

for all a such that $2bL_{\mathcal{T}} \leq a \leq \frac{b\lambda}{2} + w(r)$.

The proof appears in Appendix D.4. For $0 \leq a < \frac{b\lambda}{2} + w(r)$, we define the following regularized transport cost:

$$d_a(\mu, \nu) := \widetilde{\text{ET}}_\lambda^a(\mu, \nu) + \frac{b\lambda}{2} [\mu(\mathcal{T}) + \nu(\mathcal{T})]. \quad (43)$$

This cost function defines a genuine metric, as shown below:

Proposition B.5 (Metric Structure of d_a). *$(\mathcal{M}(\mathcal{T}), d_a)$ is a complete metric space.*

The proof is presented in Appendix D.5.

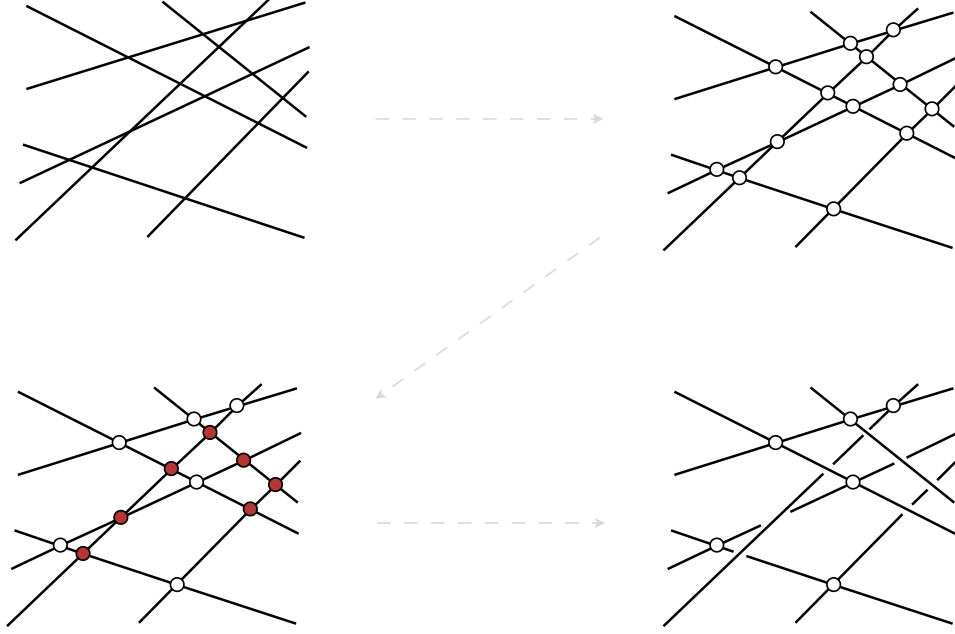


Figure 6: An illustration of the tree system construction is presented in the two-dimensional space \mathbb{R}^2 , though the method readily generalizes to higher dimensions. The process starts with a collection of infinite lines arranged without any inherent structure. All pairwise intersections among these lines are identified (some of which may not be visible in the figure due to the lines’ unbounded nature). A subset of intersections is selected and marked in red to indicate those to be discarded. The remaining intersections are retained to enforce a tree structure on the system—ensuring that any two points lying on the lines are connected by a unique path that passes only through the preserved intersections. These remaining intersections act as the essential nodes that define the tree topology. Once the red (discarded) intersections are removed, the resulting configuration forms the desired tree system.

639 C Background on Tree-Sliced Wasserstein Distance on Euclidean Spaces

640 This section reviews foundational concepts underlying the Tree-Sliced Wasserstein distance defined
 641 over Tree Systems. To ensure completeness, we revisit key definitions and theoretical formulations;
 642 detailed proofs and additional exposition are available in [77, 73].

643 C.1 Tree System

644 **Line.** A line in the Euclidean space \mathbb{R}^d is specified by a tuple $(x, \theta) \in \mathbb{R}^d \times \mathbb{S}^{d-1}$ and is expressed
 645 parametrically as $x + t \cdot \theta$ for $t \in \mathbb{R}$. Throughout, we use $l = (x_l, \theta_l) \in \mathbb{R}^d \times \mathbb{S}^{d-1}$ to denote a line,
 646 where x_l denotes the *source* point and θ_l the *direction* vector.

647 **System of lines.** Given an integer $k \geq 1$, a *system of k lines in \mathbb{R}^d* refers to a collection of k such
 648 lines. The notation $(\mathbb{R}^d \times \mathbb{S}^{d-1})^k$ is abbreviated as \mathbb{L}_k^d , representing the *space of systems of k lines*
 649 *in \mathbb{R}^d* . An element in this space, commonly denoted by \mathcal{L} , corresponds to a specific system of lines.

650 **Tree System.** A system \mathcal{L} is said to be *connected* if the union of all points lying on the constituent
 651 lines forms a connected subset of \mathbb{R}^d . By selectively *removing* certain intersection points between the
 652 lines, one can enforce a tree structure on \mathcal{L} —yielding a *tree system*—in which any two points are
 653 connected by a unique path. An illustration of this construction is provided in Figure 6.

654 **Remark C.1.** The term *tree system* is used because there is a unique path between any two points,
 655 analogous to the definition of trees in graph theory.

656 Beginning with the remaining intersections, we employ the concepts of disjoint union and quotient
 657 topology [28] to construct a tree system by coherently gluing together multiple copies of \mathbb{R} . This

topological framework induces a natural metric, under which the resulting space satisfies the properties of a tree metric space.

Sampling Procedure for Chain-Structured Tree Systems. The space of tree systems is inherently rich and diverse, primarily due to the wide range of possible underlying tree topologies. [77] presents a general framework that accommodates arbitrary tree structures, while placing particular emphasis on a subclass of chain-like trees. The following describes the sampling procedure for generating tree systems with this chain-based architecture:

Step 1. Draw an initial point $x_1 \sim \mu_1$ and a direction $\theta_1 \sim \nu_1$, where μ_1 is a probability measure on \mathbb{R}^d and ν_1 is a measure on the unit sphere \mathbb{S}^{d-1} .

Step i . For each subsequent node, sample $t_i \sim \mu_i$ and $\theta_i \sim \nu_i$, then compute $x_i = x_{i-1} + t_i \cdot \theta_{i-1}$. Here, μ_i is a distribution over \mathbb{R} and ν_i over \mathbb{S}^{d-1} .

All distributions μ_i and ν_i are assumed to be mutually independent. Specifically, we consider the following choices: The initial position distribution μ_1 is supported on a bounded subset of \mathbb{R}^d , such as the uniform distribution over the cube $[-1, 1]^d$, i.e., $\mathcal{U}([-1, 1]^d)$; For $i > 1$, each μ_i is defined on a bounded interval of \mathbb{R} —for example, $\mathcal{U}([-1, 1])$; Finally, each direction θ_i is drawn from a distribution over the unit sphere, e.g., the uniform distribution $\mathcal{U}(\mathbb{S}^{d-1})$. An example of such a tree system is illustrated in Figure 6.

Remark C.2. This generative process induces a probability measure σ on the space \mathbb{T} of all chain-structured tree systems produced via this construction.

C.2 A Variant of Radon Transform for Systems of Lines

Let $L^1(\mathbb{R}^d)$ denote the space of Lebesgue integrable functions on \mathbb{R}^d , equipped with the standard L^1 norm $\|\cdot\|_1$. Consider a configuration of k lines $\mathcal{L} \in \mathbb{L}_k^d$. A real-valued function f defined on the domain $\tilde{\mathcal{L}}$ consists of all points of \mathcal{L} , is said to be *integrable over the line system* if the following condition holds:

$$\|f\|_{\mathcal{L}} := \sum_{l \in \mathcal{L}} \int_{\mathbb{R}} |f(t_x, l)| dt_x < \infty. \quad (44)$$

The set of such functions is denoted by $L^1(\mathcal{L})$, representing the space of Lebesgue integrable functions over the line system \mathcal{L} . Recall the standard $(k-1)$ -simplex:

$$\Delta_{k-1} = \left\{ (a_l)_{l \in \mathcal{L}} \in \mathbb{R}^k \mid a_l \geq 0, \sum_{l \in \mathcal{L}} a_l = 1 \right\}. \quad (45)$$

Define the space $\mathcal{C}(\mathbb{R}^d \times \mathbb{L}_k^d, \Delta_{k-1})$ as the set of continuous maps from $\mathbb{R}^d \times \mathbb{L}_k^d$ to Δ_{k-1} , referred to as *splitting maps*. Given a line system $\mathcal{L} \in \mathbb{L}_k^d$ and a splitting map $\alpha \in \mathcal{C}(\mathbb{R}^d \times \mathbb{L}_k^d, \Delta_{k-1})$, we define a linear operator that projects a function $f \in L^1(\mathbb{R}^d)$ to the line system \mathcal{L} as follows:

$$\mathcal{R}_{\mathcal{L}}^{\alpha} f: \tilde{\mathcal{L}} \longrightarrow \mathbb{R} \quad (46)$$

$$(x, l) \longmapsto \int_{\mathbb{R}^d} f(y) \cdot \alpha(y, \mathcal{L})_l \cdot \delta(t_x - \langle y - x_l, \theta_l \rangle) dy, \quad (47)$$

where δ denotes the Dirac delta function in one dimension, and (x_l, θ_l) encodes the location and direction of line l . It can be shown that $\mathcal{R}_{\mathcal{L}}^{\alpha} f$ belongs to $L^1(\mathcal{L})$ for any $f \in L^1(\mathbb{R}^d)$, and furthermore satisfies the inequality $\|\mathcal{R}_{\mathcal{L}}^{\alpha} f\|_{\mathcal{L}} \leq \|f\|_1$. Hence, the operator $\mathcal{R}_{\mathcal{L}}^{\alpha}: L^1(\mathbb{R}^d) \rightarrow L^1(\mathcal{L})$ is well-defined. These properties are proven in [73]. Extending this to all line systems, we define the *Radon transform on Systems of Lines* as follows. For a fixed splitting map $\alpha \in \mathcal{C}(\mathbb{R}^d \times \mathbb{L}_k^d, \Delta_{k-1})$, define:

$$\mathcal{R}^{\alpha}: L^1(\mathbb{R}^d) \longrightarrow \prod_{\mathcal{L} \in \mathbb{L}_k^d} L^1(\mathcal{L}) \quad (48)$$

$$f \longmapsto (\mathcal{R}_{\mathcal{L}}^{\alpha} f)_{\mathcal{L} \in \mathbb{L}_k^d}. \quad (49)$$

If the splitting map α is invariant under the Euclidean group $E(d)$ —the group of all isometries of \mathbb{R}^d —then the operator \mathcal{R}^{α} is injective.

694 C.3 Tree-Sliced Wasserstein Distance for Probability Measures on Euclidean Spaces

695 Let $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$ be probability measures. For a tree-structured system of lines $\mathcal{L} \in \mathbb{T}$ and an
 696 $E(d)$ -invariant splitting map $\alpha \in \mathcal{C}(\mathbb{R}^d \times \mathbb{L}_k^d, \Delta_{k-1})$, the transform $\mathcal{R}_{\mathcal{L}}^\alpha$ pushes forward μ and ν to
 697 corresponding measures $\mathcal{R}_{\mathcal{L}}^\alpha \mu$ and $\mathcal{R}_{\mathcal{L}}^\alpha \nu$ on \mathcal{L} . Since each $\mathcal{L} \in \mathbb{T}$ is equipped with a tree metric $d_{\mathcal{L}}$,
 698 the 1-Wasserstein distance $W_{d_{\mathcal{L}},1}$ between the transformed measures can be computed. This leads to
 699 the following definition of the *Distance-based Tree-Sliced Wasserstein* (Db-TSW) [73] distance:

$$\text{Db-TSW}(\mu, \nu) := \int_{\mathbb{T}} W_{d_{\mathcal{L}},1}(\mathcal{R}_{\mathcal{L}}^\alpha \mu, \mathcal{R}_{\mathcal{L}}^\alpha \nu) d\sigma(\mathcal{L}), \quad (50)$$

700 where σ is a probability measure over the space of tree systems \mathbb{T} . It is important to note that the
 701 value of Db-TSW depends on the choice of the tree system space \mathbb{T} , the sampling distribution σ ,
 702 and the specific $E(d)$ -invariant splitting map α , although these dependencies are omitted from the
 703 notation for brevity. The resulting Db-TSW defines an $E(d)$ -invariant metric on $\mathcal{P}(\mathbb{R}^d)$.

704 **Remark C.3.** As established in [73], if the tree systems consist solely of a single line, the Db-TSW
 705 distance reduces exactly to the classical Sliced Wasserstein (SW) distance on \mathbb{R}^d .

706 **Constructing $E(d)$ -Invariant Splitting Maps.** The Euclidean group $E(d)$ consists of all transforma-
 707 tions of \mathbb{R}^d that preserve pairwise Euclidean distances. As such, this invariance extends not only to
 708 distances between points but also to the shortest distance from a point to a line. Given a point $x \in \mathbb{R}^d$
 709 and a system of lines $\mathcal{L} \in \mathbb{L}_k^d$, define the distance from x to a line $l \in \mathcal{L}$ by:

$$d(x, \mathcal{L})_l = \inf_{y \in l} \|x - y\|_2. \quad (51)$$

710 This quantity is preserved under the action of $E(d)$, meaning that any function constructed solely
 711 from the collection $\{d(x, \mathcal{L})_l\}_{l \in \mathcal{L}}$ will inherit $E(d)$ -invariance.

712 Based on this observation, invariant splitting maps is constructed by applying a post-processing
 713 function $\beta: \mathbb{R}^k \rightarrow \Delta_{k-1}$ to the vector of distances. The resulting splitting map,

$$\alpha(x, \mathcal{L})_l = \beta(\{d(x, \mathcal{L})_l\}_{l \in \mathcal{L}}), \quad (52)$$

714 is guaranteed to be $E(d)$ -invariant for any choice of continuous β . Empirically, effective performance
 715 in applications is achieved when β is taken to be the softmax function with a tunable scaling parameter
 716 $\xi > 0$. This yields the practical definition:

$$\alpha(x, \mathcal{L})_l = \text{softmax}(\{\xi \cdot d(x, \mathcal{L})_l\}_{l \in \mathcal{L}}), \quad (53)$$

717 which distributes weights across lines in \mathcal{L} according to their proximity to x , while respecting the
 718 geometric symmetries of the Euclidean space.

719 D Theoretical Proofs

720 To ensure completeness, we provide full derivations of the result, closely following the methodology
 721 of [38].

722 D.1 Proof for Proposition B.1

723 To ensure completeness, we provide full derivations of the result, closely following the methodology
 724 of [38].

725 *Proof.* We begin by proving the inequality $\text{KT}(\hat{\mu}, \hat{\nu}) \leq \text{ET}_\lambda(\mu, \nu)$. Let $\gamma \in \Pi_{\leq}(\mu, \nu)$ be any
 726 admissible partial transport plan, and define $\hat{\gamma}$ according to the expression in Equation (34). By
 727 construction, $\hat{\gamma} \in \Gamma(\hat{\mu}, \hat{\nu})$. Then, evaluating the cost of $\hat{\gamma}$ under the extended transport objective
 728 yields:

$$\begin{aligned} \text{KT}(\hat{\mu}, \hat{\nu}) &\leq \int_{\hat{\mathcal{T}} \times \hat{\mathcal{T}}} \hat{c}(x, y) \hat{\gamma}(dx, dy) \\ &= b \int_{\mathcal{T} \times \mathcal{T}} [d_{\mathcal{T}}(x, y) - \lambda] \gamma(dx, dy) \\ &\quad + \int_{\mathcal{T}} w(x) [1 - f_1(x)] \mu(dx) + \int_{\mathcal{T}} w(x) [1 - f_2(x)] \nu(dx). \end{aligned} \quad (54)$$

729 Taking the infimum over all $\gamma \in \Pi_{\leq}(\mu, \nu)$ on the right-hand side implies:

$$\text{KT}(\hat{\mu}, \hat{\nu}) \leq \text{ET}_{\lambda}(\mu, \nu). \quad (55)$$

730 We now establish the reverse inequality, i.e., $\text{KT}(\hat{\mu}, \hat{\nu}) \geq \text{ET}_{\lambda}(\mu, \nu)$. Let $\hat{\gamma} \in \Gamma(\hat{\mu}, \hat{\nu})$ be any feasible
 731 coupling in the balanced OT problem, and let γ be its restriction to $\mathcal{T} \times \mathcal{T}$. Then, by [38, Lemma
 732 3.2], we have $\gamma \in \Pi_{\leq}(\mu, \nu)$ and the decomposition in Equation (34) holds. We now compute the
 733 total cost of $\hat{\gamma}$ under \hat{c} :

$$\begin{aligned} \int_{\hat{\mathcal{T}} \times \hat{\mathcal{T}}} \hat{c}(x, y) \hat{\gamma}(dx, dy) &= b \int_{\mathcal{T} \times \mathcal{T}} [d_{\mathcal{T}}(x, y) - \lambda] \gamma(dx, dy) \\ &\quad + \int_{\mathcal{T}} w(x) [1 - f_1(x)] \mu(dx) + \int_{\mathcal{T}} w(x) [1 - f_2(x)] \nu(dx) \\ &\geq \text{ET}_{\lambda}(\mu, \nu). \end{aligned} \quad (56)$$

734 Taking the infimum over all admissible $\hat{\gamma} \in \Gamma(\hat{\mu}, \hat{\nu})$ yields the desired inequality:

$$\text{KT}(\hat{\mu}, \hat{\nu}) \geq \text{ET}_{\lambda}(\mu, \nu). \quad (57)$$

735 Combining both bounds, we conclude the equivalence:

$$\text{KT}(\hat{\mu}, \hat{\nu}) = \text{ET}_{\lambda}(\mu, \nu). \quad (58)$$

736 The correspondence between optimal couplings γ and $\hat{\gamma}$ follows directly from the construction and
 737 identities established above. \square

738 D.2 Proof for Theorem B.2

739 To ensure completeness, we provide full derivations of the result, closely following the methodology
 740 of [38].

741 *Proof.* We begin by establishing the intermediate result:

$$\text{ET}_{\lambda}(\mu, \nu) = \sup_{(u, v) \in \mathcal{K}} \left[\int_{\mathcal{T}} u(x) \mu(dx) + \int_{\mathcal{T}} v(x) \nu(dx) \right], \quad (59)$$

742 where the admissible set \mathcal{K} is defined as

$$\begin{aligned} \mathcal{K} := \left\{ (u, v) \in L^1(\mu) \times L^1(\nu) \mid \right. & u(x) \leq w(x), \quad \forall x \in \mathcal{T}, \\ & -b\lambda + \inf_{x \in \mathcal{T}} [b d_{\mathcal{T}}(x, y) - w(x)] \leq v(y) \leq w(y), \quad \forall y \in \mathcal{T}, \\ & \left. u(x) + v(y) \leq b[d_{\mathcal{T}}(x, y) - \lambda], \quad \forall x, y \in \mathcal{T} \right\}. \end{aligned}$$

743 This identity follows from the dual representation of $\text{KT}(\hat{\mu}, \hat{\nu})$ via Proposition B.1 and [12, Corollary
 744 2.6], which yields:

$$\begin{aligned} \text{ET}_{\lambda}(\mu, \nu) &= \sup_{\substack{\hat{u} \in L^1(\hat{\mu}), \hat{v} \in L^1(\hat{\nu}) \\ \hat{u}(x) + \hat{v}(y) \leq \hat{c}(x, y)}} \left[\int_{\hat{\mathcal{T}}} \hat{u}(x) \hat{\mu}(dx) + \int_{\hat{\mathcal{T}}} \hat{v}(y) \hat{\nu}(dy) \right] \\ &=: I. \end{aligned} \quad (60)$$

745 We aim to show that this supremum I coincides with

$$J := \sup_{(u, v) \in \mathcal{K}} \left[\int_{\mathcal{T}} u(x) \mu(dx) + \int_{\mathcal{T}} v(x) \nu(dx) \right]. \quad (61)$$

746 To show $I \geq J$, let $(u, v) \in \mathcal{K}$. Extend these functions to $\hat{\mathcal{T}}$ by setting:

$$\hat{u}(x) := \begin{cases} u(x) & \text{if } x \in \mathcal{T}, \\ 0 & \text{if } x = \hat{s}, \end{cases} \quad \hat{v}(x) := \begin{cases} v(x) & \text{if } x \in \mathcal{T}, \\ 0 & \text{if } x = \hat{s}. \end{cases}$$

747 Since $(u, v) \in \mathcal{K}$, it follows directly from the definition of \hat{c} that $\hat{u}(x) + \hat{v}(y) \leq \hat{c}(x, y)$ for all
 748 $x, y \in \hat{\mathcal{T}}$. Consequently:

$$\begin{aligned} I &\geq \int_{\hat{\mathcal{T}}} \hat{u}(x) \hat{\mu}(dx) + \int_{\hat{\mathcal{T}}} \hat{v}(x) \hat{\nu}(dx) \\ &= \int_{\mathcal{T}} u(x) \mu(dx) + \int_{\mathcal{T}} v(x) \nu(dx), \end{aligned} \quad (62)$$

749 which implies $I \geq J$.

750 To prove the reverse inequality $I \leq J$, let (\hat{u}, \hat{v}) be a maximizer for I . Without loss of generality,
 751 we can normalize $\hat{u}(\hat{s}) = 0$ by observing that replacing (\hat{u}, \hat{v}) with $(\hat{u} - \hat{u}(\hat{s}), \hat{v} + \hat{u}(\hat{s}))$ preserves
 752 admissibility and the objective value. Moreover, define:

$$v(y) := \inf_{x \in \hat{\mathcal{T}}} [\hat{c}(x, y) - \hat{u}(x)] \quad \forall y \in \hat{\mathcal{T}}. \quad (63)$$

753 Then $\hat{v}(y) \leq v(y)$, and (\hat{u}, v) remains admissible and achieves the same supremum, so we may
 754 further assume $\hat{v}(y) = \inf_x [\hat{c}(x, y) - \hat{u}(x)]$ and $\hat{u}(\hat{s}) = 0$. In particular,

$$\hat{v}(\hat{s}) = \inf_{x \in \hat{\mathcal{T}}} [\hat{c}(x, \hat{s}) - \hat{u}(x)]. \quad (64)$$

755 To proceed, we define $w(\hat{s}) := 0$ and consider two cases based on the structure of \hat{u} and \hat{v} .

756 *Case 1.* Suppose that

$$\inf_{x \in \hat{\mathcal{T}}} [w(x) - \hat{u}(x)] \geq 0. \quad (65)$$

757 In this case, observe that $\hat{u}(\hat{s}) = 0$ by assumption. Since

$$\hat{c}(\hat{s}, \hat{s}) - \hat{u}(\hat{s}) = 0 \quad \text{and} \quad \inf_{x \in \hat{\mathcal{T}}} [\hat{c}(x, \hat{s}) - \hat{u}(x)] = \inf_{x \in \hat{\mathcal{T}}} [w(x) - \hat{u}(x)] \geq 0,$$

758 we conclude that

$$\hat{v}(\hat{s}) = \inf_{x \in \hat{\mathcal{T}}} [\hat{c}(x, \hat{s}) - \hat{u}(x)] = 0. \quad (66)$$

759 Next, for all $y \in \hat{\mathcal{T}}$, we bound $\hat{v}(y)$ from above:

$$\hat{v}(y) = \inf_{x \in \hat{\mathcal{T}}} [\hat{c}(x, y) - \hat{u}(x)] \leq \hat{c}(\hat{s}, y) - \hat{u}(\hat{s}) = w(y), \quad (67)$$

760 where we have used $\hat{u}(\hat{s}) = 0$.

761 To lower-bound $\hat{v}(y)$ for $y \in \mathcal{T}$, note that $\hat{u}(x) \leq w(x)$ for all $x \in \mathcal{T}$, and $w(\hat{s}) = 0$. Therefore,

$$\begin{aligned} \hat{v}(y) &= \inf_{x \in \hat{\mathcal{T}}} [\hat{c}(x, y) - \hat{u}(x)] \\ &\geq \inf_{x \in \hat{\mathcal{T}}} [\hat{c}(x, y) - w(x)] \\ &= \inf_{x \in \mathcal{T}} [b(d_{\mathcal{T}}(x, y) - \lambda) - w(x)] \\ &= -b\lambda + \inf_{x \in \mathcal{T}} [b d_{\mathcal{T}}(x, y) - w(x)]. \end{aligned} \quad (68)$$

762 Combining both bounds, we find that $\hat{v}(y)$ satisfies all constraints in the definition of \mathcal{K} , and $\hat{u} \leq w$
 763 holds by assumption. Hence, $(\hat{u}, \hat{v}) \in \mathcal{K}$. We now compute the dual objective:

$$\begin{aligned} I &= \int_{\hat{\mathcal{T}}} \hat{u}(x) \hat{\mu}(dx) + \int_{\hat{\mathcal{T}}} \hat{v}(x) \hat{\nu}(dx) \\ &= \int_{\mathcal{T}} \hat{u}(x) \mu(dx) + \int_{\mathcal{T}} \hat{v}(x) \nu(dx) + \hat{v}(\hat{s}) \mu(\mathcal{T}) \\ &= \int_{\mathcal{T}} \hat{u}(x) \mu(dx) + \int_{\mathcal{T}} \hat{v}(x) \nu(dx) \\ &\leq J. \end{aligned} \quad (69)$$

764 Thus, under this case, the supremum I is bounded above by J , completing the proof for Case 1.

765 *Case 2.* Suppose now that

$$\inf_{x \in \mathcal{T}} [w(x) - \hat{u}(x)] < 0. \quad (71)$$

766 As in Case 1, we deduce that

$$\hat{v}(\hat{s}) = \inf_{x \in \mathcal{T}} [w(x) - \hat{u}(x)] < 0, \quad (72)$$

767 and the dual objective becomes

$$I = \int_{\mathcal{T}} \hat{u}(x) \mu(dx) + \int_{\mathcal{T}} \hat{v}(x) \nu(dx) + \mu(\mathcal{T}) \cdot \inf_{\mathcal{T}} [w - \hat{u}]. \quad (73)$$

768 Define a truncated version of \hat{u} by setting:

$$\tilde{u}(x) := \min\{\hat{u}(x), w(x)\}. \quad (74)$$

769 This ensures that $\tilde{u}(x) \leq w(x)$ and, since $\hat{u}(\hat{s}) = 0$, we also have $\tilde{u}(\hat{s}) = 0$. Furthermore, for all
770 $x, y \in \hat{\mathcal{T}}$,

$$\tilde{u}(x) + \hat{v}(y) \leq \hat{c}(x, y), \quad (75)$$

771 due to the pointwise minimum structure of \tilde{u} and the feasibility of (\hat{u}, \hat{v}) .

772 Since $\inf_{x \in \mathcal{T}} [w(x) - \hat{u}(x)] < 0$, there exists $x_0 \in \mathcal{T}$ such that $\hat{u}(x_0) > w(x_0)$. Thus, at x_0 , we
773 have $\tilde{u}(x_0) = w(x_0)$ and therefore

$$\inf_{\mathcal{T}} [w - \tilde{u}] \leq 0. \quad (76)$$

774 On the other hand, since $\tilde{u}(x) \leq w(x)$ everywhere, it follows that

$$\inf_{\mathcal{T}} [w - \tilde{u}] = 0. \quad (77)$$

775 We now rewrite the first two terms in Equation (73) as:

$$\begin{aligned} \int_{\mathcal{T}} \hat{u}(x) \mu(dx) + \mu(\mathcal{T}) \cdot \inf_{\mathcal{T}} [w - \hat{u}] &= \int_{\mathcal{T}} \tilde{u}(x) \mu(dx) + \int_{\{x: \hat{u}(x) > w(x)\}} [\hat{u}(x) - w(x)] \mu(dx) \\ &\quad + \mu(\mathcal{T}) \cdot \inf_{\mathcal{T}} [w - \hat{u}] \\ &\leq \int_{\mathcal{T}} \tilde{u}(x) \mu(dx). \end{aligned} \quad (78)$$

776 Substituting this into Equation (73), we obtain the upper bound:

$$I \leq \int_{\mathcal{T}} \tilde{u}(x) \mu(dx) + \int_{\mathcal{T}} \hat{v}(x) \nu(dx). \quad (79)$$

777 We now define a new function $\tilde{v} : \mathcal{T} \rightarrow \mathbb{R}$ by

$$\tilde{v}(y) := \inf_{x \in \hat{\mathcal{T}}} [\hat{c}(x, y) - \tilde{u}(x)]. \quad (80)$$

778 By construction, $\tilde{v}(y) \geq \hat{v}(y)$ and for all $y \in \mathcal{T}$,

$$\tilde{v}(y) \leq \hat{c}(\hat{s}, y) - \tilde{u}(\hat{s}) = w(y). \quad (81)$$

779 Furthermore, using $\tilde{u}(x) \leq w(x)$ and the form of \hat{c} , we obtain a lower bound:

$$\begin{aligned} \tilde{v}(y) &= \inf_{x \in \hat{\mathcal{T}}} [\hat{c}(x, y) - \tilde{u}(x)] \\ &\geq \inf_{x \in \mathcal{T}} [b(d_{\mathcal{T}}(x, y) - \lambda) - w(x)] \\ &= -b\lambda + \inf_{x \in \mathcal{T}} [b d_{\mathcal{T}}(x, y) - w(x)]. \end{aligned} \quad (82)$$

780 Combining these, we find that $(\tilde{u}, \tilde{v}) \in \mathcal{K}$. Hence,

$$I \leq \int_{\mathcal{T}} \tilde{u}(x) \mu(dx) + \int_{\mathcal{T}} \tilde{v}(x) \nu(dx) \leq J. \quad (83)$$

781 This completes the analysis for Case 2 and thus confirms the desired equality:

$$\text{ET}_{\lambda}(\mu, \nu) = \sup_{(u, v) \in \mathcal{K}} \left[\int_{\mathcal{T}} u(x) \mu(dx) + \int_{\mathcal{T}} v(x) \nu(dx) \right], \quad (84)$$

782 where

$$\mathcal{K} := \left\{ (u, v) : u \leq w, \quad -b\lambda + \inf_{x \in \mathcal{T}} [b d_{\mathcal{T}}(x, y) - w(x)] \leq v(y) \leq w(y), \right. \\ \left. u(x) + v(y) \leq b(d_{\mathcal{T}}(x, y) - \lambda) \right\}. \quad (85)$$

783 We are now ready to complete the proof of the theorem. Since the weight function w is b -Lipschitz, it
784 satisfies the following inequality for all $x \in \mathcal{T}$:

$$-w(x) \leq \inf_{y \in \mathcal{T}} [b d_{\mathcal{T}}(x, y) - w(y)]. \quad (86)$$

785 Let $(u, v) \in \mathcal{K}$ be arbitrary. Define the following sequence of dual potentials via infimal convolutions:

$$v^*(x) := \inf_{y \in \mathcal{T}} \{b[d_{\mathcal{T}}(x, y) - \lambda] - v(y)\} = -b\lambda + \inf_{y \in \mathcal{T}} [b d_{\mathcal{T}}(x, y) - v(y)] \geq u(x), \quad (87)$$

$$v^{**}(y) := \inf_{x \in \mathcal{T}} \{b[d_{\mathcal{T}}(x, y) - \lambda] - v^*(x)\} = -b\lambda + \inf_{x \in \mathcal{T}} [b d_{\mathcal{T}}(x, y) - v^*(x)] \geq v(y). \quad (88)$$

786 Now, observe that the lower and upper bounds for v imply that

$$-b\lambda + \inf_{x \in \mathcal{T}} [b d_{\mathcal{T}}(x, y) - w(x)] \leq v(y) \leq w(y).$$

787 Using this together with Equation (86), we can derive pointwise bounds on v^* for any $x \in \mathcal{T}$:

$$v^*(x) \leq -b\lambda - v(x) \leq -\inf_{y \in \mathcal{T}} [b d_{\mathcal{T}}(x, y) - w(y)] \leq w(x), \quad (89)$$

$$v^*(x) \geq -b\lambda + \inf_{y \in \mathcal{T}} [b d_{\mathcal{T}}(x, y) - w(y)] \geq -b\lambda - w(x). \quad (90)$$

788 We now show that v^* is b -Lipschitz. Let $x_1, x_2 \in \mathcal{T}$ and fix an arbitrary $\varepsilon > 0$. By the definition of
789 infimum, there exists $y_1 \in \mathcal{T}$ such that

$$b d_{\mathcal{T}}(x_1, y_1) - v(y_1) < v^*(x_1) + b\lambda + \varepsilon.$$

790 Then,

$$v^*(x_2) - v^*(x_1) \leq b d_{\mathcal{T}}(x_2, y_1) - v(y_1) - [b d_{\mathcal{T}}(x_1, y_1) - v(y_1)] + \varepsilon \\ = b [d_{\mathcal{T}}(x_2, y_1) - d_{\mathcal{T}}(x_1, y_1)] + \varepsilon \leq b d_{\mathcal{T}}(x_1, x_2) + \varepsilon. \quad (91)$$

791 Since this holds for all $\varepsilon > 0$, we conclude that

$$v^*(x_2) - v^*(x_1) \leq b d_{\mathcal{T}}(x_1, x_2). \quad (92)$$

792 By symmetry, the reverse inequality also holds, so

$$|v^*(x_1) - v^*(x_2)| \leq b d_{\mathcal{T}}(x_1, x_2), \quad (93)$$

793 which confirms that v^* is b -Lipschitz.

794 Thus, v^* belongs to the following class of functions:

$$\mathbb{L}' := \{f \in C(\mathcal{T}) : -b\lambda - w(x) \leq f(x) \leq w(x), \quad |f(x) - f(y)| \leq b d_{\mathcal{T}}(x, y)\}. \quad (94)$$

795 This concludes the key regularity properties needed for the dual formulation.

796 We now establish the identity $v^{**} = -b\lambda - v^*$. To begin, note from the definition that:

$$v^{**}(y) = \inf_{x \in \mathcal{T}} [b(d_{\mathcal{T}}(x, y) - \lambda) - v^*(x)] \leq -b\lambda - v^*(y). \quad (95)$$

797 On the other hand, since v^* is b -Lipschitz, we have for all $x \in \mathcal{T}$:

$$-v^*(y) \leq b d_{\mathcal{T}}(x, y) - v^*(x),$$

798 which implies

$$-b\lambda - v^*(y) \leq \inf_{x \in \mathcal{T}} [b(d_{\mathcal{T}}(x, y) - \lambda) - v^*(x)] = v^{**}(y). \quad (96)$$

799 Combining both bounds, we conclude that

$$v^{**}(y) = -b\lambda - v^*(y). \quad (97)$$

800 Using this identity, we now bound the dual objective for any $(u, v) \in \mathcal{K}$:

$$\begin{aligned} \int_{\mathcal{T}} u(x) \mu(dx) + \int_{\mathcal{T}} v(x) \nu(dx) &\leq \int_{\mathcal{T}} v^*(x) \mu(dx) + \int_{\mathcal{T}} v^{**}(x) \nu(dx) \\ &= \int_{\mathcal{T}} v^*(x) \mu(dx) - \int_{\mathcal{T}} v^*(x) \nu(dx) - b\lambda \nu(\mathcal{T}) \\ &= -b\lambda \nu(\mathcal{T}) + \int_{\mathcal{T}} v^*(x) (d\mu - d\nu). \end{aligned} \quad (98)$$

801 Since $v^* \in \mathbb{L}'$ as shown earlier, we conclude:

$$\int_{\mathcal{T}} u(x) \mu(dx) + \int_{\mathcal{T}} v(x) \nu(dx) \leq -b\lambda \nu(\mathcal{T}) + \sup_{f \in \mathbb{L}'} \int_{\mathcal{T}} f(x) (d\mu - d\nu). \quad (99)$$

802 Using the variational characterization of $\text{ET}_{\lambda}(\mu, \nu)$ (proved earlier), we deduce the upper bound:

$$\text{ET}_{\lambda}(\mu, \nu) \leq -b\lambda \nu(\mathcal{T}) + \sup_{f \in \mathbb{L}'} \int_{\mathcal{T}} f(x) (d\mu - d\nu). \quad (100)$$

803 To prove the reverse inequality, let $f \in \mathbb{L}'$ and define:

$$u := f, \quad v := -b\lambda - f.$$

804 Then:

$$u(x) \leq w(x), \quad v(x) \leq -b\lambda - (-b\lambda - w(x)) = w(x),$$

805 and

$$\begin{aligned} v(x) &= -b\lambda - f(x) \geq -b\lambda - w(x) \\ &\geq -b\lambda + \inf_{y \in \mathcal{T}} [b d_{\mathcal{T}}(x, y) - w(y)]. \end{aligned} \quad (101)$$

806 Moreover, the b -Lipschitz property of f yields:

$$u(x) + v(y) = f(x) - f(y) - b\lambda \leq b(d_{\mathcal{T}}(x, y) - \lambda), \quad (102)$$

807 which confirms that $(u, v) \in \mathcal{K}$. Applying the variational formula for ET_{λ} , we obtain:

$$-b\lambda \nu(\mathcal{T}) + \int_{\mathcal{T}} f(x) (d\mu - d\nu) = \int_{\mathcal{T}} u(x) \mu(dx) + \int_{\mathcal{T}} v(x) \nu(dx) \leq \text{ET}_{\lambda}(\mu, \nu). \quad (103)$$

808 Since this holds for all $f \in \mathbb{L}'$, we deduce:

$$-b\lambda \nu(\mathcal{T}) + \sup_{f \in \mathbb{L}'} \int_{\mathcal{T}} f(x) (d\mu - d\nu) \leq \text{ET}_{\lambda}(\mu, \nu). \quad (104)$$

809 Putting both directions together, we conclude:

$$\text{ET}_{\lambda}(\mu, \nu) = -b\lambda \nu(\mathcal{T}) + \sup_{f \in \mathbb{L}'} \int_{\mathcal{T}} f(x) (d\mu - d\nu). \quad (105)$$

810 To recover the symmetric form in Theorem B.2, let $f = \tilde{f} - \frac{b\lambda}{2}$. Then, $f \in \mathbb{L}'$ if and only if $\tilde{f} \in \mathbb{L}$.
811 Furthermore:

$$\int_{\mathcal{T}} f(x) (d\mu - d\nu) = \int_{\mathcal{T}} \left(\tilde{f}(x) - \frac{b\lambda}{2} \right) (d\mu - d\nu) = \int_{\mathcal{T}} \tilde{f}(x) (d\mu - d\nu) - \frac{b\lambda}{2} [\mu(\mathcal{T}) - \nu(\mathcal{T})]. \quad (106)$$

812 Substituting into Equation (105), we obtain the final expression:

$$\text{ET}_{\lambda}(\mu, \nu) = \sup_{f \in \mathbb{L}} \int_{\mathcal{T}} f(x) (d\mu - d\nu) - \frac{b\lambda}{2} [\mu(\mathcal{T}) + \nu(\mathcal{T})], \quad (107)$$

813 which completes the proof. \square

814 D.3 Proof for Proposition B.3

815 To ensure completeness, we provide full derivations of the result, closely following the methodology
816 of [38].

817 *Proof.* We begin by expanding the definition of the regularized entropy transport:

$$\begin{aligned} \widetilde{\text{ET}}_{\lambda}^a(\mu, \nu) &= -\frac{b\lambda}{2} [\mu(\mathcal{T}) + \nu(\mathcal{T})] \\ &\quad + \sup \left\{ s \cdot [\mu(\mathcal{T}) - \nu(\mathcal{T})] : s \in \left[-\frac{b\lambda}{2} - w(r) + a, w(r) + \frac{b\lambda}{2} - a \right] \right\} \\ &\quad + \sup \left\{ \int_{\mathcal{T}} \left(\int_{[r,x]} g(y) \omega(dy) \right) (\mu - \nu)(dx) : \|g\|_{L^{\infty}(\mathcal{T})} \leq b \right\}. \end{aligned} \quad (108)$$

818 We now evaluate each supremum separately:

819 - The first supremum corresponds to maximizing a linear function over a symmetric interval. Therefore,
820 it evaluates to

$$\left[w(r) + \frac{b\lambda}{2} - a \right] \cdot |\mu(\mathcal{T}) - \nu(\mathcal{T})|. \quad (109)$$

821 - The second supremum is equivalent to the dual representation of a Lipschitz-type transport energy
822 over tree-structured domains. As established in [20, pp. 575–576], we have:

$$\sup \left\{ \int_{\mathcal{T}} \left(\int_{[r,x]} g(y) \omega(dy) \right) (\mu - \nu)(dx) : \|g\|_{L^{\infty}(\mathcal{T})} \leq b \right\} = \int_{\mathcal{T}} |\mu(\Lambda(x)) - \nu(\Lambda(x))| \omega(dx). \quad (110)$$

823 Combining both components, we obtain the closed-form expression:

$$\begin{aligned} \widetilde{\text{ET}}_{\lambda}^a(\mu, \nu) &= \int_{\mathcal{T}} |\mu(\Lambda(x)) - \nu(\Lambda(x))| \omega(dx) - \frac{b\lambda}{2} [\mu(\mathcal{T}) + \nu(\mathcal{T})] \\ &\quad + \left[w(r) + \frac{b\lambda}{2} - a \right] \cdot |\mu(\mathcal{T}) - \nu(\mathcal{T})|. \end{aligned} \quad (111)$$

824 This concludes the proof. \square

825 D.4 Proof for Proposition B.4

826 To ensure completeness, we provide full derivations of the result, closely following the methodology
827 of [38].

828 *Proof.* We begin with the upper bound $\text{ET}_{\lambda}(\mu, \nu) \leq \widetilde{\text{ET}}_{\lambda}^0(\mu, \nu)$. This follows directly from the
829 inclusion $\mathbb{L} \subset \mathbb{L}_0$ and the dual representation of ET_{λ} established in Theorem B.2.

830 Next, consider a satisfying

$$2bL(\mathcal{T}) \leq a \leq \frac{b\lambda}{2} + w(r). \quad (112)$$

831 We will show that under this condition, the inclusion $\mathbb{L}_a \subset \mathbb{L}$ holds. Then, by Theorem B.2, it follows
832 that

$$\widetilde{\text{ET}}_\lambda^a(\mu, \nu) \leq \text{ET}_\lambda(\mu, \nu). \quad (113)$$

833 To prove $\mathbb{L}_a \subset \mathbb{L}$, we need to show that any function $f \in \mathbb{L}_a$ satisfies

$$-w(x) - \frac{b\lambda}{2} \leq f(x) \leq w(x) + \frac{b\lambda}{2}, \quad \forall x \in \mathcal{T}. \quad (114)$$

834 Let $f \in \mathbb{L}_a$. Then by definition,

$$f(x) = s + \int_{[r,x]} g(y) \omega(dy), \quad (115)$$

835 where $s \in [-w(r) - \frac{b\lambda}{2} + a, w(r) + \frac{b\lambda}{2} - a]$ and $\|g\|_{L^\infty(\mathcal{T})} \leq b$. Using this, we bound $f(x)$ from
836 above:

$$\begin{aligned} f(x) &\leq s + \|g\|_{L^\infty(\mathcal{T})} \cdot \omega([r, x]) \\ &\leq w(r) + \frac{b\lambda}{2} - a + bL(\mathcal{T}) \\ &\leq w(x) + \frac{b\lambda}{2} - a + 2bL(\mathcal{T}) \\ &\leq w(x) + \frac{b\lambda}{2}. \end{aligned} \quad (116)$$

837 For the lower bound, we have:

$$\begin{aligned} f(x) &\geq s - \|g\|_{L^\infty(\mathcal{T})} \cdot \omega([r, x]) \\ &\geq -w(r) - \frac{b\lambda}{2} + a - bL(\mathcal{T}) \\ &\geq -w(x) - \frac{b\lambda}{2} + a - 2bL(\mathcal{T}) \\ &\geq -w(x) - \frac{b\lambda}{2}. \end{aligned} \quad (117)$$

838 Hence, f satisfies the defining constraints of \mathbb{L} and we conclude that $f \in \mathbb{L}$. Therefore, $\mathbb{L}_a \subset \mathbb{L}$ for
839 all $a \geq 2bL(\mathcal{T})$.

840 It follows from Theorem B.2 and the definition of $\widetilde{\text{ET}}_\lambda^a$ that

$$\widetilde{\text{ET}}_\lambda^a(\mu, \nu) \leq \text{ET}_\lambda(\mu, \nu). \quad (118)$$

841 This concludes the proof. □

842 D.5 Proof for Proposition B.5

843 To ensure completeness, we provide full derivations of the result, closely following the methodology
844 of [38].

845 *Proof.* We first observe that the metric d_a depends solely on the value of the weight function at
846 the root r of the tree \mathcal{T} . This follows directly from the definition of \mathbb{L}_a , where only $w(r)$ appears
847 explicitly.

848 By construction, we have the variational characterization:

$$d_a(\mu, \nu) = \sup \left\{ \int_{\mathcal{T}} f(d\mu - d\nu) : f \in \mathbb{L}_a \right\}. \quad (119)$$

849 Let us now verify the metric properties:

850 **(Non-negativity)** Clearly, $d_a(\mu, \nu) \geq 0$ from the supremum structure. Moreover, $d_a(\mu, \mu) = 0$ for
 851 all μ by linearity of the integral. Now suppose that $d_a(\mu, \nu) = 0$. Using the closed-form expression
 852 for $\widetilde{\text{ET}}_\lambda^a$ in Proposition B.3, this implies:

$$\left[w(r) + \frac{b\lambda}{2} - a \right] \cdot |\mu(\mathcal{T}) - \nu(\mathcal{T})| + \int_{\mathcal{T}} |\mu(\Lambda(x)) - \nu(\Lambda(x))| \omega(dx) = 0. \quad (120)$$

853 Since the first term has a strictly positive coefficient by assumption ($a < w(r) + \frac{b\lambda}{2}$), we must have
 854 $\mu(\mathcal{T}) = \nu(\mathcal{T})$ and

$$\mu(\Lambda(x)) = \nu(\Lambda(x)) \quad \text{for all } x \in \mathcal{T}. \quad (121)$$

855 By [38, Lemma A.2], this implies that $\mu = \nu$, establishing identity of indiscernibles.

856 **(Symmetry)** Note that if $f \in \mathbb{L}_a$, then $-f \in \mathbb{L}_a$ by the symmetric definition of the function class.
 857 Therefore, from Equation (119), we obtain

$$d_a(\mu, \nu) = d_a(\nu, \mu). \quad (122)$$

858 **(Triangle Inequality)** The triangle inequality holds immediately from the supremum definition over
 859 a convex, symmetric function class:

$$d_a(\mu, \sigma) + d_a(\sigma, \nu) \geq \int_{\mathcal{T}} f(d\mu - d\sigma) + \int_{\mathcal{T}} f(d\sigma - d\nu) = \int_{\mathcal{T}} f(d\mu - d\nu), \quad (123)$$

860 for all $f \in \mathbb{L}_a$, and taking the supremum yields the inequality.

861 Hence, d_a satisfies all properties of a metric on $\mathcal{M}(\mathcal{T})$, and the proof is complete. \square

862 D.6 Proof for Equation (13)

863 *Proof.* We recall Equation (13). Let $f \in L^1(\mathbb{R}^d)$ be a non-negative density function. The Radon
 864 Transform \mathcal{R}^α maps f to a density defined on a tree system \mathcal{T} , while preserving the total mass:

$$\|f\|_1 = \int_{\mathbb{R}^d} f(x) dx = \|\mathcal{R}_\mathcal{T}^\alpha f\|_\mathcal{T}, \quad \text{for all } \mathcal{T} \in \mathbb{T}. \quad (124)$$

865 To establish this property, we first observe that the non-negativity of α ensures that the transform
 866 preserves non-negativity: if $f \geq 0$, then $\mathcal{R}_\mathcal{T}^\alpha f \geq 0$, implying that the transformed function is a
 867 valid density. The preservation of total mass then follows directly from the definition of \mathcal{R}^α , which
 868 integrates over linearly parameterized subsets aligned with the structure of \mathcal{T} .

$$\begin{aligned} \|\mathcal{R}_\mathcal{T}^\alpha f\|_\mathcal{T} &= \sum_{l \in \mathcal{L}} \int_{\mathbb{R}} |\mathcal{R}_\mathcal{T}^\alpha f(t_x, l)| dt_x \\ &= \sum_{l \in \mathcal{L}} \int_{\mathbb{R}} \left| \int_{\mathbb{R}^d} f(y) \cdot \alpha(y, \mathcal{L})_l \cdot \delta(t_x - \langle y - x_l, \theta_l \rangle) dy \right| dt_x \\ &= \sum_{l \in \mathcal{L}} \int_{\mathbb{R}} \left(\int_{\mathbb{R}^d} f(y) \cdot \alpha(y, \mathcal{L})_l \cdot \delta(t_x - \langle y - x_l, \theta_l \rangle) dy \right) dt_x \\ &= \sum_{l \in \mathcal{L}} \int_{\mathbb{R}^d} \left(\int_{\mathbb{R}} f(y) \cdot \alpha(y, \mathcal{L})_l \cdot \delta(t_x - \langle y - x_l, \theta_l \rangle) dt_x \right) dy \\ &= \sum_{l \in \mathcal{L}} \int_{\mathbb{R}^d} f(y) \cdot \alpha(y, \mathcal{L})_l \cdot \left(\int_{\mathbb{R}} \delta(t_x - \langle y - x_l, \theta_l \rangle) dt_x \right) dy \\ &= \sum_{l \in \mathcal{L}} \int_{\mathbb{R}^d} f(y) \cdot \alpha(y, \mathcal{L})_l dy \\ &= \int_{\mathbb{R}^d} f(y) \cdot \sum_{l \in \mathcal{L}} \alpha(y, \mathcal{L})_l dy \end{aligned}$$

$$\begin{aligned}
&= \int_{\mathbb{R}^d} f(y) dy \\
&= \|f\|_1.
\end{aligned} \tag{125}$$

869 The proof is completed. \square

870 **D.7 Proof for Theorem 3.3**

871 *Proof.* We consider the expression

$$\text{PartialTSW}(\mu, \nu) = \int_{\mathbb{T}} d_a(\mu_{\mathcal{T}}, \nu_{\mathcal{T}}) d\sigma(\mathcal{T}), \tag{126}$$

872 and show that it defines a metric on $\mathcal{M}(\mathbb{R}^d)$. Since the splitting map α is $E(d)$ -invariant, the Radon
873 Transform \mathcal{R}^α is injective; that is, for any $f \in L^1(\mathbb{R}^d)$, if $\mathcal{R}_{\mathcal{T}}^\alpha f = 0$ for all $\mathcal{T} \in \mathbb{T}$, then $f = 0$
874 (see [73]). We now verify the three properties required for PartialTSW to be a metric on $\mathcal{M}(\mathbb{R}^d)$.

875 **Positive definiteness.** For $\mu, \nu \in \mathcal{P}(\mathbb{R}^n)$, it is clear that $\text{PartialTSW}(\mu, \mu) = 0$ and
876 $\text{PartialTSW}(\mu, \nu) \geq 0$. Moreover, if $\text{PartialTSW}(\mu, \nu) = 0$, then $d_a(\mu_{\mathcal{T}}, \nu_{\mathcal{T}}) = 0$ for all $\mathcal{T} \in \mathbb{T}$.
877 Since d_a is a metric on $\mathcal{M}(\mathcal{T})$, it follows that $\mu_{\mathcal{T}} = \nu_{\mathcal{T}}$ for all \mathcal{T} . Hence, $\mathcal{R}_{\mathcal{T}}^\alpha f_\mu = \mathcal{R}_{\mathcal{T}}^\alpha f_\nu$ for all
878 $\mathcal{T} \in \mathbb{T}$. By the injectivity of \mathcal{R}^α , we conclude that $f_\mu = f_\nu$, and thus $\mu = \nu$.

879 **Symmetry.** For any $\mu, \nu \in \mathcal{M}(\mathbb{R}^n)$, we have:

$$\begin{aligned}
\text{PartialTSW}(\mu, \nu) &= \int_{\mathbb{T}} d_a(\mu_{\mathcal{T}}, \nu_{\mathcal{T}}) d\sigma(\mathcal{T}) \\
&= \int_{\mathbb{T}} d_a(\nu_{\mathcal{T}}, \mu_{\mathcal{T}}) d\sigma(\mathcal{T}) = \text{PartialTSW}(\nu, \mu).
\end{aligned} \tag{127}$$

880 Therefore, $\text{PartialTSW}(\mu, \nu) = \text{PartialTSW}(\nu, \mu)$.

881 **Triangle inequality.** For $\mu_1, \mu_2, \mu_3 \in \mathcal{M}(\mathbb{R}^n)$, we compute:

$$\begin{aligned}
&\text{PartialTSW}(\mu_1, \mu_2) + \text{PartialTSW}(\mu_2, \mu_3) \\
&= \int_{\mathbb{T}} d_a(\mu_{1,\mathcal{T}}, \mu_{2,\mathcal{T}}) d\sigma(\mathcal{T}) + \int_{\mathbb{T}} d_a(\mu_{2,\mathcal{T}}, \mu_{3,\mathcal{T}}) d\sigma(\mathcal{T}) \\
&= \int_{\mathbb{T}} (d_a(\mu_{1,\mathcal{T}}, \mu_{2,\mathcal{T}}) + d_a(\mu_{2,\mathcal{T}}, \mu_{3,\mathcal{T}})) d\sigma(\mathcal{T}) \\
&\geq \int_{\mathbb{T}} d_a(\mu_{1,\mathcal{T}}, \mu_{3,\mathcal{T}}) d\sigma(\mathcal{T}) \\
&= \text{PartialTSW}(\mu_1, \mu_3),
\end{aligned} \tag{128}$$

882 where the inequality follows from the triangle inequality satisfied by d_a on each tree \mathcal{T} .

883 In conclusion, PartialTSW satisfies all properties of a metric on the space $\mathcal{M}(\mathbb{R}^d)$.

884 We aim to show that PartialTSW is $E(d)$ -invariant, meaning that for any $g \in E(d)$, the following
885 holds:

$$\text{PartialTSW}(\mu, \nu) = \text{PartialTSW}(g\sharp\mu, g\sharp\nu), \tag{129}$$

886 where $g\sharp\mu$ and $g\sharp\nu$ denote the pushforwards of μ and ν , respectively, under the Euclidean transforma-
887 tion $g: \mathbb{R}^d \rightarrow \mathbb{R}^d$.

888 Let $\mathcal{T} \in \mathbb{T}$ be a tree system given by $\mathcal{T} = \{l_i = (x_i, \theta_i)\}_{i=1}^k$. Then, under the action of $g = (Q, a)$,
889 we have

$$g\mathcal{T} = \{gl_i = (Qx_i + a, Q\theta_i)\}_{i=1}^k. \tag{130}$$

890 We also note that $g\sharp f_\mu = f_{g\sharp\mu}$ and $g\sharp f_\nu = f_{g\sharp\nu}$. Since $|\det(Q)| = 1$, we compute:

$$\mathcal{R}_{g\mathcal{L}}^\alpha(g\sharp f_\mu)(gx, gl) = \int_{\mathbb{R}^d} (g\sharp f_\mu)(y) \cdot \alpha(y, g\mathcal{L})_l \cdot \delta(t_{gx} - \langle y - x_{gl}, \theta_{gl} \rangle) dy$$

$$\begin{aligned}
&= \int_{\mathbb{R}^d} f_\mu(g^{-1}y) \cdot \alpha(y, g\mathcal{L})_l \cdot \delta(t_x - \langle y - x_{gl}, \theta_{gl} \rangle) dy \\
&= \int_{\mathbb{R}^d} f_\mu(g^{-1}gy) \cdot \alpha(gy, g\mathcal{L})_l \cdot \delta(t_x - \langle gy - x_{gl}, \theta_{gl} \rangle) d(gy) \\
&= \int_{\mathbb{R}^d} f_\mu(y) \cdot \alpha(y, \mathcal{L})_l \cdot \delta(t_x - \langle gy - x_{gl}, \theta_{gl} \rangle) dy \\
&= \int_{\mathbb{R}^d} f_\mu(y) \cdot \alpha(y, \mathcal{L})_l \cdot \delta(t_x - \langle Qy + a - Qx_l - a, Q\theta_l \rangle) dy \\
&= \int_{\mathbb{R}^d} f_\mu(y) \cdot \alpha(y, \mathcal{L})_l \cdot \delta(t_x - \langle Q(y - x_l), Q\theta_l \rangle) dy \\
&= \int_{\mathbb{R}^d} f_\mu(y) \cdot \alpha(y, \mathcal{L})_l \cdot \delta(t_x - \langle y - x_l, \theta_l \rangle) dy \\
&= \mathcal{R}_{\mathcal{L}}^\alpha f_\mu(x, l).
\end{aligned} \tag{131}$$

891 A similar computation gives:

$$\mathcal{R}_{g\mathcal{L}}^\alpha(g\sharp f_\nu)(gx, gl) = \mathcal{R}_{\mathcal{L}}^\alpha f_\nu(x, l). \tag{132}$$

892 Moreover, since g acts isometrically on tree systems, the induced measures satisfy:

$$d_a(\mu_{\mathcal{T}}, \nu_{\mathcal{T}}) = d_a((g\sharp\mu)_{g\mathcal{T}}, (g\sharp\nu)_{g\mathcal{T}}). \tag{133}$$

893 Thus, we compute:

$$\begin{aligned}
\text{PartialTSW}(g\sharp\mu, g\sharp\nu) &= \int_{\mathbb{T}} d_a((g\sharp\mu)_{\mathcal{T}}, (g\sharp\nu)_{\mathcal{T}}) d\sigma(\mathcal{T}) \\
&= \int_{\mathbb{T}} d_a((g\sharp\mu)_{g\mathcal{T}}, (g\sharp\nu)_{g\mathcal{T}}) d\sigma(g\mathcal{T}) \\
&= \int_{\mathbb{T}} d_a(\mu_{\mathcal{T}}, \nu_{\mathcal{T}}) d\sigma(g\mathcal{T}) \\
&= \int_{\mathbb{T}} d_a(\mu_{\mathcal{T}}, \nu_{\mathcal{T}}) d\sigma(\mathcal{T}) \\
&= \text{PartialTSW}(\mu, \nu).
\end{aligned} \tag{134}$$

894 We conclude that PartialTSW is $E(d)$ -invariant. \square

895 **Remark D.1.** We omit almost-sure conditions in the above proof, as they are straightforward to
896 verify and would otherwise obscure the main argument.

897 E Experimental Details

898 E.1 Algorithm for Partial Tree-Sliced Wasserstein Distance

899 The computation of the Partial Tree-Sliced Wasserstein (PartialTSW) distance is outlined in Algo-
900 rithm 1. This procedure estimates the distance by averaging costs derived from multiple tree-based
901 projections of the input measures.

Algorithm 1 Partial Tree-Sliced Wasserstein distance.

Input: Measures μ and ν in $\mathcal{M}(\mathbb{R}^d)$, number of tree systems L , number of lines in tree system k , space of tree systems \mathbb{T} , splitting maps α , parameters a, b, λ , total mass $\mu(\mathcal{T}), \nu(\mathcal{T})$.

Scale total mass of μ and ν such that $\mu(\mathbb{R}^d) = \mu(\mathcal{T}), \nu(\mathbb{R}^d) = \nu(\mathcal{T})$.

for $i = 1$ to L **do**

 Sampling $x \in \mathbb{R}^d$ and $\theta_1, \dots, \theta_k \stackrel{i.i.d.}{\sim} \mathcal{U}(\mathbb{S}^{d_\theta-1})$.

 Construct tree system $\mathcal{L}_i = \{(x, \theta_1), \dots, (x, \theta_k)\}$.

 Projecting μ and ν onto \mathcal{T}_i to get $\mathcal{R}_{\mathcal{L}_i}^\alpha \mu$ and $\mathcal{R}_{\mathcal{L}_i}^\alpha \nu$.

 Compute $\widehat{\text{PartialTSW}}(\mu, \nu) = (1/L) \cdot d_a(\mathcal{R}_{\mathcal{L}_i}^\alpha \mu, \mathcal{R}_{\mathcal{L}_i}^\alpha \nu)$.

end for

Return: $\widehat{\text{PartialTSW}}(\mu, \nu)$.

E.2 Computational and Memory Complexity Analysis

This section details the computational and memory demands of our proposed PartialTSW distance. We consider input measures μ and ν represented by N samples in a d -dimensional space, with L tree constructions and k lines per tree.

Table 3 outlines the complexity of key operations. The dominant factors are the distance-based weight splitting ($O(LkNd)$) for projecting samples and the sorting of these projected 1D coordinates ($O(LkN \log N)$). Consequently, the total computational complexity is $O(LkNd + LkN \log N)$. The primary memory consumers are the storage of split weights, tree/line parameters, and the original data, leading to an overall memory requirement of $O(LkN + Lkd + Nd)$.

Table 3: Detailed complexity analysis for Partial TSW. (N = number of samples, d = dimension, L = number of trees, k = lines per tree).

Operation Category	Specific Steps Involved	Computational Cost	Memory Cost
Initial Mass Scaling	Adjusting sample weights for μ and ν to meet target total masses.	$O(N)$	$O(N)$
Distance-Based Weight Splitting	Calculation of distances from N points to Lk lines, and subsequent softmax for weight distribution.	$O(LkNd)$	$O(LkN + Lkd + Nd)$
Sorting Projected Data	Sorting the N projected coordinates along each of the Lk lines.	$O(LkN \log N)$	$O(LkN)$
Overall Total		$O(LkNd + LkN \log N)$	$O(LkN + Lkd + Nd)$

GPU Memory Optimization for Distance-Based Splitting. The practical GPU memory footprint for the distance-based splitting step can be significantly lower than a naive theoretical estimate. As highlighted by [73], this operation involves (1) computing d -dimensional distance vectors from points to lines, (2) calculating their norms, and (3) applying a softmax function across lines within each tree to obtain split weights. While a direct implementation might suggest $O(LkNd)$ memory for storing all intermediate distance vectors, modern deep learning frameworks like PyTorch, when using compilation tools (e.g., ‘torch.compile’), can perform kernel fusion. This optimization merges these sequential computations into fewer GPU kernels, potentially allowing large intermediate tensors (like the full $LkN \times d$ distance vectors) to reside in faster, smaller shared memory or be recomputed on-the-fly, rather than occupying global GPU memory. Consequently, the persistent global memory primarily stores the essential data: line parameters ($O(Lkd)$), sample coordinates ($O(Nd)$), and the resulting split weights ($O(LkN)$), aligning with the $O(LkN + Lkd + Nd)$ overall memory profile.

E.3 Empirical Runtime and Memory Performance of Partial TSW

We present an empirical evaluation of the runtime and memory usage of PartialTSW. The experiments were conducted on a single NVIDIA H100 GPU. We fixed the number of tree iterations $L = 10$ and lines per tree $k = 4$. The analysis varies the number of samples $N \in \{100, 1k, 5k, 10k, 500k\}$ and the data dimension $d \in \{50, 100, 500, 1000\}$.

Runtime Scalability. The empirical results, depicted in Figure 7 (left), illustrate how the runtime of Partial TSW scales with the number of samples N and the data dimension d . The runtime exhibits a near-linear increase with N . For instance, processing $N = 50,000$ samples takes approximately five times longer than $N = 10,000$ samples (when d, L, k are fixed), which is consistent with the $O(Nd + N \log N)$ dependency on N from our theoretical analysis (Section E.2). Regarding dimensionality, the runtime also demonstrates a linear dependency on d . For example, increasing d from 10000 to 50000 (a 5x increase) results in a correspondingly proportional increase in runtime for a fixed N . This aligns with the $O(d)$ factor in the $LkNd$ term of the complexity. These empirical observations support the theoretical computational complexity.

Memory Scalability. Figure 7 (right) showcases the memory consumption characteristics of Partial TSW. The peak memory usage scales linearly with both the number of samples N and the dimension d . This behavior is predictable and directly corresponds to our theoretical memory complexity of $O(LkN + Lkd + Nd)$, indicating efficient memory utilization that grows manageably with data size and dimensionality.

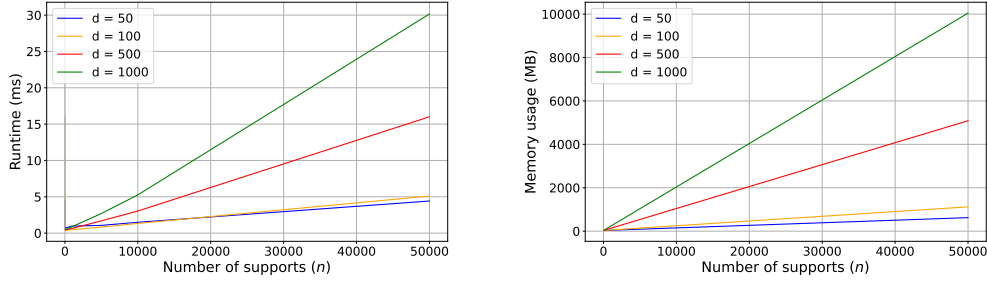


Figure 7: Empirical runtime (left) and peak memory usage (right) for Partial TSW, varying the number of samples (N) and data dimension (d). ($L = 10, k = 4$).

E.4 Discussion on hyper-parameters of evaluated methods

This section briefly outlines the key hyper-parameters for each evaluated Unbalanced Optimal Transport (UOT) and Partial Optimal Transport (POT) method and their respective roles.

SPOT [9]. The hyperparameter k specifies the number of points to be transported, thereby defining the partial nature of the matching between distributions.

SOPT [2]. The regularization parameter λ controls the "partialness" of the transport by influencing the total amount of mass that is optimally transported between distributions.

Sinkhorn [68]. The hyperparameter reg is the entropic regularization coefficient that smooths the optimal transport plan. The hyperparameter reg_m is the marginal regularization coefficient that penalizes deviations from the prescribed marginal constraints, thus allowing for mass variation.

SUOT and USOT [7]. The hyper-parameters ρ_1 and ρ_2 are regularization parameters. They respectively control the cost of deviating from the source and target marginals in the sliced domain, enabling unbalanced transport by permitting mass creation or destruction.

PAWL [13]. The hyperparameter k the number of points to be transported, effectively determining the extent of partiality in this unbalanced optimal transport formulation.

UOT-FM [21]. The hyperparameter λ influences the regularization of marginal constraints, thereby controlling the degree to which the masses of the coupled distributions must be preserved during transport.

ULightOT [26]. The hyperparameter τ governs the extent of mass conservation, adjusting how strictly the total mass of the transported distribution must adhere to the original or target masses.

Partial-TSW (Ours). The mass parameter $\nu(\mathcal{T})$ specifies the proportion of the target distribution's mass to be matched by the transport plan. The source distribution's mass proportion, $\mu(\mathcal{T})$, is typically fixed at 1, so adjusting $\nu(\mathcal{T})$ controls the partiality of the matching against the target.

E.5 Comparing Computational Efficiency

To ensure consistent and fair results, two warm-up runs were performed for each method and each sample size n before conducting 10 timed repetitions. The average runtime and peak memory usage (for GPU methods) were then recorded. Unless otherwise specified (as in the discussion on varying d below), these experiments were conducted with data of dimension $d = 2$ and for sample sizes n ranging from 10^2 to 10^5 .

Since hyperparameter choices can significantly affect algorithmic runtime, the specific settings used for each method in this runtime comparison are detailed below. For a general description of these hyperparameters and their roles, please refer to Appendix E.4.

Common settings for the compared sliced-based methods (SOPT, SPOT, USOT, SUOT, PAWL) included $L = 10$ projections. For PartialTSW (Ours), we used `num_trees` = 5 and `num_lines` = 2. This configuration for PartialTSW, where the product of `num_trees` \times `num_lines` = 10, offers a comparable number of one-dimensional sorting operations to the $L = 10$ setting in other sliced

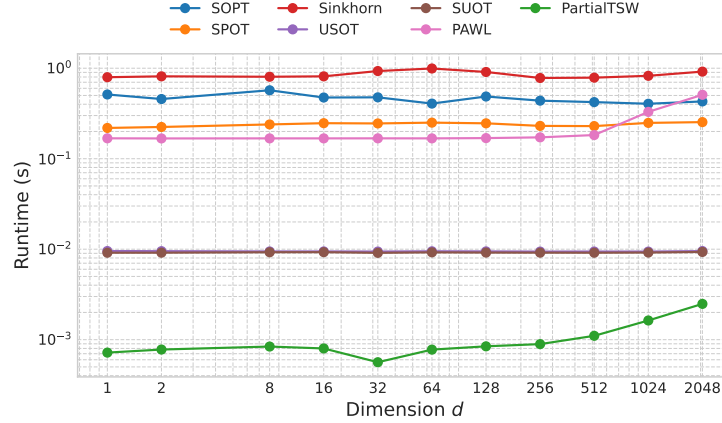


Figure 8: Runtime comparison for PartialTSW and POT/UOT solvers over data dimension d .

978 methods, aiming for a fair comparison. Specific hyperparameters for each method were then set as
 979 follows:

- 980 • **SOPT**: Regularization parameter $\lambda = 1.0$.
- 981 • **SPOT**: The number of transported points k was set to n (the input sample size for each
 982 distribution), implying a full matching was performed. (Number of projections $L = 10$, as
 983 stated above).
- 984 • **Sinkhorn**: Entropic regularization $reg = 0.1$, marginal KL regularization $reg_m = 1.0$,
 985 maximum number of Sinkhorn iterations ‘numItermax’ = 100, and stopping threshold
 986 ‘stopThr’ = 10^{-5} .
- 987 • **USOT and SUOT**: Regularization parameters $\rho_1 = 0.01$ and $\rho_2 = 1.0$.
- 988 • **PAWL**: The number of transported points k was set to n (implying a full matching).
- 989 • **PartialTSW (Ours)**: The target mass proportion $\nu(\mathcal{T})$ was set to 1.0 (with the source mass
 990 proportion $\mu(\mathcal{T})$ typically assumed to be 1.0). This choice was made because $\nu(\mathcal{T})$ does not
 991 affect the computational runtime of the PartialTSW implementation used in this benchmark.

992 Furthermore, we present a runtime comparison for varying data dimensions d in Figure 8. The results
 993 indicate that the runtime is not significantly affected when d increases.

994 The runtime comparisons for all methods were conducted with an Intel Xeon Platinum 8580 CPU
 995 and an NVIDIA H100 GPU.

996 E.6 Noisy Point Cloud Gradient Flow

997 We used clean point cloud data obtained from [2] for the dragon and bunny shapes. Each clean
 998 dataset contains 10k data points. We randomly select and add 7% noise points to the target point
 999 cloud (bunny). Inspired by [2], the noise is sampled from the region $[-0.6M, 0.6M]^3$ where
 1000 $M = \max_{i \in \{1, n\}} (\|x_i\|)$, where x_i is the point in the target. In total, the target point cloud consists
 1001 of 10k clean points and an additional 700 noise points. We use $L = 10$ projections for SW, and
 1002 $L = 5$ trees, $k = 2$ lines for TSW and PartialTSW. All methods are trained using Adam optimizer
 1003 with a learning rate of 1e-3 over 300 epochs. The results are shown in Figure 3.

1004 All experiments were conducted with an Intel Xeon Platinum 8580 CPU and an NVIDIA H100 GPU.

1005 E.7 Robust Generative Model

1006 E.7.1 Implementation detail

1007 **Pre-training an Autoencoder (AE)**. An Autoencoder (AE) is pre-trained to provide 2D latent
 1008 representations $z \in \mathbb{R}^2$ for MNIST digits. We employ a Wasserstein Autoencoder with MMD

regularization (WAE-MMD) [72] architecture. The AE is trained for 50 epochs using the Adam optimizer with a learning rate of 3×10^{-5} and a batch size of 256. The WAE-MMD loss uses a λ hyperparameter of 500.0 to balance reconstruction and MMD regularization terms. For the MMD term, we match the aggregated posterior $q(z)$ to a uniform prior distribution $p(z) \sim \mathcal{U}[-1, 1]^2$. This encourages the learned latent space to reside approximately within $[-1, 1]^2$. The training data for the AE consists of MNIST digits 0 and 1, balanced and augmented as described below. The latent dimension is set to $d = 2$.

The Autoencoder, $AE : [0, 1]^{1 \times 28 \times 28} \rightarrow [0, 1]^{1 \times 28 \times 28}$, architecture is as follows:

- **Encoder:**
 - Input: $1 \times 28 \times 28$ (MNIST image)
 - Conv2d(in_channels = 1, out_channels = 32, kernel_size = 4, stride = 2, padding = 1) \rightarrow ReLU ($32 \times 14 \times 14$)
 - Conv2d(32, 64, kernel_size = 4, stride = 2, padding = 1) \rightarrow ReLU ($64 \times 7 \times 7$)
 - Flatten: $64 \times 7 \times 7 = 3136$ features
 - Linear(in_features = 3136, out_features = 512) \rightarrow ReLU
 - Linear(512, latent_dim = 2) (for mean μ)
 - Linear(512, latent_dim = 2) (for log-variance $\log \sigma^2$)
 - Latent vector $z = \mu + \epsilon \odot \sigma$ (Reparameterization trick)
- **Decoder:**
 - Input: $z \in \mathbb{R}^{\text{latent_dim}=2}$
 - Linear(latent_dim = 2, 512) \rightarrow ReLU
 - Linear(512, 3136) \rightarrow ReLU
 - Reshape to $64 \times 7 \times 7$
 - ConvTranspose2d(64, 32, kernel_size = 4, stride = 2, padding = 1) \rightarrow ReLU ($32 \times 14 \times 14$)
 - ConvTranspose2d(32, 1, kernel_size = 4, stride = 2, padding = 1) \rightarrow Sigmoid ($1 \times 28 \times 28$)

Dataset Augmentation for Auxiliary Models. To ensure robust training of the AE and the digit classifier, we prepare a balanced and augmented training set from MNIST digits 0 and 1. The original MNIST training set contains an unequal number of samples for these digits. We balance these classes by applying data augmentation to the minority class until its sample count matches the majority class. Augmentations include random affine transformations (degrees: $\pm 15^\circ$, translation: ± 0.15 of image dimension, scale: $0.85 - 1.15 \times$) and random rotations ($\pm 15^\circ$). This balanced and augmented dataset is used exclusively for training the AE and the binary (0 vs. 1) digit classifier. We found that having a balanced dataset for training AE would lead to a balanced latent space for MNIST Digit 0 and 1.

Pre-training an MNIST Digit Classifier. A convolutional neural network classifier is pre-trained to distinguish between MNIST digits 0 and 1. It is trained for 20 epochs on the balanced and augmented dataset of these two digits, using the Adam optimizer with a learning rate of 1×10^{-3} and a Cross-Entropy loss function. This classifier achieves approximately 99.99% accuracy on a test set of unseen MNIST 0s and 1s and is subsequently used (with frozen weights) to evaluate the class labels of images generated by the main generative model.

The Classifier, $C : [0, 1]^{1 \times 28 \times 28} \rightarrow \mathbb{R}^2$, architecture is as follows:

- Input: $1 \times 28 \times 28$ (decoded image)
- Conv2d(1, 32, kernel_size = 3, stride = 1, padding = 1) \rightarrow ReLU ($32 \times 28 \times 28$)
- MaxPool2d(kernel_size = 2, stride = 2) ($32 \times 14 \times 14$)
- Conv2d(32, 64, kernel_size = 3, stride = 1, padding = 1) \rightarrow ReLU ($64 \times 14 \times 14$)
- MaxPool2d(kernel_size = 2, stride = 2) ($64 \times 7 \times 7$)
- Flatten: $64 \times 7 \times 7 = 3136$ features
- Linear(3136, 128) \rightarrow ReLU
- Linear(128, num_classes = 2) (Logits for classes 0 and 1)

1059 **Constructing the Observed (Contaminated) Dataset X_{obs}** The observed dataset X_{obs} for training
 1060 the generator G consists of latent representations. These are obtained by encoding MNIST images
 1061 of digits 0 (target class) and 1 (outlier class) using the pre-trained AE’s encoder. Specifically, X_{obs}
 1062 is a mixture comprising 90% samples from the true latent distribution of digit 0 (\mathcal{X}_0) and 10%
 1063 samples (outliers) from the true latent distribution of digit 1 (\mathcal{X}_1). To construct this, we sample latent
 1064 vectors z' from the prior $\mathcal{U}[-1, 1]^2$, decode them to images $x' = AE_{\text{dec}}(z')$, and classify x' using
 1065 the pre-trained 0/1 classifier. If x' is classified as 0 (or 1), z' is added to a pool for \mathcal{X}_0 (or \mathcal{X}_1). We
 1066 collect samples until we can form a dataset of $N_{\text{obs}} = 50,000$ latent points, with the 90/10 proportion.
 1067 These latent points constitute X_{obs} and are scaled to approximately reside within $[-1, 1]^2$.

1068 **Training the Generator G .** The generator $G : \mathcal{N}(0, I_2) \rightarrow [-1, 1]^2$ is a multi-layer perceptron
 1069 (MLP) designed to map 2D Gaussian noise $Z \sim \mathcal{N}(0, I_2)$ to the target latent space. The generator is
 1070 trained by minimizing a (Partial) Optimal Transport distance $D(G(Z), X_{\text{obs}})$, where Z is a batch of
 1071 noise samples. Training is performed for 30 epochs using the Adam optimizer with a learning rate
 1072 of 2×10^{-4} and a batch size of 256. Specific (P)OT-based distances D used for PartialTSW and
 1073 baseline methods are detailed in the main paper.

1074 The generator architecture is:

- 1075 • Input: $Z \in \mathbb{R}^2 \sim \mathcal{N}(0, I_2)$
- 1076 • Linear(2, 4) \rightarrow BatchNorm1d(4) \rightarrow LeakyReLU(0.2)
- 1077 • Linear(4, 8) \rightarrow BatchNorm1d(8) \rightarrow LeakyReLU(0.2)
- 1078 • Linear(8, 2) \rightarrow Tanh (Output $z_{\text{gen}} \in [-1, 1]^2$)

1079 **Evaluation.** To evaluate the generator’s ability to learn the target distribution \mathcal{X}_0 while ignoring
 1080 outliers from \mathcal{X}_1 , we employ two main criteria:

- 1081 1. **Outlier Rate:** We generate $N_{\text{eval}} = 5,000$ latent samples $z_{\text{gen}} = G(Z)$. These latent
 1082 samples are decoded into images $\hat{x} = AE_{\text{dec}}(z_{\text{gen}})$ using the pre-trained AE’s decoder. The
 1083 resulting images are then classified by the pre-trained 0/1 digit classifier. The outlier rate
 1084 is the percentage of generated images classified as digit 1. A lower rate indicates better
 1085 robustness.
- 1086 2. **Sample Quality and Diversity:** We qualitatively assess the generated samples by visualizing
 1087 the decoded images \hat{x} and their corresponding latent representations z_{gen} . We look for
 1088 high-fidelity generation of digit 0 and good coverage of its variations, as indicated by a
 1089 well-distributed latent space for the generated samples classified as 0.

1090 Performance summaries, including outlier rates and visual comparisons, are provided in Figure 4 and
 1091 Table 1 in the main text.

1092 **Hardware Settings.** The experiments for all methods were conducted on a system equipped with an
 1093 Intel Xeon Platinum 8580 CPU and one NVIDIA H100 GPU.

1094 E.7.2 Ablation result for baselines

1095 We evaluate the impact of hyperparameter settings on each method’s ability to isolate the target
 1096 MNIST 0 distribution from the 10% MNIST 1 outliers present in the training data. The following
 1097 summarizes these ablation results (Figures 9–15), focusing on the percentage of generated MNIST 1
 1098 outliers and, where applicable, qualitative aspects of the learned distributions and generated samples.

1099 **SPOT [9].** Figure 9 demonstrates SPOT’s varying success in isolating the target MNIST 0 data from
 1100 the 10% 1 outliers, contingent on its hyperparameter k . While very small k values (e.g., $k = 10$,
 1101 yielding 45.62% MNIST 1 outliers) or very large k values (e.g., $k = 256$, yielding 16.20% outliers)
 1102 result in poor outlier rejection, an optimal range for k around 200 – 210 reduces the MNIST 1 outlier
 1103 rate to 6 – 7%. This indicates substantial but incomplete removal of the 10% outliers.

1104 **SOPT [2].** SOPT’s effectiveness in discarding the 10% MNIST 1 outliers is modulated by its
 1105 regularization parameter λ , as shown in Figure 10. The lowest outlier percentage achieved by SOPT
 1106 is 13.28% (at $\lambda = 0.01$), which still exceeds the initial 10% contamination level. Larger values of λ
 1107 lead to even higher and relatively stable outlier rates (around 15 – 16.42%), indicating a persistent
 1108 difficulty for SOPT in cleanly separating the target distribution in this setup.

1109 **Sinkhorn [68].** Sinkhorn shows the potential for complete removal of the 10% MNIST 1 outliers
1110 when its entropic regularization reg and marginal regularization reg_m are appropriately co-tuned
1111 (Figure 11). Specifically, setting $reg = reg_m$ at values of 0.5, 0.7, or 0.9 results in 0% MNIST 1
1112 outlier generation, successfully achieving the task’s objective. However, imbalanced or overly small
1113 regularization values lead to substantial outlier contamination (e.g., 52.48% for $reg = reg_m = 0.3$,
1114 or 70.56% for $reg = 0.9, reg_m = 0.1$). Moreover, qualitative inspection of the results (Figure 11,
1115 particularly for $reg = reg_m \in \{0.5, 0.7, 0.9\}$) reveals that while Sinkhorn effectively removes
1116 outliers, the generated latent distribution for MNIST 0 digits appears clustered, and the corresponding
1117 decoded images may lack diversity compared to the true distribution. This suggests a potential
1118 trade-off between perfect outlier rejection and capturing the full diversity of the target class for this
1119 method under these settings.

1120 **SUOT [7].** The performance of SUOT in the task of removing 10% MNIST 1 outliers is consistently
1121 poor across the explored range of its marginal regularization parameters ρ_1 and ρ_2 , as detailed in
1122 Figure 12. The method yields a high MNIST 1 outlier rate of approximately 41% regardless of the
1123 hyperparameter settings tested, indicating a failure to distinguish the target MNIST 0 distribution
1124 from the contaminants.

1125 **USOT [7].** USOT, while performing better than SUOT, still struggles to fully reject the 10%
1126 MNIST 1 outliers (Figure 13). Across the tested range of its ρ_1 and ρ_2 hyperparameters, USOT yields
1127 a consistent MNIST 1 outlier rate of approximately 17.08%. This suggests that while it mitigates
1128 some contamination, it does not fully isolate the target MNIST 0 distribution in this scenario.

1129 **PAWL [13].** PAWL demonstrates exceptional success in the goal of removing 10% MNIST 1 outliers,
1130 as shown in its ablation study (Figure 14). It consistently achieves a 0% MNIST 1 outlier rate across
1131 all tested values of its hyperparameter k (from 10 to 256). This indicates PAWL’s strong capability
1132 to identify and learn the target MNIST 0 distribution while completely ignoring outliers, exhibiting
1133 robust performance across a wide range of k . However, as noted in the main text and suggested by
1134 qualitative inspection of Figure 14, this strong outlier rejection by PAWL may be accompanied by
1135 less sample diversity, with its learned latent space showing heavily concentrated clusters.

1136 **PartialTSW (Ours).** Our PartialTSW method shows strong capabilities in removing the 10%
1137 MNIST 1 outliers, with performance critically depending on its mass parameter $\nu(\mathcal{T})$ (Figure 15).
1138 Complete outlier rejection (0% MNIST 1 outliers) is achieved for $\nu(\mathcal{T})$ values between 0.3 and 0.6.
1139 Setting $\nu(\mathcal{T})$ closer to the true inlier fraction of 0.9 (which yields 9.72% MNIST 1 outliers) leads to
1140 the model fitting the outliers. This highlights that optimal robustness for PartialTSW is achieved when
1141 $\nu(\mathcal{T})$ is chosen to be somewhat less than the actual inlier data proportion in the contaminated dataset.
1142 Qualitatively, as seen in Figure 15 and highlighted in our main findings, the settings achieving
1143 complete outlier rejection (e.g., $\nu(\mathcal{T}) \in [0.3, 0.6]$) also yield a well-distributed latent space and
1144 diverse image samples for the MNIST 0 class, effectively capturing the target distribution.

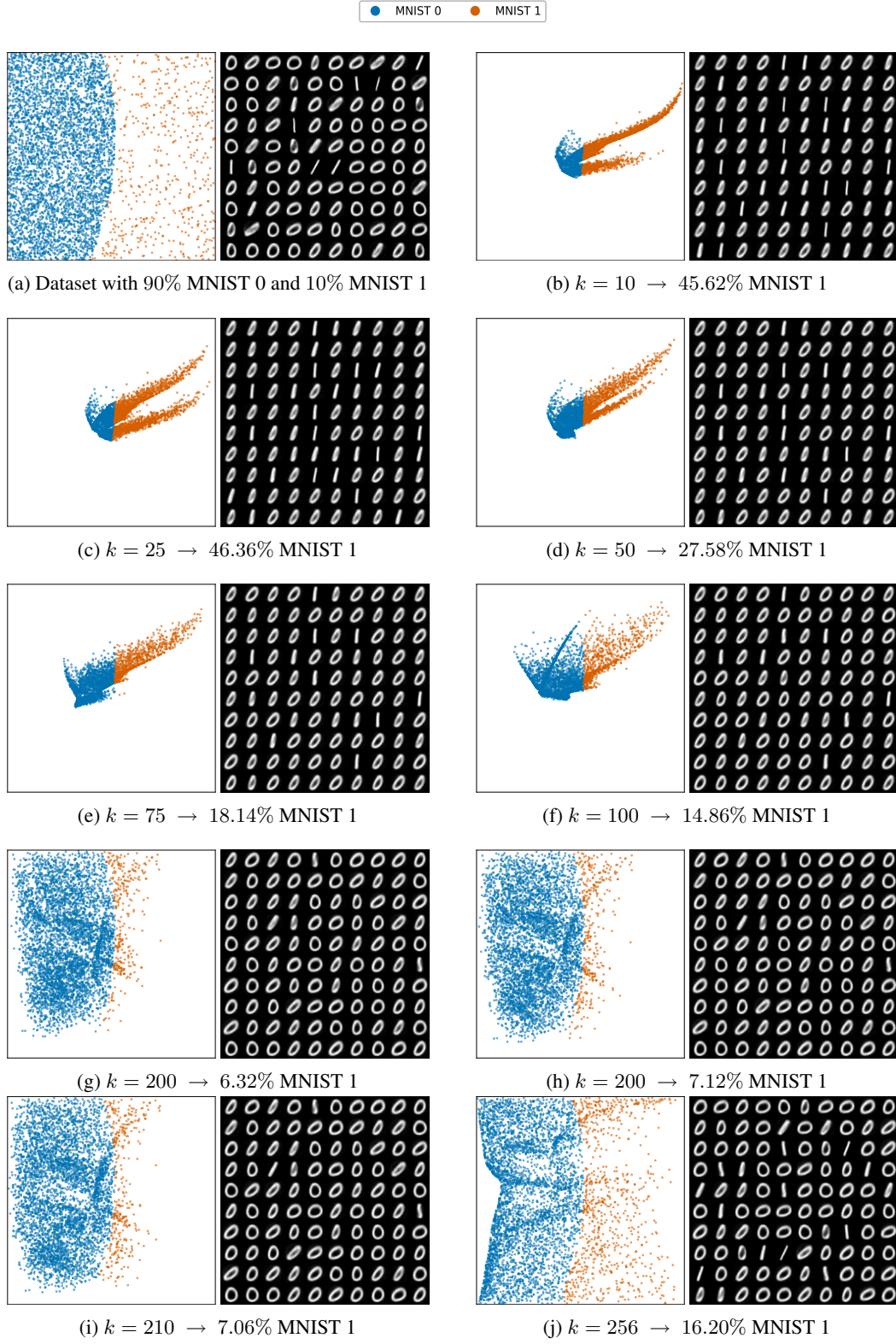


Figure 9: Ablation study of SPOT [9] for robust image generation. The figure illustrates the percentage of generated MNIST 1 digits (outliers), along with the corresponding learned latent distributions (left) and decoded image samples (right).

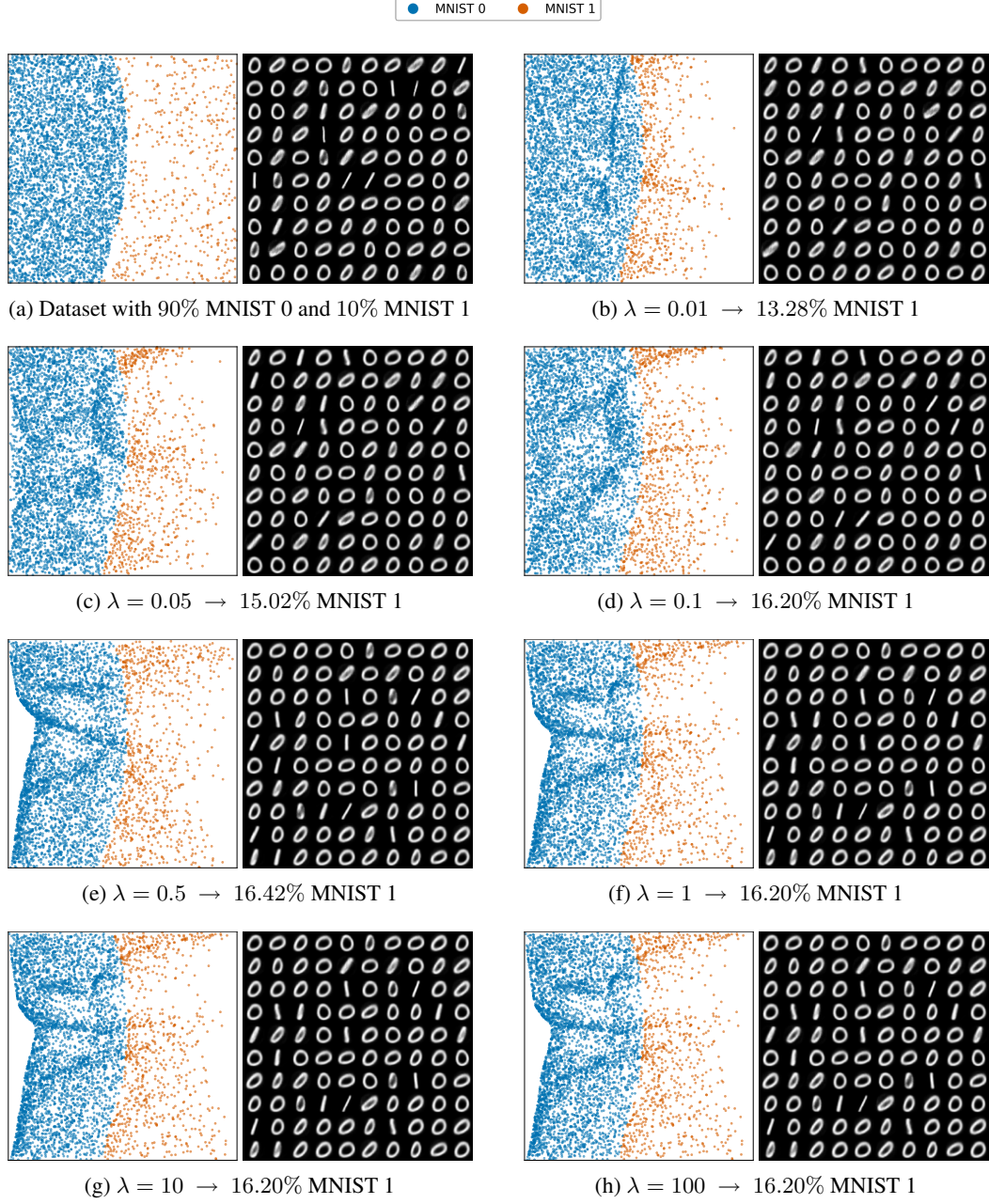


Figure 10: Ablation study of SOPT [2] for robust image generation. The figure illustrates the percentage of generated MNIST 1 digits (outliers), along with the corresponding learned latent distributions (left) and decoded image samples (right).

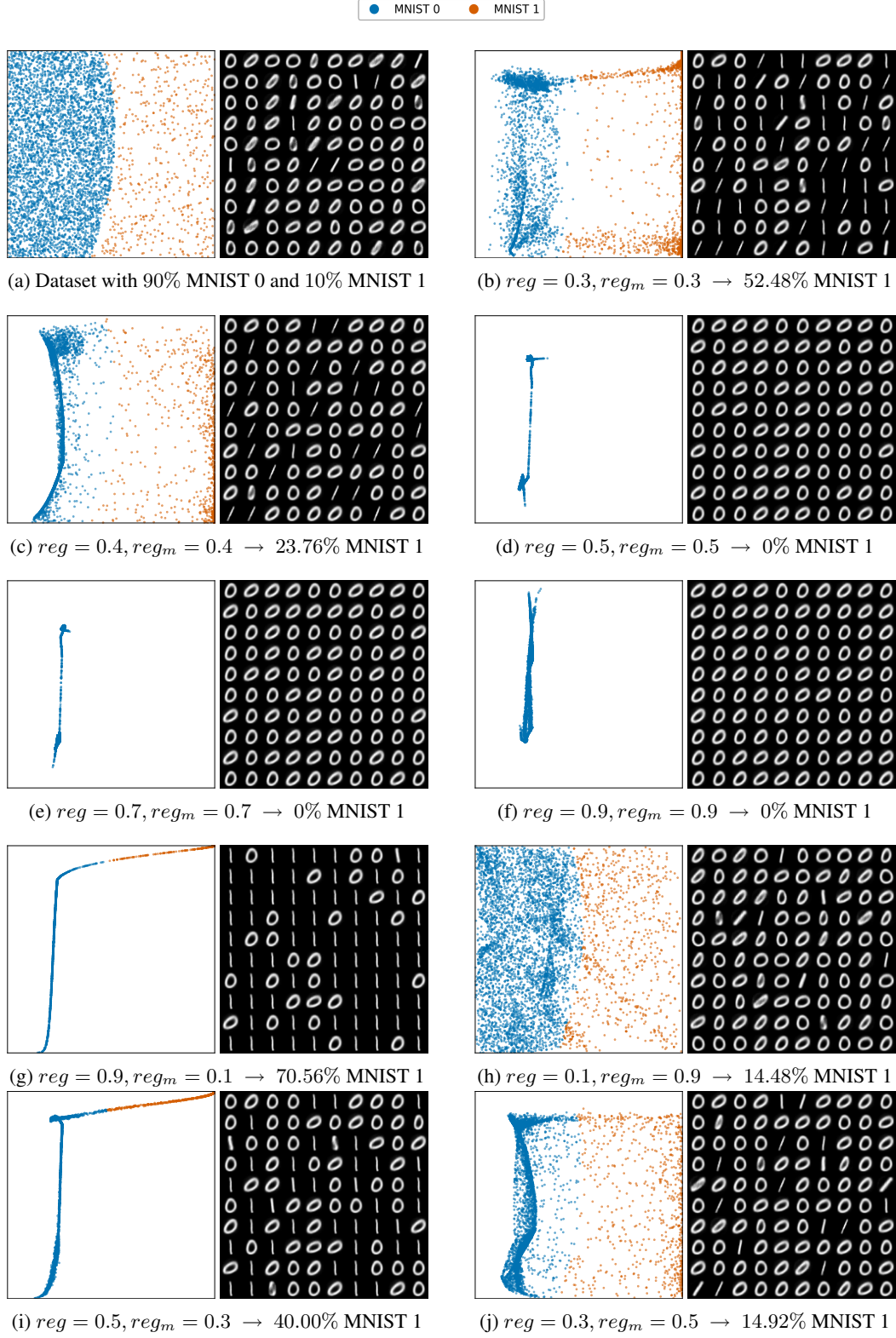


Figure 11: Ablation study of Sinkhorn [68] for robust image generation. The figure illustrates the percentage of generated MNIST 1 digits (outliers), along with the corresponding learned latent distributions (left) and decoded image samples (right).

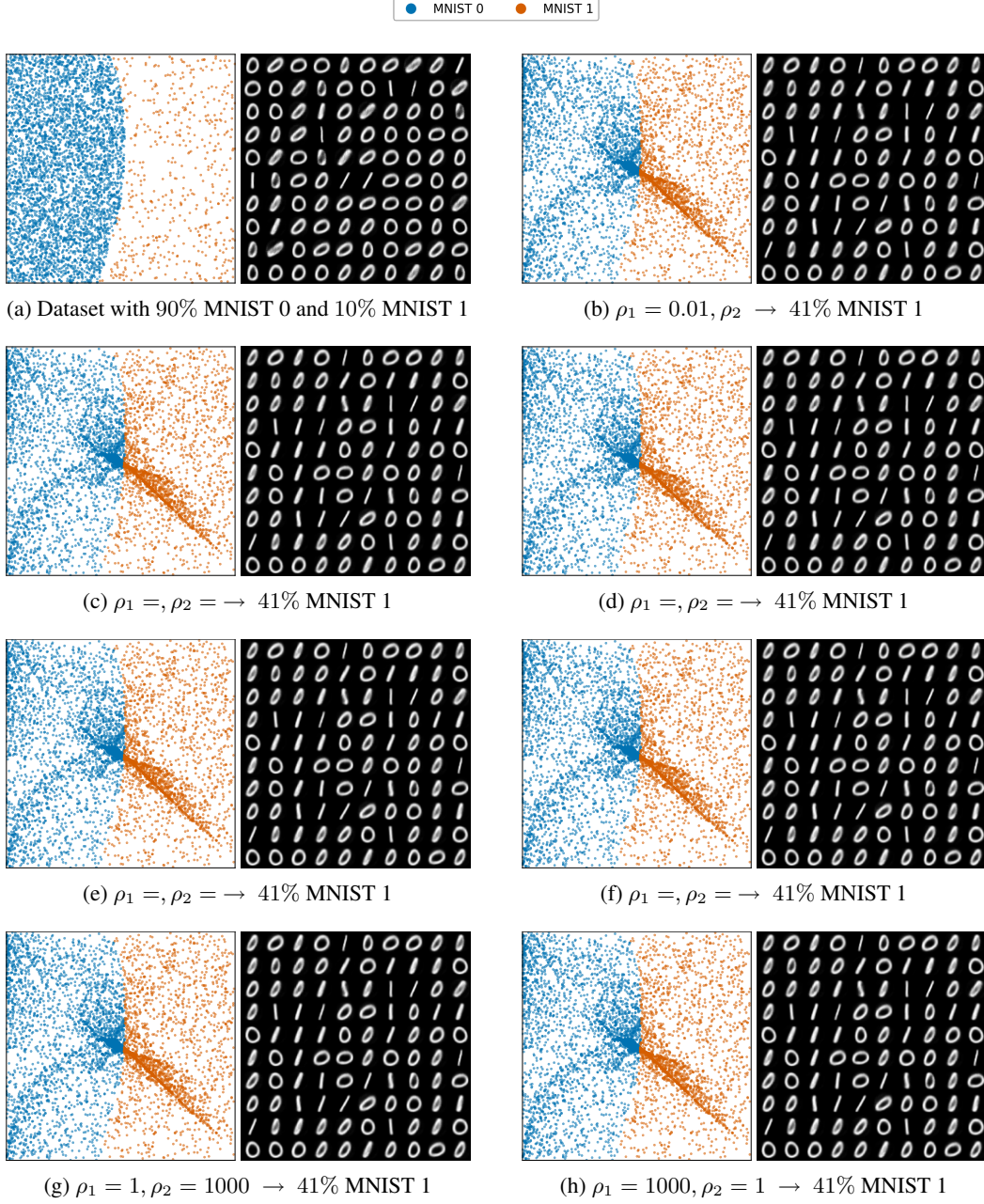


Figure 12: Ablation study of SUOT [7] for robust image generation. The figure illustrates the percentage of generated MNIST 1 digits (outliers), along with the corresponding learned latent distributions (left) and decoded image samples (right).

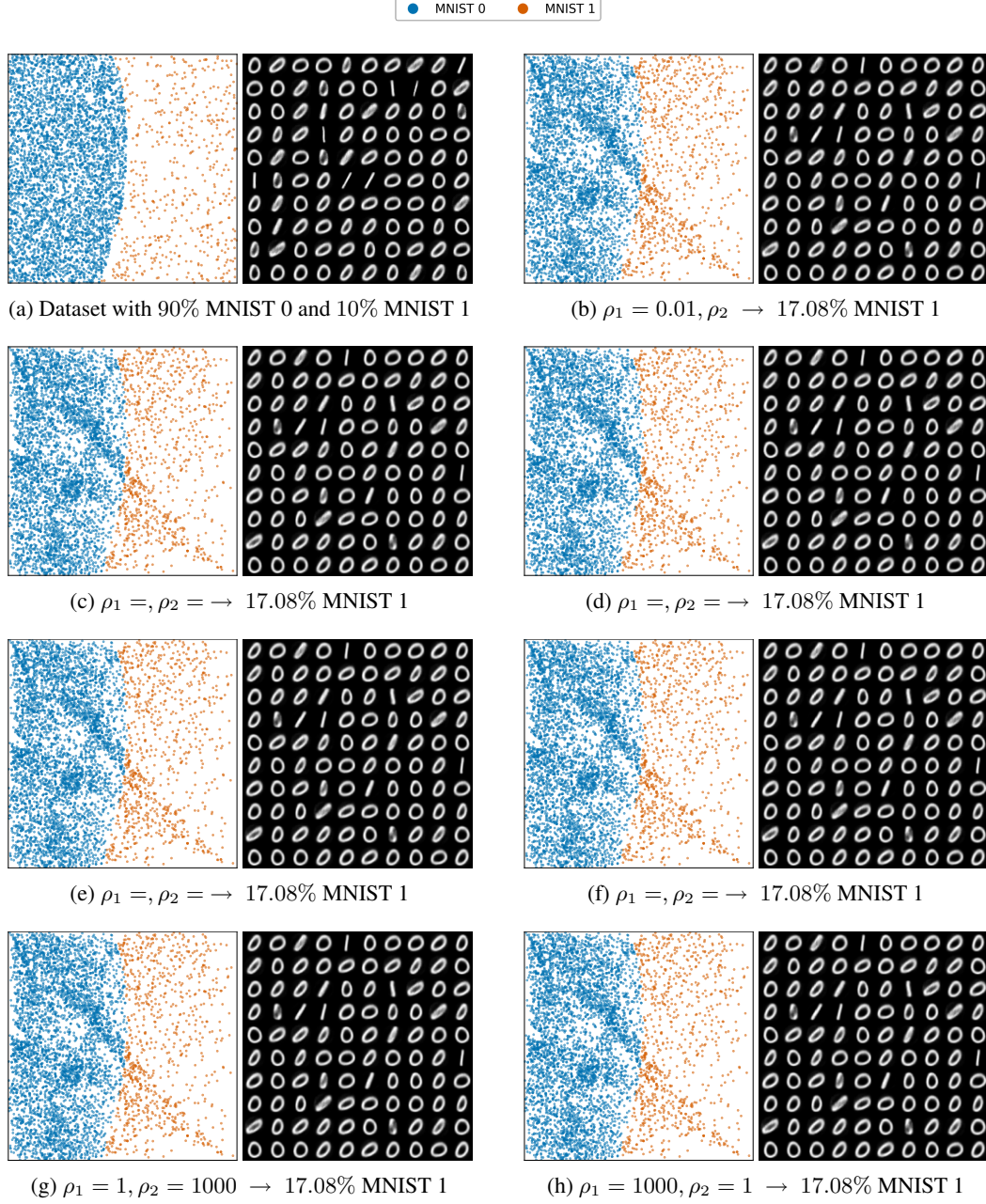


Figure 13: Ablation study of USOT [7] for robust image generation. The figure illustrates the percentage of generated MNIST 1 digits (outliers), along with the corresponding learned latent distributions (left) and decoded image samples (right).

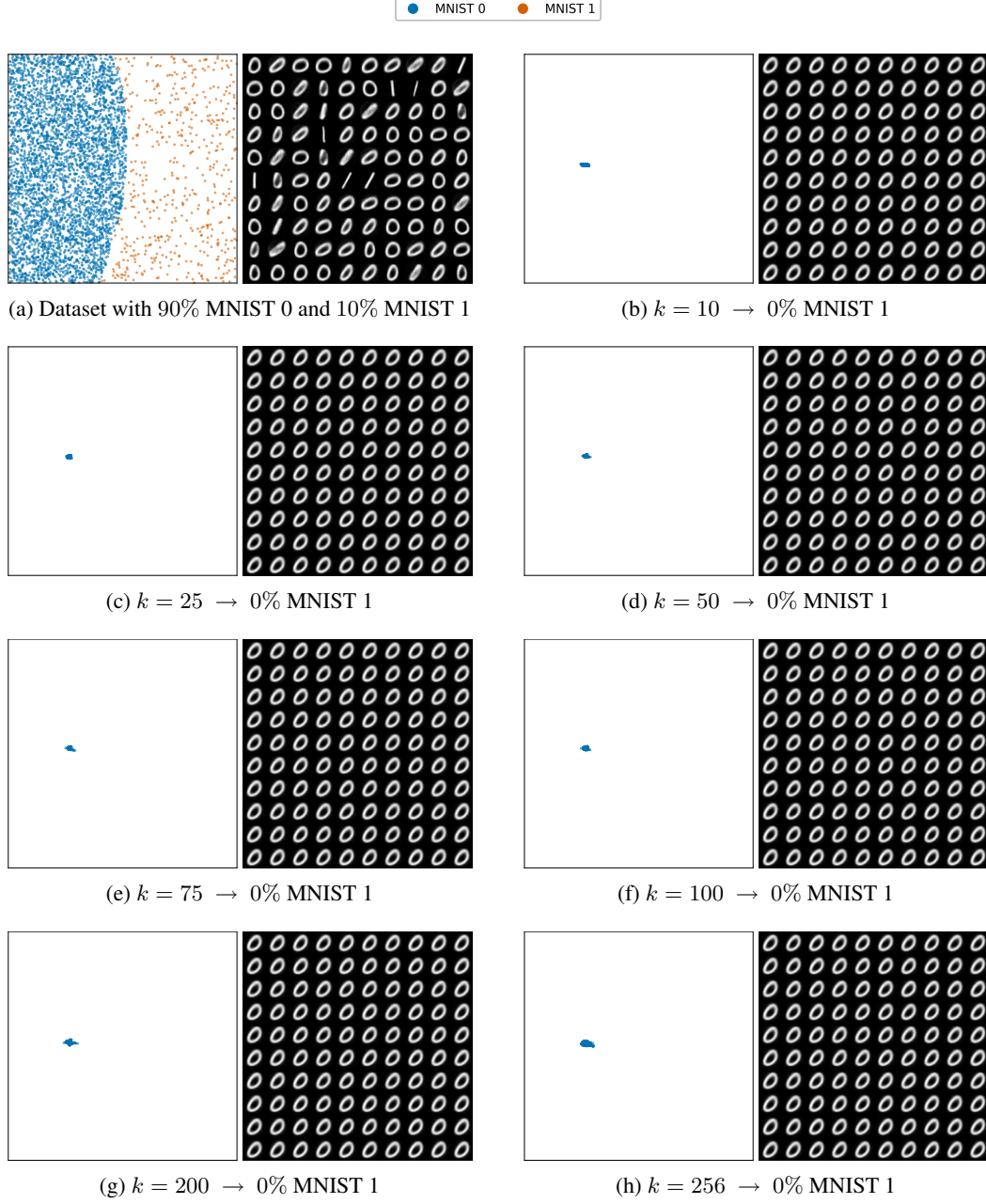


Figure 14: Ablation study of PAWL [13] for robust image generation. The figure illustrates the percentage of generated MNIST 1 digits (outliers), along with the corresponding learned latent distributions (left) and decoded image samples (right).

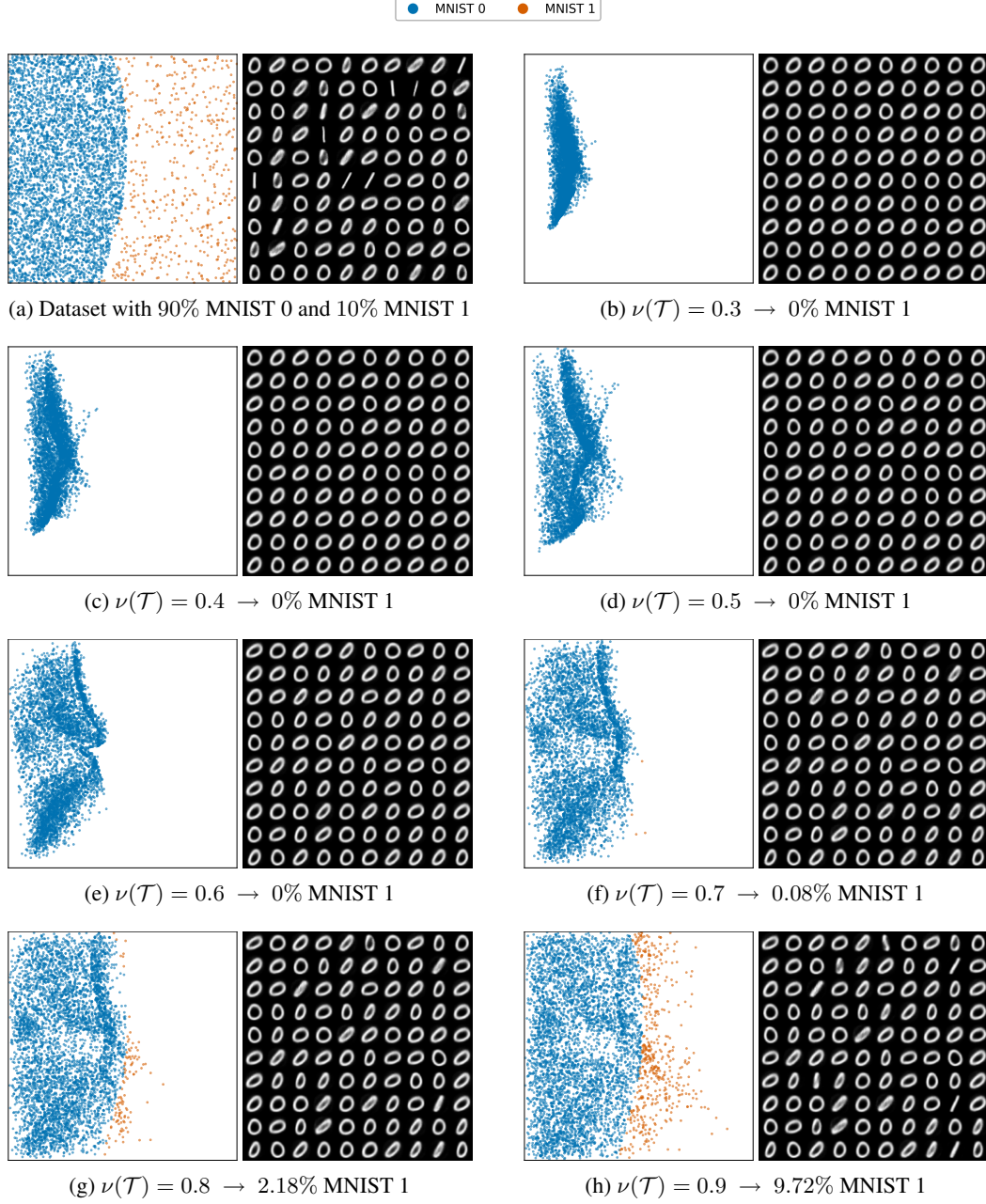


Figure 15: Ablation study of Partial-TSW (Ours) for robust image generation. The figure illustrates the percentage of generated MNIST 1 digits (outliers), along with the corresponding learned latent distributions (left) and decoded image samples (right).

1145 E.8 Imbalance Image to Image Translation

1146 This section outlines the experimental setup for the imbalanced image-to-image translation task,
1147 specifically converting “Young” faces to “Adult” faces.

1148 **Dataset.** Our experimental dataset and preprocessing follow [36]. We utilize the FFHQ dataset [32]
1149 of 1024×1024 images. These images are encoded into a 512-dimensional latent space using a
1150 pre-trained ALAE autoencoder [58]. The resulting latent representations are categorized into two
1151 classes: “Young” and “Adult”, based on a cutoff age of 45 years. This process yields an imbalanced
1152 dataset comprising approximately 38,000 “Young” latent vectors and 10,500 “Adult” latent vectors.
1153 The translation is performed within this 512-dimensional latent space.

1154 **Translation Accuracy.** To evaluate the accuracy of the translation from the “Young” to the “Adult”
1155 domain, we adapt the procedure from [26]. A classifier is pre-trained on the 512-dimensional latent
1156 vectors to distinguish between “Young” and “Adult” images. This pre-trained classifier, which
1157 achieves 99% accuracy on a held-out test set of latent vectors, is then used to assess whether
1158 the translated latent vectors $M(X)$ (where X are latents from the “Young” domain) are correctly
1159 classified as “Adult”.

1160 **Perceptual Similarity.** We measure the perceptual similarity between the original images (recon-
1161 structed from X) and the translated images (reconstructed from $M(X)$) using the Learned Perceptual
1162 Image Patch Similarity (LPIPS) metric [82]. For LPIPS calculations, we use the AlexNet backbone
1163 with pre-trained weights. While some prior work, such as [26], employed attribute-specific metrics
1164 like “Keep Accuracy” (e.g., for preserving gender), we selected LPIPS to offer a more comprehensive
1165 assessment of overall visual fidelity post-translation, rather than focusing on a single attribute.

1166 **UOT-FM Baseline.** We compare against Unbalanced Optimal Transport Flow Matching (UOT-
1167 FM) [21]. Following [26], we parameterize vector field v_θ using a 2-layer feed-forward network with
1168 512 hidden neurons and ReLU activation. We apply their default configuration for Flow Matching.
1169 Consistent with the approach in [26], we perform an ablation study over the regularization parameter
1170 λ , which controls the penalization of deviations from marginal constraints.

1171 **ULightOT Baseline.** We also include ULightOT [26] as a baseline. We adapted the publicly available
1172 code and default models for our experiments. Following the methodology in [26], we ablate the
1173 parameter τ , which governs the degree of mass conservation in the transport plan. Our empirical
1174 observations indicate that the performance of ULightOT saturates for $\tau > 1000$. For instance,
1175 increasing τ to 10000 yielded negligible changes in the Accuracy-LPIPS trade-off compared to
1176 $\tau = 1000$, as shown by the results (e.g., in Figure 5 of the main text).

1177 Mapping Network Architecture.

1178 For methods such as SW, Db-TSW, and our PartialTSW, the mapping network M is implemented us-
1179 ing a ResidualMLP. The input to this network is a latent vector $z \in \mathbb{R}^{512}$. The specific ResidualMLP
1180 configuration used has an input/output dimension of 512, with num_hidden_blocks=0 and
1181 hidden_dim_multiplier=1.

1182 The core of this network, denoted as MLP_{core} , processes the input z through the following sequence
1183 of operations:

- 1184 • Apply an initial linear transformation: `Linear(512, 512)`
- 1185 • Followed by layer normalization: `LayerNorm(512)`
- 1186 • Then, apply the GELU activation function: `GELU()`
- 1187 • Apply dropout with a rate of 0.1: `Dropout(0.1)`
- 1188 • Finally, apply an output linear projection: `Linear(512, 512)`

1189 Let the output of this sequential MLP_{core} block be $\text{MLP}_{\text{core}}(z)$.

The final output of the mapping network $M(z)$ is obtained by adding a scaled residual connection to
the original input:

$$M(z) = z + \alpha \cdot \text{MLP}_{\text{core}}(z)$$

1190 where α is a learnable scalar parameter (analogous to LayerScale) that is initialized to 0.1.

1191 **F Boarder Impacts**

1192 The introduction of the PartialTSW in this paper has a substantial societal impact by enhancing the
1193 precision and adaptability of optimal transport methods in various practical applications. This method
1194 can drive progress in numerous fields, such as healthcare, where better image processing techniques
1195 can aid in more accurate medical imaging diagnostics, or in the arts and entertainment industry,
1196 where enhanced generative models can lead to more sophisticated and creative outputs. Furthermore,
1197 the ability to handle dynamic settings efficiently opens new possibilities for real-time data analysis
1198 and decision-making in various sectors, including finance, logistics, and environmental monitoring.
1199 Ultimately, the method contributes to making advanced computational techniques more versatile and
1200 applicable to a broader range of real-world problems, thereby fostering innovation and improving
1201 societal well-being.

1202 NeurIPS Paper Checklist

1203 1. Claims

1204 Question: Do the main claims made in the abstract and introduction accurately reflect the
1205 paper's contributions and scope?

1206 Answer: [\[Yes\]](#)

1207 Justification: The main claims made in the abstract and introduction accurately reflect the
1208 paper's contributions and scope.

1209 Guidelines:

- 1210 • The answer NA means that the abstract and introduction do not include the claims
1211 made in the paper.
- 1212 • The abstract and/or introduction should clearly state the claims made, including the
1213 contributions made in the paper and important assumptions and limitations. A No or
1214 NA answer to this question will not be perceived well by the reviewers.
- 1215 • The claims made should match theoretical and experimental results, and reflect how
1216 much the results can be expected to generalize to other settings.
- 1217 • It is fine to include aspirational goals as motivation as long as it is clear that these goals
1218 are not attained by the paper.

1219 2. Limitations

1220 Question: Does the paper discuss the limitations of the work performed by the authors?

1221 Answer: [\[Yes\]](#)

1222 Justification: We have discussed the limitations of the work in the Conclusion.

1223 Guidelines:

- 1224 • The answer NA means that the paper has no limitation while the answer No means that
1225 the paper has limitations, but those are not discussed in the paper.
- 1226 • The authors are encouraged to create a separate "Limitations" section in their paper.
- 1227 • The paper should point out any strong assumptions and how robust the results are to
1228 violations of these assumptions (e.g., independence assumptions, noiseless settings,
1229 model well-specification, asymptotic approximations only holding locally). The authors
1230 should reflect on how these assumptions might be violated in practice and what the
1231 implications would be.
- 1232 • The authors should reflect on the scope of the claims made, e.g., if the approach was
1233 only tested on a few datasets or with a few runs. In general, empirical results often
1234 depend on implicit assumptions, which should be articulated.
- 1235 • The authors should reflect on the factors that influence the performance of the approach.
1236 For example, a facial recognition algorithm may perform poorly when image resolution
1237 is low or images are taken in low lighting. Or a speech-to-text system might not be
1238 used reliably to provide closed captions for online lectures because it fails to handle
1239 technical jargon.
- 1240 • The authors should discuss the computational efficiency of the proposed algorithms
1241 and how they scale with dataset size.
- 1242 • If applicable, the authors should discuss possible limitations of their approach to
1243 address problems of privacy and fairness.
- 1244 • While the authors might fear that complete honesty about limitations might be used by
1245 reviewers as grounds for rejection, a worse outcome might be that reviewers discover
1246 limitations that aren't acknowledged in the paper. The authors should use their best
1247 judgment and recognize that individual actions in favor of transparency play an impor-
1248 tant role in developing norms that preserve the integrity of the community. Reviewers
1249 will be specifically instructed to not penalize honesty concerning limitations.

1250 3. Theory assumptions and proofs

1251 Question: For each theoretical result, does the paper provide the full set of assumptions and
1252 a complete (and correct) proof?

1253 Answer: [\[Yes\]](#) .

Justification: We have provided full set of assumptions and complete proof for all theoretical results in Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have mentioned all information to reproduce the main experimental results in Appendix E. All necessary details for reproducing the main experimental results are documented. This includes comprehensive descriptions of network architectures, training procedures, and specific hyperparameters. Task-specific configurations are outlined in their respective subsections. The code is also included as supplementary material to further support reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

1308 some way (e.g., to registered users), but it should be possible for other researchers
1309 to have some path to reproducing or verifying the results.

1310 5. Open access to data and code

1311 Question: Does the paper provide open access to the data and code, with sufficient instruc-
1312 tions to faithfully reproduce the main experimental results, as described in supplemental
1313 material?

1314 Answer: [Yes]

1315 Justification: We have provided data and code in the supplemental material, with detailed
1316 instructions to reproduce the results.

1317 Guidelines:

- 1318 • The answer NA means that paper does not include experiments requiring code.
- 1319 • Please see the NeurIPS code and data submission guidelines ([https://nips.cc/](https://nips.cc/public/guides/CodeSubmissionPolicy)
1320 [public/guides/CodeSubmissionPolicy](https://nips.cc/public/guides/CodeSubmissionPolicy)) for more details.
- 1321 • While we encourage the release of code and data, we understand that this might not be
1322 possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not
1323 including code, unless this is central to the contribution (e.g., for a new open-source
1324 benchmark).
- 1325 • The instructions should contain the exact command and environment needed to run to
1326 reproduce the results. See the NeurIPS code and data submission guidelines ([https://](https://nips.cc/public/guides/CodeSubmissionPolicy)
1327 nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- 1328 • The authors should provide instructions on data access and preparation, including how
1329 to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- 1330 • The authors should provide scripts to reproduce all experimental results for the new
1331 proposed method and baselines. If only a subset of experiments are reproducible, they
1332 should state which ones are omitted from the script and why.
- 1333 • At submission time, to preserve anonymity, the authors should release anonymized
1334 versions (if applicable).
- 1335 • Providing as much information as possible in supplemental material (appended to the
1336 paper) is recommended, but including URLs to data and code is permitted.

1337 6. Experimental setting/details

1338 Question: Does the paper specify all the training and test details (e.g., data splits, hyper-
1339 parameters, how they were chosen, type of optimizer, etc.) necessary to understand the
1340 results?

1341 Answer: [Yes]

1342 Justification: We have provided all information about training and test details in Appendix
1343 E. We have included information on network architectures, optimizer types, batch sizes,
1344 number of training iterations, and the specific hyperparameters used in each experiment.
1345 In particular, Section E.6 describes the experimental setup for the point cloud experiments.
1346 Section E.7 focuses on the robust generative model, and Section E.8 outlines the details for
1347 the image translation task.

1348 Guidelines:

- 1349 • The answer NA means that the paper does not include experiments.
- 1350 • The experimental setting should be presented in the core of the paper to a level of detail
1351 that is necessary to appreciate the results and make sense of them.
- 1352 • The full details can be provided either with the code, in appendix, or as supplemental
1353 material.

1354 7. Experiment statistical significance

1355 Question: Does the paper report error bars suitably and correctly defined or other appropriate
1356 information about the statistical significance of the experiments?

1357 Answer: [Yes]

1358 Justification: Our experimental results are provided with error bars. We report statistical
1359 significance by including the mean and standard deviation over multiple runs in Table 2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We have provided sufficient information about computing resources needed to reproduce the experiments in Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We have discussed Broader impacts in Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All the assets used in the paper are properly cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The new assets provided in the paper are well documented, and the documentation is provided alongside the assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- 1517 • Depending on the country in which research is conducted, IRB approval (or equivalent)
1518 may be required for any human subjects research. If you obtained IRB approval, you
1519 should clearly state this in the paper.
- 1520 • We recognize that the procedures for this may vary significantly between institutions
1521 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
1522 guidelines for their institution.
- 1523 • For initial submissions, do not include any information that would break anonymity (if
1524 applicable), such as the institution conducting the review.

1525 16. **Declaration of LLM usage**

1526 Question: Does the paper describe the usage of LLMs if it is an important, original, or
1527 non-standard component of the core methods in this research? Note that if the LLM is used
1528 only for writing, editing, or formatting purposes and does not impact the core methodology,
1529 scientific rigorousness, or originality of the research, declaration is not required.

1530 Answer: [NA]

1531 Justification: The core method development in this research does not involve LLMs as any
1532 important, original, or non-standard components.

1533 Guidelines:

- 1534 • The answer NA means that the core method development in this research does not
1535 involve LLMs as any important, original, or non-standard components.
- 1536 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)
1537 for what should or should not be described.