PARETOTRANSPORT: GENERATIVE OPTIMIZATION BY MASS TRANSPORT TOWARD THE PARETO FRONT

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline multi-objective optimization (MOO) seeks candidate designs whose objective vectors not only lie close to the Pareto front but are also effectively distributed along it. Generative methods have recently emerged as a promising approach because they learn a distribution over feasible designs while allowing generation to be guided toward desirable designs. Existing methods, however, largely retain classical sample-wise guidance strategies, leaving the distribution-level modeling capability of generative methods underused. We formulate offline MOO as a sampling problem and motivate the 2-Wasserstein distance as a natural quality measure. Building on this view, we propose ParetoTransport, a training-free guidance method for pre-trained flow-matching models that formulates guidance directly over the population-level distribution in objective space, combining directional transport toward the Pareto front with Wasserstein matching that controls the allocation of probability mass along it. Under an exact-matching assumption, we prove that the transported distribution converges geometrically toward the Pareto front in 2-Wasserstein distance. On offline MOO benchmarks, we extend the evaluation of recent generative methods beyond hypervolume to generational distance, inverted generational distance, and the 2-Wasserstein distance. ParetoTransport achieves consistently strong performance, attaining the best average rank in convergence and probability-mass allocation across all benchmark families.

1 INTRODUCTION

Offline multi-objective optimization (MOO) seeks a set of Pareto-optimal designs that represent diverse trade-offs among multiple potentially conflicting objectives, while optimization relies only on an offline dataset of previously evaluated designs without access to further objective evaluations. This setting is particularly relevant when evaluating new designs is expensive or impractical, as in molecular, materials, and engineering design, where efficacy, toxicity, synthesizability, strength, conductivity, or cost must be balanced (Wang et al., 2021; Du et al., 2024; Pogue et al., 2023; Zeni et al., 2025; Chasparis et al., 2016).

Recently, diffusion and flow-matching models have emerged as promising approaches to offline MOO because they learn a distribution over feasible designs while allowing generation to be guided toward desirable designs (Yuan et al., 2025; Annadani et al., 2025). Training-free guidance does so at inference time, adapting a pre-trained generative model to task-specific objectives, which separates generative modeling from task-specific optimization and allows the model to be reused across tasks without retraining (Liu et al., 2023; Li et al., 2026). Since a generative model represents a distribution over designs, guiding it turns optimization into the problem of controlling the induced objective-space distribution: its displacement toward the Pareto front and its allocation of probability mass along it.

Although methods have been proposed to control displacement and mass allocation during generation, most of them control the induced objective-space distribution only indirectly. Conditional models incorporate the objective information during training and generate each sample conditioned on an objective target, with diversity obtained through the choice of targets (Shrestha et al., 2026) or through repulsive interactions between samples (Hotegni & Peitz, 2026). Training-free guidance instead uses objective information only at inference time, steering each sample of a pre-trained model through update directions based on scalarization weights (Yuan et al., 2025) or preference models

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

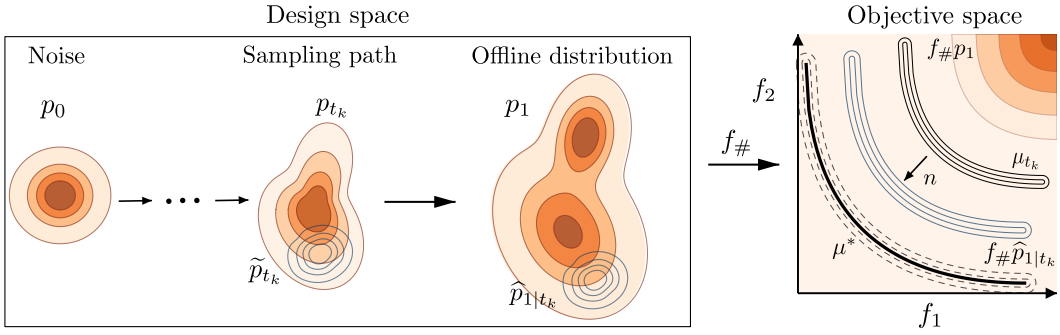


Figure 1: ParetoTransport guides the probability path $(p_{t_k})_{k=1}^K$ of a flow model from its learned offline distribution p_1 toward a guided distribution, whose induced objective-space distribution approximates a reference distribution μ^* on the Pareto front. At each step t_k, the estimated terminal objective-space distribution $f_\#\widehat{p}_{1|t_k}$ is moved toward the Pareto front through directional transport along n, while Wasserstein matching to the current proxy μ_{t_k} controls its mass allocation.

(Annadani et al., 2025). All of these methods have in common that they do not explicitly specify and refine the generated objective-space distribution, although such distribution-level control is a natural capability of generative modeling.

In this work, we introduce ParetoTransport, a training-free guidance method for pre-trained flow-matching models that explicitly specifies and refines the generated objective-space distribution. We start by formulating offline MOO as a sampling problem and approach it by mass transport in objective space: ParetoTransport iteratively moves the empirical offline distribution toward the Pareto front, combining directional transport toward the front with Wasserstein matching that controls the allocation of probability mass along it (Figure 1). Across standard offline MOO benchmarks, ParetoTransport achieves state-of-the-art performance among training-free generative methods and competitive performance relative to classical forward methods in terms of hypervolume (HV), generational distance (GD), inverted generational distance (IGD), and 2-Wasserstein distance (W_2).

Our contributions are summarized as follows:

- We formulate offline MOO as a sampling problem and motivate W_2 as a natural quality measure by relating it to the classical MOO metrics GD and IGD (Corollary 1).
- We propose ParetoTransport, a training-free guidance method for flow-matching models that formulates guidance directly over the induced distribution, combining Wasserstein matching with directional transport toward the Pareto front.
- For the idealized mass-transport procedure that alternates (i) Wasserstein matching and (ii) directional transport, we prove that the transported distribution converges geometrically toward the Pareto front in W_2 (Theorem 1).
- We demonstrate strong performance across 24 datasets, attaining the best average rank in convergence and probability-mass allocation across all benchmark families and extending recent generative evaluations beyond HV to GD, IGD, and W_2.

2 Problem: Offline Multi-Objective Optimization

The goal of multi-objective optimization (MOO) is to identify designs $x \in \mathcal{X} \subset \mathbb{R}^d$ that minimize multiple objective functions $f_i : \mathcal{X} \to \mathbb{R}$, $i = 1, \dots, m$. Since the objective functions may conflict, there is generally no unique minimizer, and MOO instead aims to identify the set of *Pareto-optimal* designs.

A design $x \in \mathcal{X}$ is said to *dominate* a design $x' \in \mathcal{X}$ if $f_i(x) \leq f_i(x')$ for all $i \in \{1, \dots, m\}$, with strict inequality for at least one i. A design $x \in \mathcal{X}$ is *Pareto-optimal* if no other design $x' \in \mathcal{X}$ dominates it. The set of all Pareto-optimal designs is called the *Pareto set*, and its image under $f := (f_1, \dots, f_m)$ is the *Pareto front*, denoted by $\mathcal{M}^*$.

Offline MOO considers scientific and engineering problems in which optimization relies only on an offline dataset[1] $(x_i, f(x_i))_{i=1}^N$ of previously evaluated designs, without access to further evaluations of the objective function f.

3 Background: Training-Free Guidance for Flow Matching

The goal of flow matching (FM) (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2022) is to learn a time-dependent velocity field $u : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ that transforms a simple *initial* distribution p_0 into a complex *terminal* distribution p_1. The flow map $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ is defined by $\phi_t(a) := x_t$, where $x : [0,1] \to \mathbb{R}^d$ is the solution of the ODE

$$\frac{dx_t}{dt} = u_t(x_t), \qquad x_0 = a. \tag{1}$$

This flow map induces a probability path $p_t := (\phi_t)_{\#} p_0$ that satisfies the continuity equation $\partial_t p_t + \nabla \cdot (u_t p_t) = 0$. The velocity field u_t is learned through a conditional probability path $p_{t|Z}$, where $Z = (X_0, X_1) \sim \pi$ and $\pi \in \Gamma(p_0, p_1)$. A standard choice is an affine conditional probability path $X_t := \alpha_t X_1 + \beta_t X_0 \sim p_{t|Z}$ with scheduler (α_t, β_t) satisfying $\alpha_0 = \beta_1 = 0, \alpha_1 = \beta_0 = 1$. This yields the simple conditional velocity field $u_{t|Z}(X_t) = \dot{\alpha}_t X_1 + \dot{\beta}_t X_0$. The resulting conditional regression objective has the same gradient as the marginal regression objective (Lipman et al., 2023), and therefore can be used to learn the marginal velocity field u_t. At inference time, sampling is performed by drawing an initial sample $x_0 \sim p_0$ and numerically solving the ODE in Eq. (1) with the learned velocity field u_t^θ, e.g., with the forward Euler method. Further details are provided in Appendix A.2.

Training-Free Guidance incorporates a task-specific loss $J : \mathbb{R}^d \to \mathbb{R}$ into the sampling of a pre-trained generative model at inference time (Liu et al., 2023; Ben-Hamu et al., 2024; Wang et al., 2025). Given an initial sample $x_0 \sim p_0$, the guided sampling process can be formulated as

$$\begin{aligned} \min_{(\widetilde{x}_t,\, c_t)_{t\in[0,1]}} \quad & J(\widetilde{x}_1) + \lambda \int_0^1 \frac{1}{2}\|c_t\|^2 \, dt \\ \text{s.t.} \quad & \widetilde{x}_0 = x_0, \quad \frac{d\widetilde{x}_t}{dt} = u_t(\widetilde{x}_t) + c_t \qquad t \in [0,1]. \end{aligned} \tag{2}$$

The optimization seeks a guidance signal $c : [0,1] \to \mathbb{R}^d$ that steers the induced trajectory $(\widetilde{x}_t)_{t\in[0,1]}$ toward a favorable terminal sample $\widetilde{x}_1$. The loss function J evaluates this terminal sample, while the second term penalizes large guidance-induced deviations from the pre-trained sampling dynamics.

Recent work (Li et al., 2026; Webber et al., 2026; Pourya et al., 2026) approximates the full-horizon optimization problem in Eq. (2) by a sequence of single-step subproblems. At each intermediate time t, a posterior mean estimator, denoted by $\widehat{x}_{1|t}$, estimates the terminal sample at which the terminal loss is evaluated, with its gradient propagated back to the intermediate sample. Li et al. (2026) and Pourya et al. (2026) additionally derive a local mapping from terminal space back to time t, which we denote by $\widehat{x}_{t|1}$. This allows each subproblem to be optimized directly in terminal space, avoiding differentiation through the terminal estimator $\widehat{x}_{1|t}$ and hence through the learned model u_t^θ. Further details are provided in Appendix A.2.

4 Offline MOO as Sampling Problem

In this work, we view the MOO goal of identifying Pareto-optimal designs (see Section 2) as a statistical sampling problem. Classically, MOO algorithms aim at solution sets that achieve both convergence toward the Pareto front and uniform coverage along it (Chen et al., 2011).

Viewing a solution set as a statistical sample rather than a set refines this goal: convergence becomes concentration of the samples' distribution near the Pareto front, and coverage becomes allocation of probability mass along it. Accordingly, *the goal of MOO algorithms becomes to find a design-space*

[1] Following standard learning-theoretic assumptions (Mohri et al., 2018; Shalev-Shwartz & Ben-David, 2014), we regard the designs $x_1, \ldots, x_N$ as independent and identically distributed samples from an unknown data distribution p_1 on $\mathcal{X}$.

distribution p whose induced objective-space distribution $f_{\#}p$ approximates a reference distribution μ^* on the (unknown) Pareto front $\mathcal{M}^*$, e.g., the uniform distribution.[2]

4.1 A Quality Measure: The Wasserstein Distance

This sampling view of MOO naturally motivates the 2-Wasserstein distance W_2 as a quality measure for generative methods, where $W_2^2(\mu,\nu) := \inf_{\gamma\in\Gamma(\mu,\nu)} \int \|x-y\|^2 \, d\gamma(x,y)$ and $\Gamma(\mu,\nu)$ is the set of couplings of $\mu,\nu \in \mathcal{P}_2(\mathbb{R}^m)$. Indeed, W_2 is related to the two well-established MOO metrics GD and IGD[3] in the following sense. For a generative method that learned to sample from a distribution p with objective-space distribution $\mu := f_{\#}p$, the value $W_2(\mu,\mu^*)$ upper bounds the continuous versions of GD and IGD,

$$\mathrm{GD}(\mu\|\mu^*) := \left(\int d(y, \operatorname{supp}\mu^*)^2 \, d\mu(y)\right)^{1/2}, \qquad \mathrm{IGD}(\mu\|\mu^*) := \mathrm{GD}(\mu^*\|\mu),$$

where $d(y,A) := \inf_{a\in A}\|y-a\|$ is the distance from y to the set A.

Corollary 1. *For $\mu,\nu \in \mathcal{P}_2(\mathbb{R}^m)$, $\mathrm{GD}(\mu\|\nu) \le W_2(\mu,\nu)$ and $\mathrm{IGD}(\mu\|\nu) \le W_2(\mu,\nu)$.*

The proof of our Corollary 1 is provided in Appendix A.5.

5 Our Method: Offline MOO by Mass Transport

The formulation of MOO as a sampling problem in Section 4 motivates us to approach offline MOO by mass transport, which leads to our novel algorithm: *ParetoTransport*.

5.1 Approach by Mass Transport

Denote by $D(\mathrm{best})$ the offline front, that is, the set of objective vectors of all non-dominated designs in the offline dataset, and denote by $\mu_0 \in \mathcal{P}_2(\mathbb{R}^m)$ the empirical distribution over the offline front, that is, $\mu_0 := \frac{1}{|D(\mathrm{best})|}\sum_{y\in D(\mathrm{best})}\delta_y$. $D(\mathrm{best})$ retains the best observed trade-offs and their structure in objective space, and thus provides a natural initialization for the transport toward the unknown Pareto front (see Figure 2).

Following the sampling view of MOO (Section 4), our approach is to iteratively transport μ_0 toward its projection onto the Pareto front in terms of the 2-Wasserstein distance in objective space.

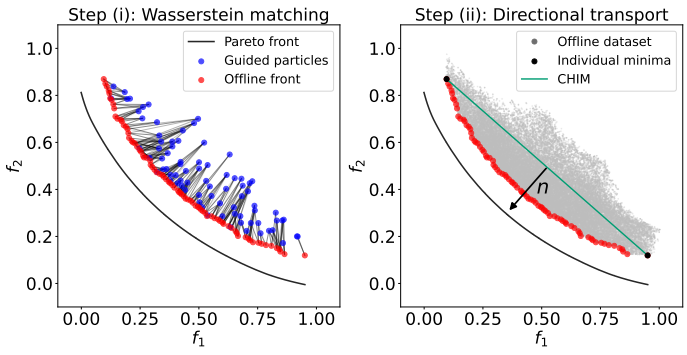


Figure 2: **Step (i):** Wasserstein matching to control probability-mass allocation along the Pareto front. **Step (ii):** Directional transport toward the front, using the CHIM normal n (RE21 task).

[2] Uniform is understood with respect to the normalized Hausdorff measure on $\mathcal{M}^*$ (Remark 3); throughout, $\mu^* \in \mathcal{P}_2(\mathbb{R}^m)$ with $\operatorname{supp}\mu^* = \mathcal{M}^*$.

[3] The squared GD (Van Veldhuizen & Lamont, 2000) between a solution set A and a finite approximation Z of the Pareto front is defined by $\mathrm{GD}^2(A,Z) := \frac{1}{|A|}\sum_{a\in A} d(a,Z)^2$. The squared IGD (Coello Coello & Reyes Sierra, 2004) is defined by $\mathrm{IGD}^2(A,Z) := \mathrm{GD}^2(Z,A)$.

Starting from μ_0, we introduce a sequence $(\mu_k)_{k\geq 0}$ of intermediate proxy distributions by alternating between two steps:

$$\mu_k \xrightarrow{\text{step (i): } W_2 \text{ matching}} q_{k+1} \xrightarrow{\text{step (ii): directional transport}} \mu_{k+1}.$$

Step (i) aims to find a design-space distribution[4]

$$q_{k+1} \in \operatorname*{argmin}_{q\in\mathcal{Q}} W_2^2(f_{\#}q, \mu_k), \tag{3}$$

whose induced objective-space distribution $f_{\#}q_{k+1}$ matches the current proxy μ_k. Step (ii) moves $f_{\#}q_{k+1}$ toward the Pareto front and defines the next intermediate proxy $\mu_{k+1} := (T_\eta \circ f)_{\#}q_{k+1}$ by the map

$$T_\eta(y) := y + \eta\big(\Pi_{\mathcal{M}^*}(y) - y\big), \tag{4}$$

where $\eta \in (0, 1]$ and Π_{M^*} denotes the projection onto $\mathcal{M}^*$. Thus, T_η moves each objective vector by a fraction η of the displacement toward its closest point on the Pareto front.

The two-step procedure described above solves the MOO problem in the following sense, where we use $\bar{\mu} := (\Pi_{\mathcal{M}^*})_{\#}\mu_0$ to denote the pushforward of μ_0 under the projection $\Pi_{\mathcal{M}^*}$.

Theorem 1. *If Step (i) achieves exact matching, $W_2(f_{\#}q_{k+1}, \mu_k) = 0$ for all k, and the technical Assumptions (1)–(3) in Appendix A.5 hold, then $W_2(\mu_k, \bar{\mu}) = (1-\eta)^k\, W_2(\mu_0, \bar{\mu})$, and consequently*

$$W_2(\mu_k, \mu^*) \leq (1-\eta)^k W_2(\mu_0, \bar{\mu}) + W_2(\bar{\mu}, \mu^*). \tag{5}$$

The proof is provided in Appendix A.5. Theorem 1 provides a quality guarantee for our iterative procedure. Assume that the offline front distribution is sufficiently representative in the sense that its projection $\bar{\mu}$ onto the Pareto front satisfies $W_2(\bar{\mu}, \mu^*) < \epsilon$ for some small ϵ. Then, Theorem 1 guarantees $\lim_{k\to\infty} W_2(\mu_k, \mu^*) < \epsilon$. Moreover, the first term in Eq. (5) decreases exponentially with k. In the terminology of Holly et al. (2026), $W_2^2(\mu_0, \bar{\mu}) = \mathrm{GD}^2(\mu_0 \| \mu^*)$ is the offline-frontier shift (Remark 4).

5.2 Our ParetoTransport Algorithm

We introduce *ParetoTransport*, a training-free guidance method for flow-matching models that lifts guidance from individual sampling trajectories to the induced probability path. Existing training-free guidance methods for offline MOO use sample-wise guidance objectives, based on scalarization weights (Yuan et al., 2025) or preference models (Annadani et al., 2025). In contrast, ParetoTransport formulates guidance directly over the probability path, combining Wasserstein matching with directional transport.

The mass-transport procedure of Section 5.1 motivates the design of our distribution-level guidance loss. To apply such a loss during sampling, we first formulate the guided sampling problem in Eq. (2) over the probability path:

$$\begin{aligned} \min_{(\widetilde{p}_t,\, c_t)_{t\in[0,1]}} \quad & J(\widetilde{p}_1) + \lambda \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2}\|c_t(x)\|^2 \, d\widetilde{p}_t(x)\, dt \\ \text{s.t.} \quad & \widetilde{p}_0 = p_0, \quad \partial_t \widetilde{p}_t + \nabla \cdot \big((u_t + c_t)\widetilde{p}_t\big) = 0, \qquad t \in [0,1]. \end{aligned} \tag{6}$$

The optimization seeks a velocity field $c_t : \mathbb{R}^d \to \mathbb{R}^d$ that steers the induced probability path $(\widetilde{p}_t)_{t\in[0,1]}$ toward a favorable terminal distribution $\widetilde{p}_1$. The functional $J : \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}$ evaluates this terminal distribution, while the regularization term penalizes large deviations from the pre-trained sampling dynamics.

In practice, we discretize the probability path on a grid $0 = t_1 < \ldots < t_K = 1$, and use forward Euler. We introduce a guidance onset time t_{guided}: steps with $t_k \leq t_{\text{guided}}$ consist of plain Euler updates, while subsequent steps incorporate guidance. Each intermediate distribution $\widetilde{p}_{t_k}$ is represented by a particle system $(\widetilde{x}^i_{t_k})_{i=1}^N$, approximating the probability path by a set of particle

[4] We assume that q is chosen from a set $\mathcal{Q}$ of admissible design-space distributions for which Eq. (3) admits a minimizer, e.g., $\mathcal{Q} \subseteq \mathcal{P}_2(\mathcal{X})$ is nonempty and weakly closed, with compact $\mathcal{X} \subseteq \mathbb{R}^d$ and continuous $f : \mathcal{X} \to \mathbb{R}^m$ (Villani, 2009).

Algorithm 1 ParetoTransport: Mass transport toward the Pareto front

1: Sample $x_0^i \sim p_0$ and initialize $\widetilde{x}_{t_1}^i := x_0^i, \quad i = 1, \dots, N$
2: Initialize proxy $\mu_{t_{k'}} := \mu_0$, where $k' := \min\{k : t_k > t_{\text{guided}}\}$
3: **for** $k = 1$ to $K - 1$ **do**
4: Take one Euler step: $x_{t_{k+1}}^i := \widetilde{x}_{t_k}^i + \Delta t_k u_{t_k}^\theta(\widetilde{x}_{t_k}^i), \quad i = 1, \dots, N$
5: **if** $k < k'$ **then**
6: $\widetilde{x}_{t_{k+1}}^i := x_{t_{k+1}}^i, \qquad i = 1, \dots, N$
7: **continue**
8: **end if**
9: Estimate terminal particles: $\widehat{x}_1^i := \widehat{x}_{1|t_{k+1}}(x_{t_{k+1}}^i) = \dfrac{\beta_{t_{k+1}} u_{t_{k+1}}^\theta(x_{t_{k+1}}^i) - \dot{\beta}_{t_{k+1}} x_{t_{k+1}}^i}{\dot{\alpha}_{t_{k+1}} \beta_{t_{k+1}} - \alpha_{t_{k+1}} \dot{\beta}_{t_{k+1}}}$
10: Solve the optimization problem in terminal space:

$$\big(x_{\text{opt}}^i\big)_{i=1}^N \in \arg\min_{(x_1^i)_{i=1}^N} J_k\big(\frac{1}{N}\sum_{i=1}^N \delta_{x_1^i}\big) + \frac{\lambda}{N}\sum_{i=1}^N \frac{\alpha_{t_{k+1}}^2}{2\Delta t_k}\|x_1^i - \widehat{x}_1^i\|^2,$$

11: Update proxy: $\mu_{t_{k+1}} := \mathcal{R}\big(\mu_{t_k}, \big(f(x_{\text{opt}}^i)\big)_{i=1}^N\big)$
12: Sample $\varepsilon_k^i \sim p_0$ and map back $\widetilde{x}_{t_{k+1}}^i := \widehat{x}_{t_{k+1}|1}(x_{\text{opt}}^i; \varepsilon_k^i) = \alpha_{t_{k+1}} x_{\text{opt}}^i + \beta_{t_{k+1}} \varepsilon_k^i$
13: **end for**
14: **return** Guided terminal particles $(\widetilde{x}_1^i)_{i=1}^N$

trajectories. The particle trajectories are coupled through the loss functional. For a fixed loss functional J, we apply the single-step approximation of Li et al. (2026) to the particle formulation of Eq. (6), replacing the full-horizon optimization problem by a sequence of single-step subproblems; the regularization term is expressed in terms of the terminal estimates $\widehat{x}_1^i$ (Algorithm 1, line 10). Its derivation is provided in Appendix A.3.1.

ParetoTransport adapts this construction by updating the functional between guided sampling steps. We realize the matching and directional transport steps of the mass-transport procedure jointly through design-space optimization rather than by explicitly transporting the proxy in objective space. This avoids constructing increasingly optimistic objective-space vectors that need not correspond to feasible designs. At each step k, we hold the current proxy μ_{t_k} fixed throughout the optimization and use the loss functional

$$J_k(q) := W_2^2(f_\# q, \mu_{t_k}) + \gamma L_n(f_\# q, \mu_{t_k}), \quad L_n(\nu, \mu) := -\left(\mathbb{E}_{Y\sim\nu}[Y] - \mathbb{E}_{Y\sim\mu}[Y]\right)^\top n, \quad (7)$$

where q denotes the terminal design-space distribution and $n \in \mathbb{R}^m$ is the CHIM (Convex Hull of Individual Minima) normal (Das & Dennis, 1998), used as a global approximation for the unknown projection direction toward the Pareto front.[5] The Wasserstein term controls the probability-mass allocation relative to the current proxy, while the directional term encourages transport along n. We update the proxy by combining the optimized objective vectors with the support of the current proxy to retain information from previous steps. The proxy-update operator $\mathcal{R}$ selects the new support and assigns crowding-based probability masses, giving more mass to sparsely represented regions and thereby empirically promoting mass redistribution along the front. Further details are provided in Appendix A.3. The pseudocode is provided in Algorithm 1. We use the linear scheduler $(\alpha_t, \beta_t) := (t, 1 - t)$, for which Corollary 2 gives the corresponding posterior-mean estimator $\widehat{x}_{1|t}$. Remark 1 defines the local reverse parameterization $\widehat{x}_{t|1}$.

6 Related Work

Forward methods for offline optimization learn surrogate models on the offline dataset and optimize them in design space as proxies for the unknown objectives (Xue et al., 2024; Kim et al., 2026). Since the search exploits surrogate errors beyond the data, single-objective methods regularize the surrogate against over-optimization (Trabucco et al., 2021; Qi et al., 2022; Yuan et al., 2023; Yu

[5] If the individual-minimum objective vectors are affinely independent, then the CHIM spans an $(m-1)$-dimensional affine hyperplane and therefore has a unique normal direction up to sign. Further details are provided in Appendix A.3.3.

et al., 2021; Chen et al., 2023); Off-MOO-Bench (Xue et al., 2024) carries these surrogates over to multiple objectives and searches them with evolutionary algorithms (Deb et al., 2002a).

Generative methods form an inverse paradigm, producing designs from objective information. In conditional models, objective information enters during training (Kumar & Levine, 2020; Shrestha et al., 2026; Hotegni & Peitz, 2026); in training-free guidance, an unconditionally trained model is steered only at inference time, which separates generative modeling from task-specific optimization and allows the same model to be reused across tasks (Yuan et al., 2025; Annadani et al., 2025). All of these methods determine the objective-space distribution of the generated population only implicitly, through conditioning targets, scalarization weights, repulsive interactions, or preference labels. ParetoTransport belongs to the training-free family and instead specifies and refines this distribution explicitly. An extended discussion is provided in Appendix A.7.

7 Experiments

We evaluate our method on standard offline MOO benchmarks (Xue et al., 2024), ranging from synthetic tasks (Zitzler et al., 2000; Deb et al., 2002b) to real-world engineering tasks (Tanabe & Ishibuchi, 2020). Each task provides an offline dataset of 60k designs and their corresponding objective vectors. The synthetic benchmark suite contains 12 tasks widely used in MOO, including the ZDT (Zitzler et al., 2000) tasks with 2 objectives and the DTLZ (Deb et al., 2002b) tasks with 3 objectives. The objective functions are analytical, and their Pareto fronts are known in closed form. The RE benchmark suite (Tanabe & Ishibuchi, 2020) contains 15 tasks based on real-world engineering design problems, including bar truss, welded beam, and disc brake design, of which we use the 12 with $m \leq 3$ objectives. For evaluation, we use Pareto front approximations provided with the RE benchmark. Further details on the benchmarks are provided in Appendix A.4.3.

Baselines. We compare ParetoTransport against state-of-the-art offline MOO methods. The forward baselines follow Off-MOO-Bench (Xue et al., 2024) and search learned surrogates with NSGA-II (Deb et al., 2002a), using either one multi-head surrogate trained with the multi-task learning methods GradNorm (Chen et al., 2018) or PcGrad (Yu et al., 2020) (*Multi-Head*), or one surrogate per objective trained with COM (Trabucco et al., 2021), IOM (Qi et al., 2022), ICT (Yuan et al., 2023), RoMA (Yu et al., 2021), or TriMentoring (Chen et al., 2023) (*Multiple-Models*); in both cases, Vanilla denotes the surrogate trained without any of these methods. The inverse baselines are the two training-free guidance methods for offline MOO: ParetoFlow (Yuan et al., 2025) guides a pre-trained flow-matching model with surrogate gradients under scalarization weights, and PGD (Annadani et al., 2025) guides a pre-trained diffusion model with a preference model that predicts Pareto dominance.

Implementation. ParetoTransport uses a flow-matching model based on the architecture of ParetoFlow (Yuan et al., 2025) and one surrogate per objective trained with the Vanilla Multiple-Models configuration of Off-MOO-Bench (Xue et al., 2024). We sample $N = 256$ designs with $K = 100$ Euler steps, start guidance at $t_{\text{guided}} = 0.8$, and run 10 Adam steps per guided step with $\gamma = 0.5$ and $\lambda = 10^{-5}$; further training and sampling details are given in Appendix A.4.1.

Evaluation. We evaluate all methods using HV, GD, IGD, and W_2. HV measures the dominated objective-space volume, while GD and IGD provide complementary set-based measures of convergence toward the Pareto front and coverage along it. We further report W_2 between the generated objective-space distribution and the empirical uniform distribution of the Pareto front reference set provided with each benchmark. Unlike the set-based metrics, W_2 additionally accounts for probability-mass allocation along the front. Holly et al. (2026) also evaluate generative offline MOO beyond HV, using GD and IGD; to our knowledge, this is the first offline MOO evaluation to use W_2. Further details on the metrics are provided in Appendix A.4.2.

7.1 Results

We evaluate each method over 5 random seeds using a population of $N = 256$ candidate designs. All metrics are computed in the normalized objective space using the fixed min–max statistics of the offline dataset. Figure 3 visualizes guided sampling on ZDT1: from $t_{\text{guided}} = 0.8$ on, the terminal estimates move from the offline dataset toward the Pareto front while the proxy is updated to control their allocation along it.

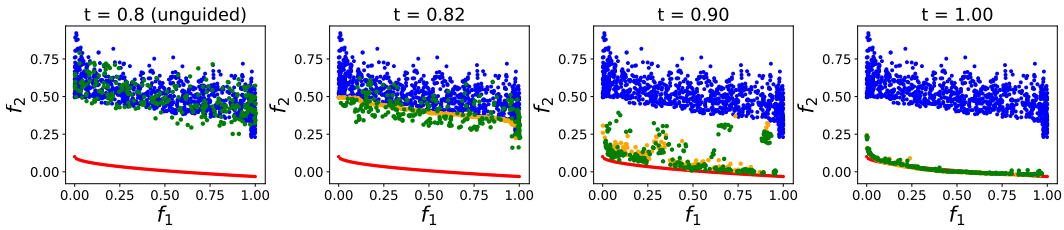


Figure 3: ParetoTransport on ZDT1. Blue points show the offline dataset in objective space, while green points show the objective vectors of terminal estimates $\widehat{x}_{1|t}$. For $t > 0.8$, ParetoTransport guides sampling toward the Pareto front (red) and iteratively updates the proxy (orange) to guide allocation along the front.

Table 1 reports HV on a representative subset of tasks across all three benchmark families. ParetoTransport remains competitive in terms of HV with both forward and inverse methods. However, HV aggregates convergence and coverage into a single number and is strongly influenced by individual extreme solutions. We therefore additionally evaluate GD, IGD, and W_2, which assess convergence, coverage, and probability-mass allocation separately.

Table 2 reports the average rank across tasks within each benchmark family for HV, GD, IGD, and W_2. ParetoTransport achieves the best average rank on all four metrics for both the ZDT and DTLZ tasks. On the RE tasks, it remains competitive, achieving the best average rank in GD and W_2, and the second-best rank in IGD. Most notably, ParetoTransport attains the best average rank in terms of convergence (GD) and probability-mass allocation (W_2) across all three benchmark families, supporting our approach of coupling transport toward the Pareto front with Wasserstein matching to control allocation along it. Complete per-task results are provided in Appendix A.6.

Table 1: HV results (↑). Bold = best per column, underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	ZDT2	ZDT6	DTLZ1	DTLZ6	DTLZ7	RE21	RE22
D(best)	4.677	4.608	10.614	10.671	8.992	4.176	4.806
MultiHead-Vanilla	5.612 ± 0.098	4.774 ± 0.003	10.643 ± 0.002	10.900 ± 0.019	10.715 ± 0.027	**4.598 ± 0.000**	**4.840 ± 0.000**
MultiHead-PcGrad	5.605 ± 0.047	4.776 ± 0.013	10.644 ± 0.001	10.918 ± 0.011	10.598 ± 0.028	**4.598 ± 0.001**	4.828 ± 0.023
MultiHead-GradNorm	5.483 ± 0.115	4.709 ± 0.025	10.537 ± 0.239	10.848 ± 0.060	9.654 ± 0.735	4.120 ± 0.606	4.721 ± 0.218
MultipleModels-Vanilla	5.600 ± 0.043	4.774 ± 0.004	10.644 ± 0.001	10.928 ± 0.008	10.535 ± 0.079	**4.598 ± 0.001**	**4.840 ± 0.000**
MultipleModels-COM	5.093 ± 0.131	3.435 ± 0.750	10.642 ± 0.002	10.786 ± 0.017	9.809 ± 0.273	4.238 ± 0.012	4.809 ± 0.012
MultipleModels-IOM	5.530 ± 0.101	4.777 ± 0.005	10.643 ± 0.002	10.886 ± 0.008	10.544 ± 0.126	4.594 ± 0.007	**4.840 ± 0.000**
MultipleModels-ICT	5.428 ± 0.004	4.723 ± 0.051	10.644 ± 0.001	10.781 ± 0.022	10.342 ± 0.092	**4.598 ± 0.001**	**4.840 ± 0.000**
MultipleModels-RoMA	5.496 ± 0.067	2.597 ± 0.027	10.641 ± 0.002	10.742 ± 0.015	10.596 ± 0.036	4.566 ± 0.003	4.543 ± 0.664
MultipleModels-Trimentoring	5.548 ± 0.041	4.045 ± 0.122	10.642 ± 0.007	10.778 ± 0.049	10.392 ± 0.423	4.595 ± 0.001	**4.840 ± 0.000**
ParetoFlow	4.695 ± 0.071	4.683 ± 0.068	10.626 ± 0.006	10.686 ± 0.039	8.875 ± 0.099	4.133 ± 0.093	4.815 ± 0.028
PGD	5.367 ± 0.064	**4.823 ± 0.004**	10.645 ± 0.001	10.336 ± 0.204	9.767 ± 0.141	4.479 ± 0.046	4.836 ± 0.003
ParetoTransport (ours)	**5.654 ± 0.022**	4.822 ± 0.007	**10.647 ± 0.000**	**11.045 ± 0.009**	**10.749 ± 0.062**	4.577 ± 0.007	**4.840 ± 0.000**

Table 2: Average rank per metric across benchmark families (lower is better). Bold = best per column; underline = second best. Each method is run for 5 seeds and evaluated on 256 designs.

	ZDT				DTLZ				RE			
	HV	GD	IGD	W2	HV	GD	IGD	W2	HV	GD	IGD	W2
MultiHead-Vanilla	4.100	4.000	5.400	4.000	3.714	5.929	5.214	5.714	4.833	6.375	5.833	5.667
MultiHead-PcGrad	6.200	5.400	5.400	4.200	3.071	4.214	4.571	4.286	7.000	7.250	7.333	7.000
MultiHead-GradNorm	8.100	7.100	8.400	7.000	9.000	5.357	7.714	5.714	11.500	7.917	10.375	9.042
MultipleModels-Vanilla	5.900	3.800	5.600	6.400	4.357	4.143	5.714	4.286	**3.417**	5.708	5.333	5.500
MultipleModels-COM	8.800	7.800	8.200	7.600	7.643	8.571	7.143	7.714	8.750	5.292	7.250	5.917
MultipleModels-IOM	4.000	5.400	4.200	4.200	5.214	5.714	5.357	5.429	3.708	6.500	**3.375**	5.000
MultipleModels-ICT	7.600	6.800	7.800	7.600	7.143	9.286	7.643	9.286	4.417	5.292	5.875	5.750
MultipleModels-RoMA	7.000	7.400	7.600	7.400	8.714	10.857	9.786	10.000	7.667	9.250	7.542	8.583
MultipleModels-Trimentoring	8.000	9.500	7.600	9.200	9.071	6.714	7.286	6.714	5.708	6.083	7.083	6.042
ParetoFlow	9.600	8.000	9.600	9.400	9.857	8.429	8.571	8.714	9.333	6.042	6.917	6.500
PGD	6.800	9.200	6.200	7.800	8.571	7.214	7.000	8.571	7.083	8.417	6.000	8.208
ParetoTransport (ours)	**1.900**	**3.600**	**2.000**	**3.200**	**1.643**	**1.571**	**2.000**	**1.571**	4.583	**3.875**	5.083	**4.792**

Table 3: Ablation of Wasserstein term $W_2(\cdot, \mu_k)$ and directional transport term L_n. Results are averaged over all tasks within each benchmark family and reported as changes Δ relative to the offline front D(best); bold = best per column.

			ZDT				DTLZ				RE			
	$W_2(\cdot, \mu_k)$	L_n	GD (↓)	IGD (↓)	W_2 (↓)	HV (↑)	GD (↓)	IGD (↓)	W_2 (↓)	HV (↑)	GD (↓)	IGD (↓)	W_2 (↓)	HV (↑)
D(best)			0.262	0.261	0.374	4.812	0.373	0.247	0.427	11.542	0.075	0.123	0.226	7.368
		✓	**-0.139**	-0.057	+0.026	+0.423	-0.151	-0.123	-0.101	+0.285	+0.006	-0.014	+0.074	+0.256
Δ	✓		-0.060	-0.089	-0.066	+0.423	-0.065	-0.120	-0.064	+0.286	-0.001	-0.031	-0.013	+0.342
	✓	✓	-0.121	**-0.150**	**-0.096**	**+0.465**	**-0.179**	**-0.142**	**-0.174**	**+0.337**	**-0.026**	**-0.056**	**-0.035**	**+0.410**

7.2 Ablations

We study the contributions of the Wasserstein and directional transport terms in Eq. (7), as well as the role of γ. Further ablations on the regularization parameter λ, the number of optimization steps, and the mass-assignment scheme in the proxy-update operator are provided in Appendix A.1.

Wasserstein and directional transport. Table 3 reports the changes Δ in HV, GD, IGD, and W_2 relative to the offline front D(best) when using only the matching term $W_2(\cdot, \mu_k)$, only the directional transport term L_n, or both. Results are averaged over 5 seeds per task and then across all tasks within each benchmark family. On DTLZ and RE, their combination yields the largest improvements in all metrics. On ZDT, L_n alone yields the largest reduction in GD, consistent with its role in moving the candidate distribution toward the Pareto front. Without the matching term, however, L_n increases W_2 on ZDT and RE, showing that Wasserstein matching is needed to control mass allocation along the front. Overall, the two terms are complementary: L_n mainly drives convergence (GD), while their combination improves convergence, coverage (IGD) and probability-mass allocation (W_2). Complete per-task ablation results are provided in Appendix A.1.1.

Transport weight γ. We further study the effect of γ in Eq. (7). For each metric, we rank the γ values per task based on their mean performance across 5 seeds and average these ranks across tasks. Per-task results are provided in Appendix A.1.2. Figure 4 shows that increasing γ from 0 to 0.5 improves the average rank in all metrics. IGD and W_2 are best at $\gamma = 0.5$, HV saturates for $\gamma \in \{0.2, 0.5, 1\}$, and GD improves until $\gamma = 2$, consistent with stronger directional transport toward the Pareto front. For larger γ, the ranks worsen, indicating that directional transport must be balanced with Wasserstein matching. We use $\gamma = 0.5$ as the default.

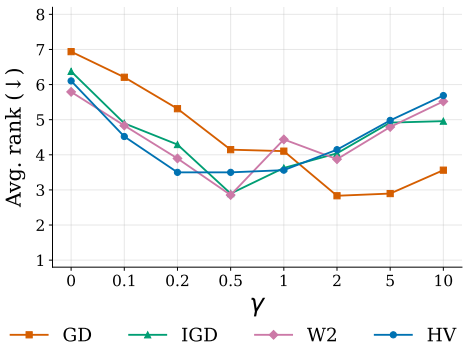


Figure 4: Avg. rank of γ across all tasks.

8 Conclusion

In this work, we approached offline MOO by mass transport toward the Pareto front and introduced a training-free guidance method that explicitly refines the generated objective-space distribution. More broadly, generative models inherently operate on distributions over designs, making it natural to treat the induced distribution itself as the object of optimization rather than generating individual candidate solutions independently. Generative methods are also promising for offline MOO problems whose feasible designs lie on low-dimensional manifolds in high-dimensional ambient spaces (Song & Ermon, 2019), where forward methods operate in the ambient space and may leave the manifold of feasible designs.

Limitations and future work. While the focus on approximating the projection $\bar{\mu}$ of the offline front distribution allows a strong convergence result (Theorem 1), with the remaining gap to μ^* given by $W_2(\bar{\mu}, \mu^*)$, it leaves extrapolation beyond the projection for future research; provable convergence in such settings will require a different focus and different assumptions. Further, we leave the many-objective setting ($m \geq 4$) for future work, where the single global normal that approximates the projection in practice may be insufficient to capture the higher-dimensional Pareto-front geometry.

AI USE STATEMENT

In this work, we used generative AI tools for coding assistance, for discussing mathematical statements, and for polishing writing. We used Claude Code to implement and refactor parts of the experimental code: the ablation harness, evaluation and plotting notebooks, aggregation of results into tables and figures, and code optimization. We did not use generative AI tools to implement the proposed method, and all numbers, figures, and tables were generated from our own experimental runs. We read, tested, and verified all AI-assisted code, and checked every table and figure for consistency with our results. We have used Claude and ChatGPT to discuss mathematical claims and assist in the writing of their proofs. We have not used generative AI tools to develop the conceptual framework of mass transport toward the Pareto front or to formulate Corollary 1 and Theorem 1. We verified all AI-assisted mathematical statements against the standard literature. We also used ChatGPT and Claude to polish writing for readability and clarity. We reviewed and revised each suggested text change. We have not used generative AI tools to generate synthetic datasets, clean or reformat the datasets, develop ideas and conceptual frameworks, propose or refine hypotheses, provide feedback on research methodology and experiments, or interpret results. Translation assistance and support for qualitative or thematic data analysis are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The method is specified in full in Section 5.2: the loss functional in Eq. (7) and the pseudocode in Algorithm 1, with the particle-level derivation and the remaining algorithmic details in Appendix A.3. All hyperparameters needed to reproduce our runs, including the surrogate and flow-matching architectures, and the optimizer settings are listed in Appendix A.4, and the sensitivity of the method to the key choices is reported in Appendix A.1. The benchmark suites and their reference points are described in Appendix A.4.3; we use the publicly available offline MOO benchmark of Xue et al. (2024). The evaluation protocol, including the exact definitions of the metrics and the normalization used to compute them, is given in Appendix A.4.2. For the theoretical results, the assumptions are stated explicitly in Appendix A.5 and complete proofs of Corollary 1 and Theorem 1 are given there. Code will be released upon publication.

REFERENCES

Michael Samuel Albergo and Eric Vanden-Eijnden. Building Normalizing Flows with Stochastic Interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Yashas Annadani, Syrine Belakaria, Stefano Ermon, Stefan Bauer, and Barbara Engelhardt. Preference-Guided Diffusion for Multi-Objective Offline Optimization. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 15738–15761. Curran Associates, Inc., 2025. doi: 10.52202/085713-0533. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/175fa4fc8f275f877ec85340131c5d7a-Paper-Conference.pdf.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-Flow: Differentiating through Flows for Controlled Generation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 3462–3483. PMLR, 21–27 Jul 2024.

David Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 773–782. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/brookes19a.html.

Guillermo Cañas and Lorenzo Rosasco. Learning Probability Measures with respect to Optimal Transport Metrics. In F. Pereira, C.J. Burges, L. Bottou, and K. Weinberger (eds.),

Advances in Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c54e7837e0cd0ced286cb5995327d1ab-Paper.pdf.

Georgios Chasparis, Werner Zellinger, Verena Haunschmid, Markus Riedenbauer, and Reinhard Stumptner. On the optimization of material usage in power transformer manufacturing. In *2016 IEEE 8th International Conference on Intelligent Systems (IS)*, pp. 680–685. IEEE, 2016. doi: 10.1109/IS.2016.7737384.

Can (Sam) Chen, Christopher Beckham, Zixuan Liu, Xue (Steve) Liu, and Chris Pal. Parallel-mentoring for Offline Model-based Optimization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 76619–76636. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/f189e7580acad0fc7fd45405817ddee3-Paper-Conference.pdf.

Yu Chen, Xiufen Zou, and Weicheng Xie. Convergence of multi-objective evolutionary algorithms to a uniformly distributed representation of the Pareto front. *Information Sciences*, 181(16):3336–3355, 2011. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2011.04.004. URL https://www.sciencedirect.com/science/article/pii/S0020025511001721.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 794–803. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/chen18a.html.

Hyungjin Chung, Jeongsol Kim, Michael Thompson McCann, Marc Louis Klasky, and Jong Chul Ye. Diffusion Posterior Sampling for General Noisy Inverse Problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Carlos A. Coello Coello and Margarita Reyes Sierra. A Study of the Parallelization of a Coevolutionary Multi-objective Evolutionary Algorithm. In Raúl Monroy, Gustavo Arroyo-Figueroa, Luis Enrique Sucar, and Humberto Sossa (eds.), *MICAI 2004: Advances in Artificial Intelligence*, pp. 688–697, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-24694-7.

Indraneel Das and J. E. Dennis. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998. doi: 10.1137/S1052623496307510. URL https://doi.org/10.1137/S1052623496307510.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002a. ISSN 1089-778X. doi: 10.1109/4235.996017. URL https://doi.org/10.1109/4235.996017.

Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 1, pp. 825–830, 2002b. doi: 10.1109/CEC.2002.1007032.

Yuanqi Du, Arian R. Jamasb, Jeff Guo, Tianfan Fu, Charles Harris, Yingheng Wang, Chenru Duan, Pietro Liò, Philippe Schwaller, and Tom L. Blundell. Machine learning-aided generative molecular design. *Nature Machine Intelligence*, 6(6):589–604, 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00843-5. URL https://doi.org/10.1038/s42256-024-00843-5.

Ruiqi Feng, Chenglei Yu, Wenhao Deng, Peiyan Hu, and Tailin Wu. On the Guidance of Flow Matching. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 16993–17029. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/feng25s.html.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python Optimal Transport. *Journal of Machine Learning Research*, 22(78): 1–8, 2021. URL http://jmlr.org/papers/v22/20-451.html.

Stephanie Holly, Alexandru-Ciprian Zăvoianu, Siegfried Silber, Sepp Hochreiter, and Werner Zellinger. The Offline-Frontier Shift: Diagnosing Distributional Limits in Generative Multi-Objective Optimization. In *Workshop on Scientific Methods for Understanding Deep Learning*, 2026. URL https://openreview.net/forum?id=4XN7MqWVKD.

Sedjro Salomon Hotegni and Sebastian Peitz. SPREAD: Sampling-based Pareto front Refinement via Efficient Adaptive Diffusion. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4731mIqv89.

Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011. ISSN 2210-6502. doi: https://doi.org/10.1016/j.swevo.2011.05.001. URL https://www.sciencedirect.com/science/article/pii/S2210650211000198.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492, 1998. ISSN 0925-5001. doi: 10.1023/a:1008306431147.

Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. FlowDPS: Flow-Driven Posterior Sampling for Inverse Problems. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12328–12337, 2025a. doi: 10.1109/ICCV51701.2025.01146.

Kwanyoung Kim and Jong Chul Ye. Noise2Score: Tweedie's Approach to Self-Supervised Image Denoising without Clean Images. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 864–874. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/077b83af57538aa183971a2fe0971ec1-Paper.pdf.

Minsu Kim, Jiayao Gu, Ye Yuan, Taeyoung Yun, Zixuan Liu, Yoshua Bengio, and Can Chen. Offline Model-Based Optimization: Comprehensive Review. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=QcSZWo1TLl. Survey Certification.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time Alignment of Diffusion Models without Reward Over-optimization. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=vi3DjUhFVm.

Joshua Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1): 50–66, 2006. doi: 10.1109/TEVC.2005.851274.

Siddarth Krishnamoorthy, Satvik Mehul Mashkaria, and Aditya Grover. Diffusion Models for Black-Box Optimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17842–17857. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/krishnamoorthy23a.html.

Aviral Kumar and Sergey Levine. Model Inversion Networks for Model-Based Optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5126–5137. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/373e4c5d8edfa8b74fd4b6791d0cf6dc-Paper.pdf.

Zeyang Li, Kaveh Alim, and Navid Azizan. HardFlow: Hard-Constrained Sampling for Flow-Matching Models via Trajectory Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(10):13619–13635, 2026. doi: 10.1109/TPAMI.2026.3708824.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022. URL https://openreview.net/forum?id=gWxpdtQpiYV.

Xingchao Liu, Lemeng Wu, Shujian Zhang, Chengyue Gong, Wei Ping, and Qiang Liu. FlowGrad: Controlling the Output of Generative ODEs With Gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24335–24344, June 2023. doi: 10.1109/CVPR52729.2023.02331.

Satvik Mehul Mashkaria, Siddarth Krishnamoorthy, and Aditya Grover. Generative Pretraining for Black-Box Optimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24173–24197. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/mashkaria23a.html.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, 2 edition, 2018. ISBN 9780262039406.

Gabriel Peyré and Marco Cuturi. Computational Optimal Transport with Applications to Data Sciences. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 02 2019. ISSN 1935-8237. doi: 10.1561/2200000073. URL https://doi.org/10.1561/2200000073.

Elizabeth A. Pogue, Alexander New, Kyle McElroy, Nam Q. Le, Michael J. Pekala, Ian McCue, Eddie Gienger, Janna Domenico, Elizabeth Hedrick, Tyrel M. McQueen, Brandon Wilfong, Christine D. Piatko, Christopher R. Ratto, Andrew Lennon, Christine Chung, Timothy Montalbano, Gregory Bassen, and Christopher D. Stiles. Closed-loop superconducting materials discovery. *npj Computational Materials*, 9(181), 2023. ISSN 2057-3960. doi: 10.1038/s41524-023-01131-3. URL https://doi.org/10.1038/s41524-023-01131-3.

Mehrsa Pourya, Bassam El Rawas, and Michael Unser. Flower: A Flow-Matching Solver for Inverse Problems. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=QGd34p02mI.

Han Qi, Yi Su, Aviral Kumar, and Sergey Levine. Data-Driven Offline Decision-Making via Invariant Representation Learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 13226–13237. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/559726fdfb19005e368be4ce3d40e3e5-Paper-Conference.pdf.

Filippo Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*, volume 87 of *Progress in Nonlinear Differential Equations and Their Applications*. Birkhäuser Cham, 2015. ISBN 978-3-319-20828-2. doi: 10.1007/978-3-319-20828-2. URL https://doi.org/10.1007/978-3-319-20828-2.

Oliver Schütze, Xavier Esquivel, Adriana Lara, and Carlos A. Coello Coello. Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522, 2012. doi: 10.1109/TEVC.2011.2161872.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Jatan Shrestha, Santeri Heiskanen, Kari Hepola, Severi Rissanen, Pekka Jääskeläinen, and Joni Pajarinen. Pareto-Conditioned Diffusion Models for Offline Multi-Objective Optimization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=S2Q00li155.

Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf.

Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2020.106078. URL https://www.sciencedirect.com/science/article/pii/S1568494620300181.

Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative Objective Models for Effective Offline Model-Based Optimization. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10358–10368. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/trabucco21a.html.

D.A. Van Veldhuizen and G.B. Lamont. On measuring multiobjective evolutionary algorithm performance. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, volume 1, pp. 204–211, 2000. doi: 10.1109/CEC.2000.870296.

Titouan Vayer and Rémi Gribonval. Controlling Wasserstein Distances by Kernel Norms with Application to Compressive Statistical Learning. *Journal of Machine Learning Research*, 24(149): 1–51, 2023. URL https://jmlr.org/papers/v24/21-1516.html.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, Heidelberg, 2009. ISBN 978-3-540-71049-3. doi: 10.1007/978-3-540-71050-9.

Handing Wang, Yaochu Jin, Chaoli Sun, and John Doherty. Offline Data-Driven Evolutionary Optimization Using Selective Surrogate Ensembles. *IEEE Transactions on Evolutionary Computation*, 23(2):203–216, 2019. doi: 10.1109/TEVC.2018.2834881.

Jike Wang, Chang-Yu Hsieh, Mingyang Wang, Xiaorui Wang, Zhenxing Wu, Dejun Jiang, Benben Liao, Xujun Zhang, Bo Yang, Qiaojun He, Dongsheng Cao, Xi Chen, and Tingjun Hou. Multi-constraint molecular generation based on conditional transformer, knowledge distillation and reinforcement learning. *Nature Machine Intelligence*, 3(10):914–922, 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00403-1. URL https://doi.org/10.1038/s42256-021-00403-1.

Luran Wang, Chaoran Cheng, Yizhen Liao, Yanru Qu, and Ge Liu. Training Free Guided Flow-Matching with Optimal Control. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=61ss5RA1MM.

George Webber, Alexander Denker, Riccardo Barbano, and Andrew J. Reader. Solving Inverse Problems with Flow-based Models via Model Predictive Control. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=57lp1pmJVo.

Ke Xue, Rongxi Tan, Xiaobin Huang, and Chao Qian. Offline Multi-Objective Optimization. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 55595–55624. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/xue24b.html.

Cuie Yang, Jinliang Ding, Yaochu Jin, and Tianyou Chai. Offline Data-Driven Multiobjective Optimization: Knowledge Transfer Between Surrogates and Generation of Final Solutions. *IEEE Transactions on Evolutionary Computation*, 24(3):409–423, 2020. doi: 10.1109/TEVC.2019.2925959.

Sihyun Yu, Sungsoo Ahn, Le Song, and Jinwoo Shin. RoMA: Robust Model Adaptation for Offline Model-based Optimization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4619–4631. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/24b43fb034a10d78bec71274033b4096-Paper.pdf.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient Surgery for Multi-Task Learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf.

Ye Yuan, Can (Sam) Chen, Zixuan Liu, Willie Neiswanger, and Xue (Steve) Liu. Importance-aware Co-Teaching for Offline Model-based Optimization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 55718–55733. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ae8b0b5838ba510daff1198474e7b984-Paper-Conference.pdf.

Ye Yuan, Can Chen, Christopher Pal, and Xue Liu. ParetoFlow: Guided Flows in Multi-Objective Optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mLyyB4le5u.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, Roberto Sordillo, Lixin Sun, Jake Smith, Bichlien Nguyen, Hannes Schulz, Sarah Lewis, Chin-Wei Huang, Ziheng Lu, Yichi Zhou, Han Yang, Hongxia Hao, Jielan Li, Chunlei Yang, Wenjie Li, Ryota Tomioka, and Tian Xie. A generative model for inorganic materials design. *Nature*, 639(8055):624–632, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-08628-5. URL https://doi.org/10.1038/s41586-025-08628-5.

Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999. doi: 10.1109/4235.797969.

Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000. doi: 10.1162/106365600568202.

A Appendix

Appendix Contents

A.1 Ablation study 15
A.2 Training-free guidance of flow-matching models 22
A.3 Algorithmic details 25
A.4 Implementation details 27
A.5 Theoretical results 29
A.6 Additional results 31
A.7 Extended related work 34

A.1 Ablation study

We study the effect of the transport components and the sensitivity of ParetoTransport to key hyperparameters.

A.1.1 TRANSPORT COMPONENTS

We study the contributions of the Wasserstein and directional transport terms in Eq. (7). Each experiment uses 5 seeds, $N = 256$ candidate designs, and $\gamma = 0.5$. Per-task results are provided in Tables 4-6.

Table 4: Ablation of the Wasserstein term W_2 and directional transport term L_n on ZDT. Bold = best per column and metric. Each experiment is run for 5 seeds and evaluated on 256 designs.

W_2	L_n	Metric	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
		HV (↑)	4.750 ± 0.011	5.496 ± 0.024	**5.628** ± 0.177	5.483 ± 0.002	4.816 ± 0.008
	✓	GD (↓)	**0.017** ± 0.003	**0.041** ± 0.048	**0.193** ± 0.028	0.347 ± 0.021	**0.014** ± 0.013
		IGD (↓)	0.116 ± 0.060	0.372 ± 0.142	**0.142** ± 0.081	0.372 ± 0.067	0.020 ± 0.003
		W2 (↓)	0.354 ± 0.026	0.511 ± 0.014	0.355 ± 0.128	0.591 ± 0.012	0.187 ± 0.068
		HV (↑)	4.747 ± 0.014	5.610 ± 0.036	5.509 ± 0.035	**5.485** ± 0.000	**4.822** ± 0.006
✓		GD (↓)	0.182 ± 0.019	0.203 ± 0.025	0.302 ± 0.014	**0.299** ± 0.027	0.020 ± 0.006
		IGD (↓)	0.191 ± 0.015	0.157 ± 0.022	0.234 ± 0.007	**0.266** ± 0.100	**0.017** ± 0.001
		W2 (↓)	0.255 ± 0.020	0.294 ± 0.020	0.360 ± 0.008	**0.512** ± 0.021	0.117 ± 0.023
		HV (↑)	**4.837** ± 0.022	**5.654** ± 0.022	5.589 ± 0.069	5.483 ± 0.004	**4.822** ± 0.007
✓	✓	GD (↓)	0.046 ± 0.045	0.076 ± 0.021	0.235 ± 0.014	0.333 ± 0.034	0.016 ± 0.002
		IGD (↓)	**0.030** ± 0.040	**0.028** ± 0.010	0.154 ± 0.021	0.329 ± 0.054	**0.017** ± 0.002
		W2 (↓)	**0.210** ± 0.112	**0.191** ± 0.063	**0.328** ± 0.029	0.550 ± 0.012	**0.108** ± 0.017

Table 5: Ablation of the Wasserstein term W_2 and directional transport term L_n on DTLZ. Bold = best per column and metric. Each experiment is run for 5 seeds and evaluated on 256 designs.

W_2	L_n	Metric	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
		HV (↑)	10.645 ± 0.001	12.440 ± 0.001	**9.898** ± 0.001	17.324 ± 0.250	10.739 ± 0.006	11.004 ± 0.008	10.738 ± 0.009
	✓	GD (↓)	0.448 ± 0.024	0.027 ± 0.002	**0.394** ± 0.013	**0.050** ± 0.010	**0.026** ± 0.004	0.514 ± 0.008	0.095 ± 0.026
		IGD (↓)	**0.062** ± 0.012	0.038 ± 0.001	**0.107** ± 0.009	0.238 ± 0.045	0.015 ± 0.003	0.295 ± 0.003	0.112 ± 0.012
		W2 (↓)	0.449 ± 0.025	0.121 ± 0.018	**0.394** ± 0.013	0.410 ± 0.008	0.102 ± 0.011	0.520 ± 0.009	0.281 ± 0.038
		HV (↑)	10.645 ± 0.001	12.441 ± 0.000	9.891 ± 0.004	**17.619** ± 0.023	10.730 ± 0.002	10.895 ± 0.037	10.577 ± 0.052
✓		GD (↓)	0.419 ± 0.005	0.044 ± 0.003	0.533 ± 0.008	0.124 ± 0.014	0.082 ± 0.012	0.703 ± 0.006	0.252 ± 0.041
		IGD (↓)	0.081 ± 0.031	0.036 ± 0.001	0.153 ± 0.047	0.159 ± 0.011	0.020 ± 0.001	0.287 ± 0.001	0.149 ± 0.019
		W2 (↓)	0.420 ± 0.005	0.065 ± 0.002	0.533 ± 0.008	0.342 ± 0.017	0.106 ± 0.020	0.712 ± 0.006	0.365 ± 0.028
		HV (↑)	**10.647** ± 0.000	**12.448** ± 0.000	9.896 ± 0.002	17.612 ± 0.010	**10.757** ± 0.000	**11.045** ± 0.009	**10.749** ± 0.062
✓	✓	GD (↓)	**0.362** ± 0.015	**0.023** ± 0.002	0.421 ± 0.006	0.098 ± 0.016	0.048 ± 0.006	**0.346** ± 0.009	**0.058** ± 0.013
		IGD (↓)	0.067 ± 0.017	**0.018** ± 0.001	0.118 ± 0.020	**0.150** ± 0.009	**0.004** ± 0.000	**0.283** ± 0.000	**0.093** ± 0.063
		W2 (↓)	**0.363** ± 0.015	**0.039** ± 0.004	0.421 ± 0.006	**0.323** ± 0.023	**0.066** ± 0.011	**0.352** ± 0.009	**0.205** ± 0.096

Table 6: Ablation of the Wasserstein term W_2 and directional transport term L_n on RE. Bold = best per column and metric. Each experiment is run for 5 seeds and evaluated on 256 designs.

W_2	L_n	Metric	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33	RE34	RE35	RE36	RE37
		HV (↑)	4.295 ± 0.038	4.811 ± 0.011	4.794 ± 0.101	4.816 ± 0.001	**4.840** ± 0.000	10.608 ± 0.002	10.628 ± 0.017	10.618 ± 0.001	9.985 ± 0.007	10.546 ± 0.009	9.254 ± 0.160	6.296 ± 0.075
	✓	GD (↓)	0.018 ± 0.009	0.045 ± 0.034	0.032 ± 0.018	0.239 ± 0.026	0.033 ± 0.070	**0.060** ± 0.044	0.251 ± 0.189	**0.009** ± 0.002	0.012 ± 0.002	**0.005** ± 0.007	0.244 ± 0.042	0.034 ± 0.007
		IGD (↓)	0.051 ± 0.007	0.023 ± 0.016	0.178 ± 0.119	0.064 ± 0.003	0.001 ± 0.000	**0.324** ± 0.006	0.145 ± 0.093	0.068 ± 0.009	0.036 ± 0.008	0.208 ± 0.075	0.127 ± 0.019	0.072 ± 0.007
		W2 (↓)	0.222 ± 0.027	**0.106** ± 0.035	0.543 ± 0.080	0.253 ± 0.027	0.199 ± 0.049	**0.415** ± 0.020	0.431 ± 0.243	0.245 ± 0.067	0.159 ± 0.036	0.431 ± 0.031	0.425 ± 0.011	0.169 ± 0.008
		HV (↑)	4.520 ± 0.006	**4.840** ± 0.000	4.838 ± 0.003	4.831 ± 0.004	4.838 ± 0.001	**10.624** ± 0.008	**10.637** ± 0.002	10.612 ± 0.001	10.024 ± 0.021	10.394 ± 0.010	9.833 ± 0.195	6.530 ± 0.039
✓		GD (↓)	0.070 ± 0.004	0.030 ± 0.041	0.035 ± 0.046	0.079 ± 0.031	0.033 ± 0.020	0.136 ± 0.085	**0.088** ± 0.023	0.049 ± 0.017	0.030 ± 0.006	0.059 ± 0.008	0.208 ± 0.016	0.075 ± 0.005
		IGD (↓)	0.056 ± 0.004	**0.012** ± 0.004	**0.038** ± 0.022	0.014 ± 0.007	0.001 ± 0.001	0.332 ± 0.013	0.224 ± 0.058	0.071 ± 0.023	0.041 ± 0.001	0.081 ± 0.005	0.133 ± 0.029	0.094 ± 0.004
		W2 (↓)	0.107 ± 0.006	0.142 ± 0.037	0.285 ± 0.050	0.099 ± 0.031	0.040 ± 0.023	0.472 ± 0.082	0.328 ± 0.011	0.249 ± 0.031	0.172 ± 0.038	**0.212** ± 0.031	0.273 ± 0.020	0.174 ± 0.009
		HV (↑)	**4.577** ± 0.007	**4.840** ± 0.000	**4.839** ± 0.000	**4.833** ± 0.002	4.838 ± 0.001	10.614 ± 0.010	10.632 ± 0.007	**10.626** ± 0.001	**10.091** ± 0.018	**10.584** ± 0.009	**10.186** ± 0.139	**6.681** ± 0.018
✓	✓	GD (↓)	**0.001** ± 0.000	**0.018** ± 0.017	**0.007** ± 0.006	**0.076** ± 0.018	**0.030** ± 0.013	0.137 ± 0.062	0.123 ± 0.071	0.023 ± 0.008	**0.008** ± 0.001	0.007 ± 0.010	**0.138** ± 0.036	**0.018** ± 0.002
		IGD (↓)	**0.006** ± 0.001	0.018 ± 0.007	0.083 ± 0.082	**0.006** ± 0.003	**0.000** ± 0.000	0.325 ± 0.009	**0.133** ± 0.091	**0.060** ± 0.014	**0.022** ± 0.002	**0.052** ± 0.048	**0.042** ± 0.030	**0.054** ± 0.003
		W2 (↓)	**0.065** ± 0.008	0.139 ± 0.016	**0.249** ± 0.095	**0.096** ± 0.018	**0.036** ± 0.016	0.466 ± 0.072	**0.298** ± 0.018	**0.224** ± 0.050	**0.152** ± 0.010	0.214 ± 0.035	**0.211** ± 0.045	**0.138** ± 0.011

A.1.2 HYPERPARAMETER SENSITIVITY

We study the sensitivity of ParetoTransport to key hyperparameters and implementation choices, including the regularization parameter λ, the number of inner optimization steps, the mass-assignment scheme in the proxy-update operator, and the transport weight γ. We report results on ZDT1, DTLZ2, and RE37. Each experiment is run for 5 seeds and evaluated on 256 designs.

Regularization parameter. We vary $\lambda \in \{0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. Figures 5–8 report HV, GD, IGD, and W_2 as a function of λ, with the offline front D(best) shown for reference. On both synthetic and real-world tasks (Figures 5, 6 and 7), performance is insensitive to λ up to 10^{-3}–10^{-2}, and then deteriorates, with the metrics moving toward their D(best) values. Thus, stronger regularization restricts exploration and keeps the generated samples close to the offline dataset. For the highly multimodal ZDT4, we additionally evaluate $\lambda \in \{0, 0.01, 0.05, 0.1, 0.5, 1\}$. Figure 8

shows the same mechanism with the opposite effect: here D(best) is better than the generated solution set in GD, IGD, and W_2, so staying closer to the data tends to improve these metrics, while HV is unchanged. Figure 9 reports the average rank across all tasks. We use $\lambda = 10^{-5}$ as the default.

Since $\lambda = 0$ performs on par, drift from the prior is in practice limited mainly by the delayed guidance onset $t_{\text{guided}} = 0.8$; larger λ acts as a safeguard for unreliable surrogates, as on ZDT4.

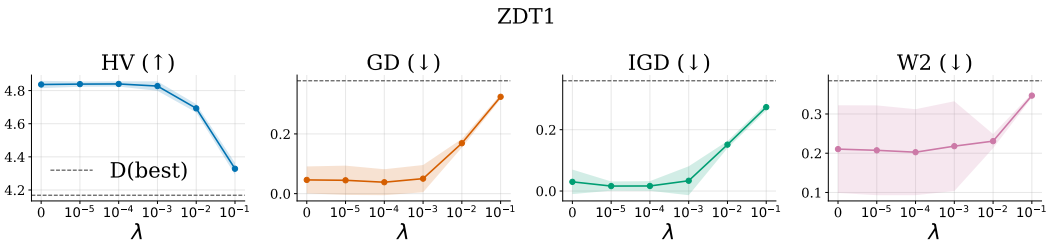


Figure 5: Effect of regularization weight λ on ZDT1.

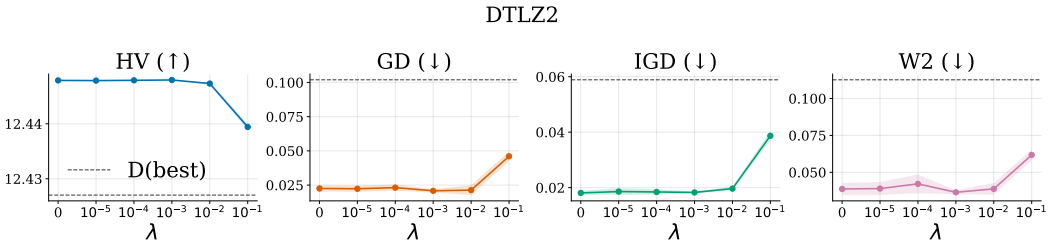


Figure 6: Effect of regularization weight λ on DTLZ2.

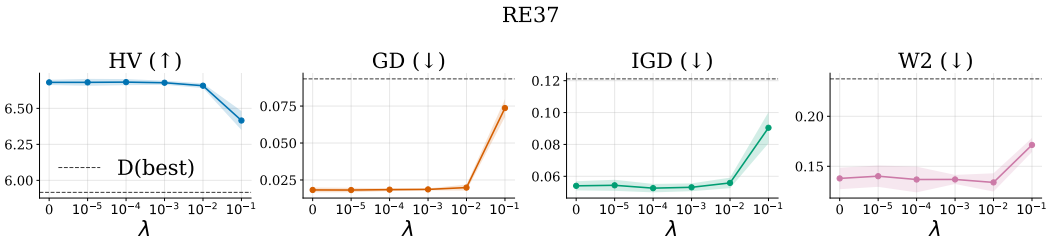


Figure 7: Effect of regularization weight λ on RE37.

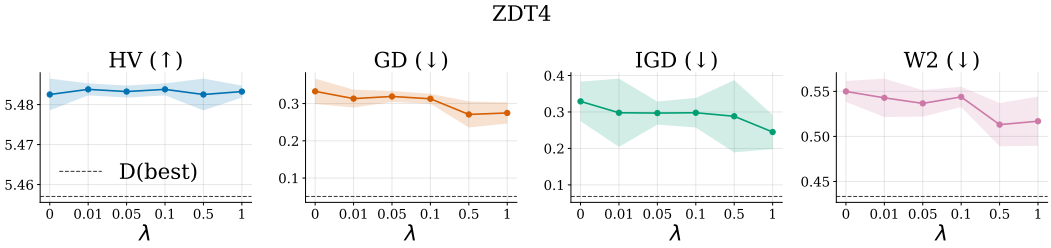


Figure 8: Effect of regularization weight λ on ZDT4 (highly multimodal task).

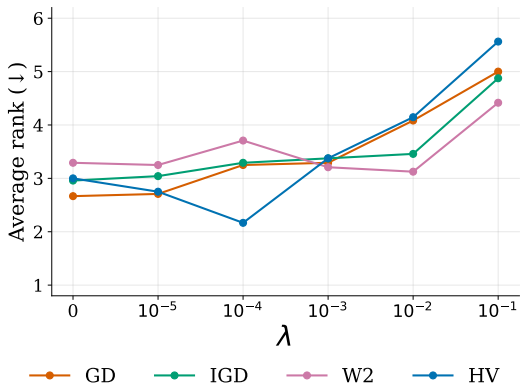


Figure 9: Average rank of λ across all tasks.

Optimization steps. We vary the number of optimization steps used to minimize the loss functional in Eq. (7) over $\{1, 5, 10, 20, 30\}$. Figures 10–12 report HV, GD, IGD, and W_2 as the number of optimization steps increases, with the offline front D(best) shown for reference. Across synthetic and real-world tasks, performance generally improves up to 5 steps and then saturates. Figure 13 reports the average rank across all tasks. We use 10 as the default.

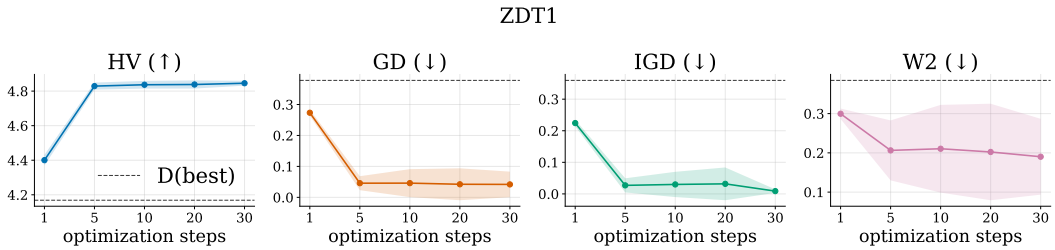


Figure 10: Effect of the number of optimization steps on ZDT1.

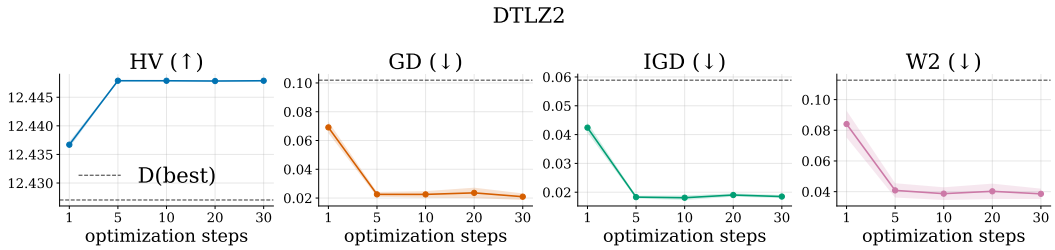


Figure 11: Effect of the number of optimization steps on DTLZ2.

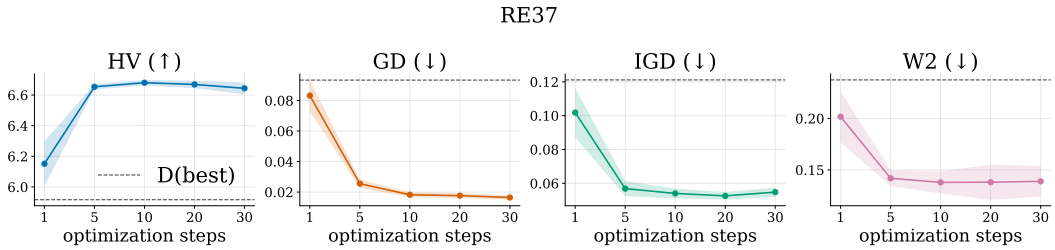


Figure 12: Effect of the number of optimization steps on RE37.

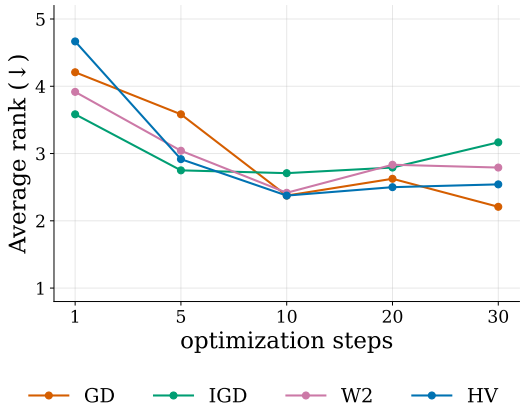


Figure 13: Average rank of optimization-step settings across all tasks.

Probability-mass assignment. The proxy-update operator $\mathcal{R}$ computes the support $(y^j)_{j=1}^M$ of the next proxy distribution and assigns probability masses $(b^j)_{j=1}^M$. We compare equal weights $b^j = \frac{1}{M}$ with crowding-based weights proportional to crowding distance (Appendix A.3.2). Table 7 reports the changes Δ relative to D(best). The difference is minor on the synthetic tasks. On the real-world engineering tasks, crowding gives a clearer W_2 improvement but trades some GD for IGD (Figure 14). This reflects the intended trade-off: assigning more mass to points in sparsely represented regions improves coverage along the Pareto front, but can shift mass toward less-converged regions. We use crowding-based weights as the default.

Table 7: Ablation of proxy weights. Results are averaged over all tasks within each benchmark family and reported as changes Δ relative to the offline front D(best); bold = best per column.

	ZDT				DTLZ				RE			
	HV (↑)	GD (↓)	IGD (↓)	W_2 (↓)	HV (↑)	GD (↓)	IGD (↓)	W_2 (↓)	HV (↑)	GD (↓)	IGD (↓)	W_2 (↓)
D(best)	4.812	0.262	0.261	0.374	11.542	0.373	0.247	0.427	7.368	0.075	0.123	0.226
equal	+0.458	-0.113	**-0.159**	-0.075	+0.336	-0.174	-0.137	-0.158	+0.405	**-0.031**	-0.045	-0.009
crowding	**+0.465**	**-0.121**	-0.150	**-0.096**	**+0.337**	**-0.179**	**-0.142**	**-0.174**	**+0.410**	-0.026	**-0.056**	**-0.035**

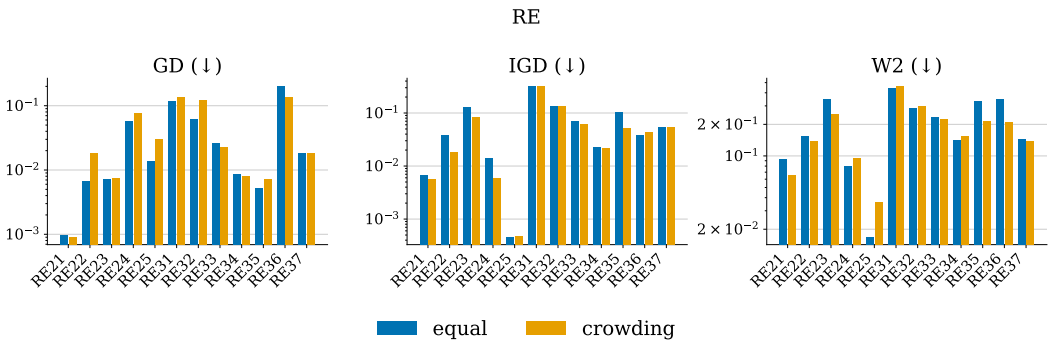


Figure 14: Ablation of equal versus crowding-based proxy weights on RE.

Transport weight. We vary $\gamma \in \{0, 0.1, 0.2, 0.5, 1, 2, 5, 10\}$ and provide per-task results for HV, GD, IGD, and W_2. A rank summary is provided in Figure 4.

Table 8: Ablation of the transport weight γ on ZDT tasks. HV is computed over a population of 256 candidate designs and averaged over 5 random seeds.

γ	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
0	4.747 ± 0.014	5.610 ± 0.036	5.509 ± 0.035	$\mathbf{5.485 \pm 0.000}$	4.822 ± 0.006
0.1	4.791 ± 0.018	5.646 ± 0.023	5.533 ± 0.047	5.484 ± 0.001	$\mathbf{4.825 \pm 0.001}$
0.2	4.835 ± 0.013	$\mathbf{5.667 \pm 0.030}$	5.524 ± 0.016	5.484 ± 0.002	$\mathbf{4.825 \pm 0.001}$
0.5	4.837 ± 0.022	5.654 ± 0.022	5.589 ± 0.069	5.483 ± 0.004	4.822 ± 0.007
1	$\mathbf{4.841 \pm 0.018}$	5.664 ± 0.032	5.542 ± 0.050	5.484 ± 0.001	4.824 ± 0.001
2	4.834 ± 0.023	5.653 ± 0.029	5.553 ± 0.123	5.483 ± 0.002	$\mathbf{4.825 \pm 0.001}$
5	4.828 ± 0.029	5.659 ± 0.043	5.589 ± 0.173	5.483 ± 0.002	4.824 ± 0.000
10	4.815 ± 0.026	5.640 ± 0.045	$\mathbf{5.593 \pm 0.163}$	5.483 ± 0.001	4.819 ± 0.007

Table 9: Ablation of the transport weight γ on ZDT tasks. GD is computed over a population of 256 candidate designs and averaged over 5 random seeds.

γ	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
0	0.182 ± 0.019	0.203 ± 0.025	0.302 ± 0.014	0.299 ± 0.027	0.020 ± 0.006
0.1	0.113 ± 0.017	0.133 ± 0.021	0.299 ± 0.009	$\mathbf{0.295 \pm 0.017}$	0.020 ± 0.011
0.2	0.056 ± 0.024	0.103 ± 0.033	0.285 ± 0.024	0.307 ± 0.026	0.020 ± 0.011
0.5	0.046 ± 0.045	0.076 ± 0.021	0.235 ± 0.014	0.333 ± 0.034	0.016 ± 0.002
1	0.031 ± 0.031	0.058 ± 0.020	0.203 ± 0.014	0.319 ± 0.024	0.019 ± 0.011
2	0.026 ± 0.034	0.033 ± 0.020	0.191 ± 0.026	0.344 ± 0.016	0.017 ± 0.011
5	0.019 ± 0.023	0.027 ± 0.014	$\mathbf{0.186 \pm 0.030}$	0.335 ± 0.022	$\mathbf{0.014 \pm 0.014}$
10	$\mathbf{0.018 \pm 0.013}$	$\mathbf{0.020 \pm 0.013}$	0.188 ± 0.028	0.348 ± 0.010	0.016 ± 0.012

Table 10: Ablation of the transport weight γ on ZDT tasks. IGD is computed over a population of 256 candidate designs and averaged over 5 random seeds.

γ	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
0	0.191 ± 0.015	0.157 ± 0.022	0.234 ± 0.007	$\mathbf{0.266 \pm 0.100}$	0.017 ± 0.001
0.1	0.095 ± 0.013	0.035 ± 0.007	0.208 ± 0.014	0.275 ± 0.059	$\mathbf{0.016 \pm 0.001}$
0.2	0.017 ± 0.003	0.029 ± 0.007	0.171 ± 0.012	0.320 ± 0.085	$\mathbf{0.016 \pm 0.001}$
0.5	0.030 ± 0.040	0.028 ± 0.010	0.154 ± 0.021	0.329 ± 0.054	0.017 ± 0.002
1	$\mathbf{0.016 \pm 0.013}$	0.029 ± 0.013	0.154 ± 0.020	0.379 ± 0.077	0.017 ± 0.001
2	0.036 ± 0.054	$\mathbf{0.025 \pm 0.013}$	0.159 ± 0.056	0.353 ± 0.050	0.017 ± 0.001
5	0.040 ± 0.060	0.027 ± 0.014	$\mathbf{0.148 \pm 0.080}$	0.358 ± 0.032	0.018 ± 0.001
10	0.037 ± 0.053	0.026 ± 0.010	$\mathbf{0.148 \pm 0.078}$	0.397 ± 0.079	0.019 ± 0.003

Table 11: Ablation of the transport weight γ on ZDT tasks. W_2 is computed over a population of 256 candidate designs and averaged over 5 random seeds.

γ	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
0	0.255 ± 0.020	0.294 ± 0.020	0.360 ± 0.008	$\mathbf{0.512 \pm 0.021}$	0.117 ± 0.023
0.1	0.177 ± 0.012	0.252 ± 0.039	0.353 ± 0.006	0.531 ± 0.022	0.132 ± 0.050
0.2	$\mathbf{0.166 \pm 0.092}$	0.210 ± 0.064	0.329 ± 0.023	0.544 ± 0.022	0.139 ± 0.049
0.5	0.210 ± 0.112	$\mathbf{0.191 \pm 0.063}$	$\mathbf{0.328 \pm 0.029}$	0.550 ± 0.012	$\mathbf{0.108 \pm 0.017}$
1	0.213 ± 0.109	0.233 ± 0.106	0.388 ± 0.055	0.569 ± 0.008	0.190 ± 0.084
2	0.209 ± 0.112	0.235 ± 0.074	0.385 ± 0.094	0.574 ± 0.011	0.166 ± 0.053
5	0.210 ± 0.127	0.232 ± 0.087	0.369 ± 0.106	0.581 ± 0.006	0.195 ± 0.082
10	0.217 ± 0.130	0.244 ± 0.107	0.372 ± 0.107	0.587 ± 0.012	0.161 ± 0.120

Table 12: Ablation of the transport weight γ on DTLZ tasks. HV is computed over a population of 256 candidate designs and averaged over 5 random seeds.

γ	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
0	10.645 ± 0.001	12.441 ± 0.000	9.891 ± 0.004	$\mathbf{17.619 \pm 0.023}$	10.730 ± 0.002	10.895 ± 0.037	10.577 ± 0.052
0.1	$\mathbf{10.647 \pm 0.000}$	$\mathbf{12.448 \pm 0.000}$	9.893 ± 0.001	17.617 ± 0.023	10.754 ± 0.000	10.944 ± 0.033	10.741 ± 0.063
0.2	$\mathbf{10.647 \pm 0.001}$	$\mathbf{12.448 \pm 0.000}$	9.893 ± 0.002	$\mathbf{17.619 \pm 0.023}$	10.755 ± 0.001	10.992 ± 0.024	$\mathbf{10.755 \pm 0.058}$
0.5	$\mathbf{10.647 \pm 0.000}$	$\mathbf{12.448 \pm 0.000}$	9.896 ± 0.002	17.612 ± 0.010	$\mathbf{10.757 \pm 0.000}$	11.045 ± 0.009	10.749 ± 0.062
1	$\mathbf{10.647 \pm 0.000}$	$\mathbf{12.448 \pm 0.000}$	$\mathbf{9.898 \pm 0.001}$	17.613 ± 0.006	10.756 ± 0.000	11.054 ± 0.009	10.753 ± 0.057
2	$\mathbf{10.647 \pm 0.000}$	12.447 ± 0.000	$\mathbf{9.898 \pm 0.000}$	17.614 ± 0.006	10.756 ± 0.000	$\mathbf{11.062 \pm 0.003}$	10.734 ± 0.071
5	$\mathbf{10.647 \pm 0.000}$	12.444 ± 0.001	$\mathbf{9.898 \pm 0.001}$	17.609 ± 0.011	10.749 ± 0.001	11.012 ± 0.006	10.722 ± 0.072
10	10.646 ± 0.001	12.441 ± 0.001	$\mathbf{9.898 \pm 0.001}$	17.603 ± 0.011	10.741 ± 0.005	11.012 ± 0.006	10.753 ± 0.007

Table 13: Ablation of the transport weight γ on DTLZ tasks. GD is computed over a population of 256 candidate designs and averaged over 5 random seeds.

γ	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
0	0.419 ± 0.005	0.044 ± 0.003	0.533 ± 0.008	0.124 ± 0.014	0.082 ± 0.012	0.703 ± 0.006	0.252 ± 0.041
0.1	0.393 ± 0.012	0.023 ± 0.003	0.493 ± 0.020	0.130 ± 0.010	0.064 ± 0.008	0.604 ± 0.006	0.129 ± 0.014
0.2	0.379 ± 0.007	0.023 ± 0.002	0.467 ± 0.026	0.124 ± 0.012	0.057 ± 0.007	0.509 ± 0.014	0.094 ± 0.022
0.5	0.362 ± 0.015	0.023 ± 0.002	0.421 ± 0.006	0.098 ± 0.016	0.048 ± 0.006	0.346 ± 0.009	0.058 ± 0.013
1	0.359 ± 0.012	$\mathbf{0.022 \pm 0.005}$	0.402 ± 0.011	0.083 ± 0.007	0.045 ± 0.009	$\mathbf{0.313 \pm 0.010}$	0.056 ± 0.011
2	$\mathbf{0.357 \pm 0.019}$	$\mathbf{0.022 \pm 0.005}$	$\mathbf{0.383 \pm 0.010}$	0.064 ± 0.007	0.039 ± 0.008	0.335 ± 0.014	0.049 ± 0.010
5	$\mathbf{0.357 \pm 0.024}$	0.023 ± 0.002	0.386 ± 0.012	0.055 ± 0.009	0.037 ± 0.007	0.474 ± 0.015	$\mathbf{0.018 \pm 0.002}$
10	0.395 ± 0.031	0.026 ± 0.003	0.385 ± 0.016	$\mathbf{0.053 \pm 0.008}$	$\mathbf{0.036 \pm 0.005}$	0.485 ± 0.013	0.039 ± 0.018

Table 14: Ablation of the transport weight γ on DTLZ tasks. IGD is computed over a population of 256 candidate designs and averaged over 5 random seeds.

γ	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
0	0.081 ± 0.031	0.036 ± 0.001	0.153 ± 0.047	0.159 ± 0.011	0.020 ± 0.001	0.287 ± 0.001	0.149 ± 0.019
0.1	$\mathbf{0.056 \pm 0.013}$	0.022 ± 0.001	0.148 ± 0.036	0.162 ± 0.012	0.006 ± 0.000	0.290 ± 0.005	0.100 ± 0.054
0.2	0.070 ± 0.020	0.020 ± 0.001	0.141 ± 0.030	0.162 ± 0.015	0.006 ± 0.000	0.288 ± 0.002	0.097 ± 0.058
0.5	0.067 ± 0.017	$\mathbf{0.018 \pm 0.001}$	0.118 ± 0.020	$\mathbf{0.150 \pm 0.009}$	$\mathbf{0.004 \pm 0.000}$	$\mathbf{0.283 \pm 0.000}$	$\mathbf{0.093 \pm 0.063}$
1	0.072 ± 0.026	0.019 ± 0.001	0.109 ± 0.009	0.158 ± 0.013	$\mathbf{0.004 \pm 0.000}$	$\mathbf{0.283 \pm 0.000}$	0.095 ± 0.061
2	0.068 ± 0.020	0.024 ± 0.001	0.109 ± 0.009	0.151 ± 0.011	0.005 ± 0.000	0.284 ± 0.001	0.122 ± 0.082
5	0.068 ± 0.017	0.032 ± 0.002	0.117 ± 0.012	0.152 ± 0.013	0.010 ± 0.001	0.291 ± 0.002	0.144 ± 0.091
10	0.065 ± 0.012	0.036 ± 0.000	$\mathbf{0.107 \pm 0.011}$	0.156 ± 0.008	0.014 ± 0.002	0.291 ± 0.002	0.102 ± 0.017

Table 15: Ablation of the transport weight γ on DTLZ tasks. W_2 is computed over a population of 256 candidate designs and averaged over 5 random seeds.

γ	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
0	0.420 ± 0.005	0.065 ± 0.002	0.533 ± 0.008	0.342 ± 0.017	0.106 ± 0.020	0.712 ± 0.006	0.365 ± 0.028
0.1	0.393 ± 0.012	0.040 ± 0.004	0.494 ± 0.020	0.358 ± 0.027	0.090 ± 0.017	0.611 ± 0.006	0.257 ± 0.060
0.2	0.379 ± 0.007	$\mathbf{0.038 \pm 0.002}$	0.467 ± 0.026	0.350 ± 0.022	0.074 ± 0.009	0.517 ± 0.015	0.239 ± 0.067
0.5	0.363 ± 0.015	0.039 ± 0.004	0.421 ± 0.006	0.323 ± 0.023	$\mathbf{0.066 \pm 0.011}$	0.352 ± 0.009	$\mathbf{0.205 \pm 0.096}$
1	0.359 ± 0.012	0.043 ± 0.005	0.403 ± 0.011	0.330 ± 0.023	0.071 ± 0.015	$\mathbf{0.319 \pm 0.010}$	0.227 ± 0.095
2	0.358 ± 0.019	0.062 ± 0.008	$\mathbf{0.383 \pm 0.010}$	$\mathbf{0.311 \pm 0.019}$	0.078 ± 0.011	0.343 ± 0.014	0.255 ± 0.108
5	$\mathbf{0.357 \pm 0.024}$	0.109 ± 0.019	0.386 ± 0.012	0.322 ± 0.006	0.114 ± 0.015	0.485 ± 0.013	0.286 ± 0.075
10	0.396 ± 0.031	0.114 ± 0.012	0.385 ± 0.016	0.336 ± 0.022	0.112 ± 0.012	0.494 ± 0.012	0.233 ± 0.037

Table 16: Ablation of the transport weight γ on RE tasks. HV is computed over a population of 256 candidate designs and averaged over 5 random seeds.

γ	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33	RE34	RE35	RE36	RE37
0	4.520 ± 0.006	$\mathbf{4.840 \pm 0.000}$	4.838 ± 0.003	4.831 ± 0.004	4.838 ± 0.001	$\mathbf{10.624 \pm 0.008}$	$\mathbf{10.637 \pm 0.002}$	10.612 ± 0.001	10.024 ± 0.021	10.394 ± 0.010	9.833 ± 0.195	6.530 ± 0.039
0.1	$\mathbf{4.588 \pm 0.005}$	$\mathbf{4.840 \pm 0.000}$	4.839 ± 0.000	4.830 ± 0.005	4.838 ± 0.001	$\mathbf{10.624 \pm 0.013}$	10.633 ± 0.006	10.620 ± 0.004	10.088 ± 0.007	10.524 ± 0.020	10.112 ± 0.156	6.662 ± 0.007
0.2	$\mathbf{4.588 \pm 0.003}$	$\mathbf{4.840 \pm 0.000}$	4.839 ± 0.000	4.831 ± 0.002	4.838 ± 0.000	10.619 ± 0.015	10.627 ± 0.011	10.623 ± 0.001	$\mathbf{10.096 \pm 0.008}$	10.578 ± 0.008	10.216 ± 0.115	$\mathbf{6.706 \pm 0.004}$
0.5	4.577 ± 0.007	$\mathbf{4.840 \pm 0.000}$	4.839 ± 0.000	$\mathbf{4.833 \pm 0.002}$	4.838 ± 0.001	10.614 ± 0.010	10.632 ± 0.007	$\mathbf{10.626 \pm 0.001}$	10.091 ± 0.018	$\mathbf{10.584 \pm 0.009}$	10.186 ± 0.139	6.681 ± 0.018
1	4.568 ± 0.008	$\mathbf{4.840 \pm 0.000}$	4.839 ± 0.000	4.832 ± 0.002	4.838 ± 0.001	10.611 ± 0.008	10.624 ± 0.011	$\mathbf{10.626 \pm 0.001}$	10.049 ± 0.021	10.579 ± 0.009	$\mathbf{10.263 \pm 0.080}$	6.643 ± 0.027
2	4.543 ± 0.005	4.839 ± 0.002	4.839 ± 0.000	4.832 ± 0.002	$\mathbf{4.839 \pm 0.000}$	10.614 ± 0.018	10.626 ± 0.011	$\mathbf{10.626 \pm 0.002}$	10.011 ± 0.010	10.577 ± 0.013	10.216 ± 0.185	6.576 ± 0.028
5	4.506 ± 0.010	4.838 ± 0.003	$\mathbf{4.840 \pm 0.000}$	$\mathbf{4.833 \pm 0.002}$	$\mathbf{4.839 \pm 0.000}$	10.611 ± 0.008	10.628 ± 0.015	10.623 ± 0.003	9.990 ± 0.013	10.574 ± 0.017	10.132 ± 0.172	6.498 ± 0.040
10	4.466 ± 0.011	4.839 ± 0.002	$\mathbf{4.840 \pm 0.000}$	4.828 ± 0.007	$\mathbf{4.839 \pm 0.001}$	10.608 ± 0.003	10.629 ± 0.014	$\mathbf{10.626 \pm 0.002}$	9.987 ± 0.011	10.574 ± 0.006	9.967 ± 0.138	6.487 ± 0.027

Table 17: Ablation of the transport weight γ on RE tasks. GD is computed over a population of 256 candidate designs and averaged over 5 random seeds.

γ	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33	RE34	RE35	RE36	RE37
0	0.070 ± 0.004	0.030 ± 0.041	0.035 ± 0.046	0.079 ± 0.031	0.033 ± 0.020	0.136 ± 0.085	$\mathbf{0.088 \pm 0.023}$	0.049 ± 0.017	0.030 ± 0.006	0.059 ± 0.008	0.208 ± 0.016	0.075 ± 0.005
0.1	0.002 ± 0.001	$\mathbf{0.018 \pm 0.020}$	0.041 ± 0.054	0.087 ± 0.033	0.038 ± 0.020	0.137 ± 0.074	0.094 ± 0.033	0.037 ± 0.015	0.014 ± 0.004	0.038 ± 0.008	0.152 ± 0.026	0.035 ± 0.003
0.2	$\mathbf{0.001 \pm 0.000}$	0.020 ± 0.025	0.018 ± 0.025	0.086 ± 0.030	0.030 ± 0.017	0.145 ± 0.054	0.112 ± 0.075	0.028 ± 0.010	0.010 ± 0.005	0.016 ± 0.015	0.139 ± 0.032	0.022 ± 0.001
0.5	$\mathbf{0.001 \pm 0.000}$	$\mathbf{0.018 \pm 0.017}$	0.007 ± 0.006	0.076 ± 0.018	0.030 ± 0.013	0.137 ± 0.062	0.123 ± 0.071	0.023 ± 0.008	$\mathbf{0.008 \pm 0.001}$	0.007 ± 0.010	0.138 ± 0.036	0.018 ± 0.002
1	$\mathbf{0.001 \pm 0.000}$	0.027 ± 0.033	0.024 ± 0.045	0.077 ± 0.019	0.038 ± 0.014	0.137 ± 0.054	0.166 ± 0.100	0.016 ± 0.005	0.010 ± 0.003	0.005 ± 0.006	0.140 ± 0.043	$\mathbf{0.017 \pm 0.002}$
2	$\mathbf{0.001 \pm 0.000}$	0.019 ± 0.021	0.003 ± 0.002	0.074 ± 0.018	0.027 ± 0.013	$\mathbf{0.099 \pm 0.044}$	0.197 ± 0.118	0.014 ± 0.003	0.012 ± 0.003	$\mathbf{0.004 \pm 0.006}$	$\mathbf{0.131 \pm 0.044}$	$\mathbf{0.017 \pm 0.001}$
5	$\mathbf{0.001 \pm 0.000}$	0.021 ± 0.031	$\mathbf{0.002 \pm 0.001}$	$\mathbf{0.067 \pm 0.023}$	0.021 ± 0.011	0.114 ± 0.075	0.233 ± 0.076	$\mathbf{0.012 \pm 0.003}$	0.014 ± 0.003	0.005 ± 0.007	0.137 ± 0.038	$\mathbf{0.017 \pm 0.002}$
10	$\mathbf{0.001 \pm 0.000}$	0.019 ± 0.034	0.006 ± 0.006	0.122 ± 0.097	$\mathbf{0.019 \pm 0.012}$	0.103 ± 0.072	0.229 ± 0.088	0.013 ± 0.006	0.012 ± 0.004	0.005 ± 0.007	0.147 ± 0.047	$\mathbf{0.017 \pm 0.002}$

Table 18: Ablation of the transport weight γ on RE tasks. IGD is computed over a population of 256 candidate designs and averaged over 5 random seeds.

γ	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33	RE34	RE35	RE36	RE37
0	0.056 ± 0.004	$\mathbf{0.012 \pm 0.004}$	$\mathbf{0.038 \pm 0.022}$	0.014 ± 0.007	0.001 ± 0.001	0.332 ± 0.013	0.224 ± 0.058	0.071 ± 0.023	0.041 ± 0.001	0.081 ± 0.005	0.133 ± 0.029	0.094 ± 0.004
0.1	$\mathbf{0.006 \pm 0.000}$	0.017 ± 0.011	0.131 ± 0.078	0.013 ± 0.011	$\mathbf{0.000 \pm 0.000}$	0.334 ± 0.015	0.134 ± 0.080	0.066 ± 0.015	0.027 ± 0.002	0.067 ± 0.027	0.061 ± 0.025	0.061 ± 0.004
0.2	0.007 ± 0.001	0.026 ± 0.031	0.148 ± 0.078	0.010 ± 0.004	0.001 ± 0.000	0.329 ± 0.007	0.123 ± 0.076	0.065 ± 0.015	0.024 ± 0.002	$\mathbf{0.043 \pm 0.008}$	0.040 ± 0.020	0.053 ± 0.002
0.5	$\mathbf{0.006 \pm 0.001}$	0.018 ± 0.007	0.083 ± 0.082	$\mathbf{0.006 \pm 0.003}$	$\mathbf{0.000 \pm 0.000}$	0.325 ± 0.009	0.133 ± 0.091	$\mathbf{0.060 \pm 0.014}$	$\mathbf{0.022 \pm 0.002}$	0.052 ± 0.048	0.042 ± 0.030	0.054 ± 0.003
1	$\mathbf{0.006 \pm 0.001}$	0.023 ± 0.017	0.138 ± 0.069	0.008 ± 0.003	0.001 ± 0.000	$\mathbf{0.322 \pm 0.007}$	0.138 ± 0.099	0.061 ± 0.017	0.024 ± 0.002	0.099 ± 0.060	$\mathbf{0.031 \pm 0.017}$	$\mathbf{0.052 \pm 0.003}$
2	0.007 ± 0.001	0.031 ± 0.027	0.092 ± 0.079	0.010 ± 0.003	$\mathbf{0.000 \pm 0.000}$	0.324 ± 0.007	0.127 ± 0.094	0.061 ± 0.015	0.028 ± 0.002	0.113 ± 0.068	0.040 ± 0.031	0.055 ± 0.002
5	0.012 ± 0.001	0.023 ± 0.024	0.076 ± 0.069	0.015 ± 0.012	$\mathbf{0.000 \pm 0.000}$	$\mathbf{0.322 \pm 0.006}$	$\mathbf{0.104 \pm 0.050}$	0.066 ± 0.010	0.032 ± 0.002	0.124 ± 0.071	0.041 ± 0.022	0.056 ± 0.003
10	0.019 ± 0.002	0.017 ± 0.007	0.041 ± 0.021	0.021 ± 0.024	$\mathbf{0.000 \pm 0.000}$	0.324 ± 0.006	0.132 ± 0.098	0.069 ± 0.009	0.035 ± 0.005	0.117 ± 0.041	0.058 ± 0.019	0.054 ± 0.005

Table 19: Ablation of the transport weight γ on RE tasks. W_2 is computed over a population of 256 candidate designs and averaged over 5 random seeds.

γ	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33	RE34	RE35	RE36	RE37
0	0.107 ± 0.006	0.142 ± 0.037	0.285 ± 0.050	0.099 ± 0.031	0.040 ± 0.023	0.472 ± 0.082	0.328 ± 0.011	0.249 ± 0.031	0.172 ± 0.038	0.212 ± 0.031	0.273 ± 0.020	0.174 ± 0.009
0.1	0.079 ± 0.010	0.141 ± 0.021	0.329 ± 0.041	0.108 ± 0.033	0.043 ± 0.020	0.476 ± 0.069	0.310 ± 0.018	0.219 ± 0.042	$\mathbf{0.122 \pm 0.022}$	0.215 ± 0.051	0.216 ± 0.045	0.142 ± 0.006
0.2	0.068 ± 0.005	0.150 ± 0.025	0.350 ± 0.032	0.106 ± 0.030	0.036 ± 0.020	0.478 ± 0.047	0.314 ± 0.011	$\mathbf{0.208 \pm 0.030}$	0.132 ± 0.007	$\mathbf{0.192 \pm 0.017}$	0.214 ± 0.038	$\mathbf{0.130 \pm 0.007}$
0.5	$\mathbf{0.065 \pm 0.008}$	$\mathbf{0.139 \pm 0.016}$	0.249 ± 0.095	0.096 ± 0.018	0.036 ± 0.016	0.466 ± 0.072	$\mathbf{0.298 \pm 0.018}$	0.224 ± 0.050	0.152 ± 0.010	0.214 ± 0.035	0.211 ± 0.045	0.138 ± 0.011
1	0.068 ± 0.010	0.150 ± 0.027	0.320 ± 0.060	0.097 ± 0.019	0.045 ± 0.017	0.458 ± 0.041	0.327 ± 0.043	0.237 ± 0.055	0.141 ± 0.018	0.261 ± 0.058	0.208 ± 0.042	0.138 ± 0.005
2	0.066 ± 0.005	0.143 ± 0.026	0.292 ± 0.069	0.094 ± 0.018	0.034 ± 0.015	$\mathbf{0.429 \pm 0.032}$	0.367 ± 0.134	0.256 ± 0.068	0.165 ± 0.022	0.277 ± 0.057	$\mathbf{0.195 \pm 0.038}$	0.136 ± 0.012
5	0.072 ± 0.008	0.147 ± 0.027	0.217 ± 0.079	$\mathbf{0.086 \pm 0.020}$	$\mathbf{0.030 \pm 0.014}$	0.454 ± 0.055	0.389 ± 0.096	0.255 ± 0.065	0.173 ± 0.033	0.323 ± 0.043	0.247 ± 0.032	0.136 ± 0.014
10	0.082 ± 0.007	0.146 ± 0.020	$\mathbf{0.188 \pm 0.093}$	0.137 ± 0.096	$\mathbf{0.030 \pm 0.024}$	0.451 ± 0.045	0.430 ± 0.136	0.263 ± 0.060	0.148 ± 0.030	0.325 ± 0.067	0.292 ± 0.047	0.136 ± 0.014

A.1.3 Computational Cost

All timing experiments are conducted on a single NVIDIA A100-PCIe 40GB GPU. We report sampling time only: the wall-clock time to generate a population of $N = 256$ candidate designs from pre-trained generative models, excluding surrogate and generative-model training, oracle evaluation, and metric computation. Table 20 reports the cost for PGD, ParetoFlow and ParetoTransport on two representative tasks from each benchmark family. The methods are evaluated using their benchmark configurations: PGD and ParetoFlow use $K = 1000$, whereas ParetoTransport uses $K = 100$. ParetoTransport generates 256 designs in 7.52–7.7 seconds, against 4.68–5.51 seconds for PGD and 10.53–12.05 seconds for ParetoFlow.

Table 20: Sampling time in seconds to generate 256 candidate designs.

	ZDT1	ZDT2	DTLZ2	DTLZ7	RE21	RE37
PGD	5.51	5.06	4.94	4.68	4.73	5.15
ParetoFlow	11.87	12.05	10.53	11.96	12.05	11.68
ParetoTransport (ours)	7.68	7.68	7.65	7.70	7.52	7.57

Table 21: Decomposition of sampling time for ParetoTransport.

	ZDT1	ZDT2	DTLZ2	DTLZ7	RE21	RE37
Total (s)	7.68	7.68	7.65	7.70	7.52	7.57
Sinkhorn (s)	5.95	6.08	5.77	5.73	5.77	5.64
Proxy update (s)	0.43	0.30	0.39	0.48	0.55	0.44
Sinkhorn per iteration (µs)	153	157	149	148	149	145

We further decompose the ParetoTransport sampling loop into Sinkhorn computations, proxy updates, and all remaining operations. With $K = 100$, the sampler performs 99 Euler steps. Because $t_{\text{guided}} = 0.8$, guidance is applied during the final 19 steps. Each guided step performs 10 inner optimization steps and recomputes the transport plan at every inner step, resulting in 190 Sinkhorn solves per generated population. Sinkhorn computations account for 74–79% of the total sampling time, while proxy updates, including block and crowding-based mass assignment, account for 4–7%. The remaining operations comprise surrogate evaluations, evaluation of the Wasserstein and directional terms, gradient computation, optimizer updates, and the 80 unguided Euler steps. The cost of a single Sinkhorn iteration is 145–157 μs and essentially invariant across tasks.

A.2 Training-free guidance of flow-matching models

Flow matching (FM) (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2022) is an efficient and simulation-free framework for generative modeling. The goal is to learn a time-dependent velocity field $u : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ that transforms a simple *initial* distribution p_0 into a complex *terminal* distribution p_1. The flow map $\phi : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ is defined by $\phi_t(x_0) := x(t)$, where $x : [0, 1] \to \mathbb{R}^d$ is the solution of the ordinary differential equation (ODE)

$$\frac{dx(t)}{dt} = u_t\big(x(t)\big), \qquad x(0) = x_0. \tag{8}$$

This flow map induces a probability path $p_t := (\phi_t)_{\#}p_0$ that satisfies the continuity equation $\partial_t p_t + \nabla \cdot (u_t p_t) = 0$. FM learns a model u_t^θ by minimizing the regression loss w.r.t. the ground

truth velocity field u_t,

$$\mathcal{L}_{\mathrm{FM}}(\theta) := \mathbb{E}_{t\sim\mathcal{U}[0,1],\, X_t\sim p_t} \left\| u_t^\theta(X_t) - u_t(X_t) \right\|^2 .$$

Since the velocity field u_t is unknown, the FM loss is intractable. FM therefore introduces a conditional probability path $p_{t|Z}$ with $Z = (X_0, X_1) \sim \pi$, where $\pi \in \Gamma(p_0, p_1)$ is a coupling of $X_0 \sim p_0$ and $X_1 \sim p_1$. A standard choice is an affine conditional probability path $X_t := \alpha_t X_1 + \beta_t X_0$ with scheduler (α_t, β_t) satisfying $\alpha_0 = \beta_1 = 0, \alpha_1 = \beta_0 = 1$. The induced velocity field $u_{t|Z}$ is then given by $u_{t|Z}(X_t) = \dot{\alpha}_t X_1 + \dot{\beta}_t X_0$. This yields a tractable conditional FM loss

$$\mathcal{L}_{\mathrm{CFM}}(\theta) := \mathbb{E}_{t\sim\mathcal{U}[0,1],\, Z\sim\pi,\, X_t\sim p_{t|Z}} \left\| u_t^\theta(X_t) - u_{t|Z}(X_t) \right\|^2 .$$

The marginal and conditional velocity fields are related through

$$u_t(x) = \mathbb{E}\left[u_{t|Z}(X_t) \mid X_t = x \right].$$

Moreover, it has been shown that $\nabla\mathcal{L}_{\mathrm{FM}}(\theta) = \nabla\mathcal{L}_{\mathrm{CFM}}(\theta)$ (Lipman et al., 2023). Therefore, the velocity field u_t^θ can be trained by minimizing the conditional FM loss $\mathcal{L}_{\mathrm{CFM}}(\theta)$. At inference time, sampling is performed by drawing an initial sample $x_0 \sim p_0$ and integrating the ODE in (8) with the learned velocity field.

Training-free guidance. A key challenge in training-free guidance is the approximation of the full-horizon optimization problem by a sequence of single-step subproblems. Recent work (Li et al., 2026; Pourya et al., 2026) further introduces a reverse parameterization that reformulates the optimization in terms of the terminal sample, thereby avoiding differentiation of the terminal loss through the neural network.

The full-horizon optimization problem (Liu et al., 2023) is formulated as

$$\begin{aligned} \min_{\left(\widetilde{x}(t),\, c(t)\right)_{t\in[0,1]}} \quad & J\big(\widetilde{x}(1)\big) + \lambda \int_0^1 \frac{1}{2}\|c(t)\|^2\, dt \\ \text{s.t.} \quad & \widetilde{x}(0) = x_0, \\ & \frac{d\widetilde{x}(t)}{dt} = u_t\big(\widetilde{x}(t)\big) + c(t) \qquad t \in [0,1]. \end{aligned} \tag{9}$$

In practice, the continuous-time dynamics are discretized on a time grid $0 = t_1 < \ldots < t_K = 1$, $\Delta t_k := t_{k+1} - t_k$. Using forward Euler, the full-horizon optimization problem becomes

$$\begin{aligned} \min_{(\widetilde{x}_{t_k})_{k=1}^K,\, (c_{t_k})_{k=1}^{K-1}} \quad & J\big(\widetilde{x}_1\big) + \lambda \sum_{k=1}^{K-1} \frac{\Delta t_k}{2}\|c_{t_k}\|^2 \\ \text{s.t.} \quad & \widetilde{x}_0 = x_0, \\ & \widetilde{x}_{t_{k+1}} = \widetilde{x}_{t_k} + \Delta t_k \left(u_{t_k}(\widetilde{x}_{t_k}) + c_{t_k}\right) \qquad k = 1, \ldots, K-1. \end{aligned} \tag{10}$$

Although Eq. (10) is written in terms of both the trajectory and guidance variables, the Euler recursion uniquely determines the trajectory from the guidance variables once the initial samples are fixed. Hence, the problem can equivalently be viewed as a full-horizon optimization over $(c_{t_k})_{k=1}^{K-1}$ alone.

Prior work (Li et al., 2026; Webber et al., 2026) approximates this discrete-time full-horizon problem by a sequence of single-step subproblems. At step k, future guidance velocities $c_{t_{k+1}}, \ldots, c_{t_{K-1}}$ are ignored, and the posterior mean $\widehat{x}_{1|t_{k+1}}$ is used to approximate the terminal sample resulting from the current guidance step. Hence, rather than jointly optimizing the guidance velocities over the entire sampling path, each subproblem optimizes only c_{t_k}. Given $x_0 \sim p_0$ and $\widetilde{x}_0 := x_0$, the resulting recursion for $k = 1, \ldots, K-1$ is

$$\begin{aligned} c_{\mathrm{opt}} := \arg\min_{c_{t_k}} \quad & J(\widehat{x}_1) + \lambda \frac{\Delta t_k}{2}\|c_{t_k}\|^2 \\ \text{s.t.} \quad & \widehat{x}_1 = \widehat{x}_{1|t_{k+1}}\big(\widetilde{x}_{t_k} + \Delta t_k \left(u_{t_k}(\widetilde{x}_{t_k}) + c_{t_k}\right)\big), \\ \widetilde{x}_{t_{k+1}} := \widetilde{x}_{t_k} + \Delta t_k & \left(u_{t_k}(\widetilde{x}_{t_k}) + c_{\mathrm{opt}}\right). \end{aligned} \tag{11}$$

Since the terminal loss depends on c_{t_k} only through the estimated terminal sample, Li et al. (2026) reformulate each subproblem directly in terminal space. The optimized terminal sample is then mapped back to the sampling path using the local reverse parameterization, denoted here by $\widehat{x}_{t_{k+1}|1}$. The regularization term on the guidance velocity can be equivalently expressed in terms of the terminal sample (Li et al., 2026), resulting in the following recursion for $k = 1, \dots, K-1$:

$$\begin{aligned} \widehat{x}_1 &:= \widehat{x}_{1|t_{k+1}}\big(\widetilde{x}_{t_k} + \Delta t_k u_{t_k}(\widetilde{x}_{t_k})\big), \\ x_{\mathrm{opt}} &:= \arg\min_{x_1} \quad J(x_1) + \lambda \frac{\alpha_{t_{k+1}}^2}{2\Delta t_k} \|x_1 - \widehat{x}_1\|^2, \\ \widetilde{x}_{t_{k+1}} &:= \widehat{x}_{t_{k+1}|1}(x_{\mathrm{opt}}). \end{aligned} \tag{12}$$

A detailed analysis of the approximation error is provided by Li et al. (2026). Pourya et al. (2026) provide an alternative reverse parameterization which re-samples noise at every step (Remark 1).

Posterior mean estimation. Prior work in diffusion (Kim & Ye, 2021; Chung et al., 2023; Kim et al., 2025b) has leveraged Tweedie's formula to obtain the posterior mean estimator of the terminal sample. Under a Gaussian initial distribution and an affine conditional probability path, Tweedie's formula has also been adopted for flow matching (Kim et al., 2025a; Pourya et al., 2026). However, flow matching also admits a direct derivation of the same posterior mean estimator that relies only on the affine conditional probability path, without explicit use of Tweedie's formula (Feng et al., 2025; Li et al., 2026; Webber et al., 2026). This is essential in our setting, where p_0 is uniform rather than Gaussian (Appendix A.4.1).

Lemma 1. *(Feng et al., 2025) Let $Z := (X_0, X_1) \sim \pi \in \Gamma(p_0, p_1)$ and consider the affine conditional probability path $X_t := \alpha_t X_1 + \beta_t X_0, t \in [0,1]$, where (α_t, β_t) is a differentiable scheduler satisfying $\alpha_0 = \beta_1 = 0, \alpha_1 = \beta_0 = 1$. The induced conditional velocity field $u_{t|Z}$ is therefore $u_{t|Z}(X_t) = \dot{\alpha}_t X_1 + \dot{\beta}_t X_0$. Define $\Lambda_t := \alpha_t \dot{\beta}_t - \dot{\alpha}_t \beta_t$. Assume that $\Lambda_t \neq 0$ for all $t \in [0,1]$. Then, for every $x \in \mathbb{R}^d$, the posterior means are given by*

$$\widehat{x}_{0|t}(x) := \mathbb{E}[X_0 \mid X_t = x] = \frac{-\dot{\alpha}_t\, x + \alpha_t\, u_t(x)}{\Lambda_t}.$$

and

$$\widehat{x}_{1|t}(x) := \mathbb{E}[X_1 \mid X_t = x] = \frac{\dot{\beta}_t\, x - \beta_t\, u_t(x)}{\Lambda_t}.$$

Moreover, they satisfy the identity

$$x = \alpha_t \widehat{x}_{1|t}(x) + \beta_t \widehat{x}_{0|t}(x). \tag{13}$$

Corollary 2. *Consider the linear scheduler $\alpha_t := t, \beta_t := 1 - t$ corresponding to the affine conditional probability path $X_t := tX_1 + (1-t)X_0$. Then the posterior means simplify to*

$$\widehat{x}_{0|t}(x) = x - t u_t(x), \qquad \widehat{x}_{1|t}(x) = x + (1-t) u_t(x). \tag{14}$$

Remark 1. *The identity in (13) yields an implicit characterization of the reverse parameterization. For $\alpha_t \neq 0$,*

$$x_1 = \widehat{x}_{1|t}(x_t) \quad \Longleftrightarrow \quad x_t = \alpha_t x_1 + \beta_t \widehat{x}_{0|t}(x_t).$$

Therefore, recovering x_t from a given terminal sample x_1 is equivalent to finding a fixed point of the mapping

$$T_{x_1}(x) := \alpha_t x_1 + \beta_t \widehat{x}_{0|t}(x).$$

If T_{x_1} is a contraction, the Banach fixed-point theorem guarantees that, for any initialization $x^{(0)}$, the iterates

$$x^{(k+1)} = T_{x_1}(x^{(k)})$$

converge to the unique fixed point.

Given a reference intermediate sample x_t^{ref}, Li et al. (2026) approximate the solution of the fixed-point equation by a single iteration initialized at $x^{(0)} := x_t^{\mathrm{ref}}$. Thus, for a given terminal sample x_1,

$$\widehat{x}_{t|1}\big(x_1; x_t^{\mathrm{ref}}\big) := x^{(1)} = T_{x_1}(x^{(0)}) = \alpha_t x_1 + \beta_t \widehat{x}_{0|t}\big(x_t^{\mathrm{ref}}\big).$$

Pourya et al. (2026) instead define the reverse parameterization by

$$\widehat{x}_{t|1}(x_1; x_0) := \alpha_t x_1 + \beta_t x_0, \tag{15}$$

where $x_0 \sim p_0$ is independently sampled at each step.

A.3 Algorithmic details

A.3.1 Particle-level approximation of distribution-level guidance

We derive the particle-level approximation of the distributional guidance problem in Eq. (6). We approximate the probability path $(\widetilde{p}_t)_{t\in[0,1]}$ by the empirical path and, with a slight abuse of notation, write $\widetilde{p}_t = \frac{1}{N}\sum_{i=1}^N \delta_{\widetilde{x}_t^i}$. Thus, the probability path $(\widetilde{p}_t)_{t\in[0,1]}$ is represented by a system of particle trajectories $\{(\widetilde{x}_t^i)_{t\in[0,1]} \mid i = 1,\dots,N\}$. The continuity equation in Eq. (6) then takes the form

$$\frac{d}{dt}\widetilde{x}_t^i = u_t(\widetilde{x}_t^i) + c_t(\widetilde{x}_t^i), \qquad \widetilde{x}_0^i = x_0^i, \qquad i = 1,\dots,N. \tag{16}$$

On the time grid $0 = t_1 < \ldots < t_K = 1$, $\Delta t_k := t_{k+1} - t_k$, the forward Euler discretization of Eq. (16) is

$$\widetilde{x}_{t_{k+1}}^i = \widetilde{x}_{t_k}^i + \Delta t_k \left(u_{t_k}(\widetilde{x}_{t_k}^i) + c_{t_k}(\widetilde{x}_{t_k}^i)\right). \tag{17}$$

Thus, the particle-level time-discrete approximation of the guidance regularization term in Eq. (6) becomes

$$\lambda \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2}\|c_t(x)\|^2 \, d\widetilde{p}_t(x)\, dt = \frac{\lambda}{N}\sum_{i=1}^N \int_0^1 \frac{1}{2}\|c_t(\widetilde{x}_t^i)\|^2 \, dt \approx \frac{\lambda}{N}\sum_{i=1}^N \sum_{k=1}^{K-1} \frac{\Delta t_k}{2}\|c_{t_k}(\widetilde{x}_{t_k}^i)\|^2.$$

The particle-level time-discrete approximation of the distributional guidance problem in Eq. (6) is

$$\begin{aligned} \min_{(\widetilde{x}_{t_k}^i)_{i,k},\,(c_{t_k}^i)_{i,k}} \quad & J\left(\frac{1}{N}\sum_{i=1}^N \delta_{\widetilde{x}_1^i}\right) + \frac{\lambda}{N}\sum_{i=1}^N \sum_{k=1}^{K-1} \frac{\Delta t_k}{2}\|c_{t_k}^i\|^2 \\ \text{s.t.} \quad & \widetilde{x}_0^i = x_0^i, \qquad i = 1,\dots,N, \\ & \widetilde{x}_{t_{k+1}}^i = \widetilde{x}_{t_k}^i + \Delta t_k \left(u_{t_k}(\widetilde{x}_{t_k}^i) + c_{t_k}^i\right), \quad i = 1,\dots,N, \quad k = 1,\dots,K-1. \end{aligned} \tag{18}$$

The single-step approximation (Li et al., 2026) described in Appendix A.2 replaces the corresponding time-discrete full-horizon optimization problem by a sequence of single-step terminal-space subproblems. We apply this approximation to the particle system. Thus, at step k, only the current guidance variables $(c_{t_k}^i)_{i=1}^N$ are considered. For each particle, the terminal estimate is obtained by evaluating the posterior mean map at the unguided Euler update,

$$\widehat{x}_1^i := \widehat{x}_{1|t_{k+1}}\left(\widetilde{x}_{t_k}^i + \Delta t_k u_{t_k}(\widetilde{x}_{t_k}^i)\right), \qquad i = 1,\dots,N.$$

The guidance variables $(c_{t_k}^i)_{i=1}^N$ are then reparameterized by terminal variables $(x_1^i)_{i=1}^N$. The resulting subproblem is

$$\left(x_{\mathrm{opt}}^i\right)_{i=1}^N \in \arg\min_{(x_1^i)_{i=1}^N} \; J\left(\frac{1}{N}\sum_{i=1}^N \delta_{x_1^i}\right) + \frac{\lambda}{N}\sum_{i=1}^N \frac{\alpha_{t_{k+1}}^2}{2\Delta t_k}\left\|x_1^i - \widehat{x}_1^i\right\|^2.$$

Unlike in the single-trajectory setting, the loss functional J acts on the empirical distribution of the terminal particles and therefore couples the terminal variables in a joint optimization. Finally, the optimized terminal particles are mapped back to the sampling path using the local reverse parameterization in Eq. (15),

$$\widetilde{x}_{t_{k+1}}^i := \widehat{x}_{t_{k+1}|1}\big(x_{\mathrm{opt}}^i; \varepsilon_k^i\big), \qquad i = 1,\dots,N.$$

A.3.2 Intermediate proxy distributions

Objective-space normalization. All objective-space computations in ParetoTransport are performed after min–max normalization using statistics computed once from the offline dataset $\mathcal{D}_{\mathrm{off}}$. Since the W_2 term uses Euclidean distances in objective space, normalization ensures that its transport cost is not dominated by differences in objective scales. For objective $r = 1,\dots,m$, let

$$f_r^{\min} := \min_{x\in\mathcal{D}_{\mathrm{off}}} f_r(x), \qquad f_r^{\max} := \max_{x\in\mathcal{D}_{\mathrm{off}}} f_r(x).$$

We define the normalized objective functions

$$\bar{f}_r(x) := \frac{f_r(x) - f_r^{\min}}{f_r^{\max} - f_r^{\min}}, \qquad r = 1, \dots, m,$$

and keep these normalization constants fixed throughout surrogate training, flow model training, and sampling. For simplicity, we write f for the normalized objective function $\bar{f} := (\bar{f}_1, \dots, \bar{f}_m)$.

Crowding distance. For a set $\{y^j\}_{j=1}^M \subset \mathbb{R}^m$, we compute the standard NSGA-II crowding distance (Deb et al., 2002a). For each objective r, the points are sorted according to their r-th objective value. For an interior point y^j, its contribution along objective r is

$$d_r^j := \frac{y_r^{j^+} - y_r^{j^-}}{y_r^{\max} - y_r^{\min}},$$

where j^- and j^+ denote the adjacent points in the ordering of objective r. The total crowding distance is $d^j := \sum_{r=1}^m d_r^j$. As in NSGA-II, boundary points are assigned infinite crowding distance.

Proxy-update operator. We denote by $\mathcal{R}$ the proxy-update operator that combines the current proxy distribution μ_{t_k} with the optimized objective vectors to construct the next proxy distribution, updating its support and assigning probability masses according to local crowding so that sparsely represented regions receive more mass and densely represented regions receive less.

In particular, we maintain one block per offline-front point $z^j \in D(\text{best})$, $j = 1, \dots, M$. Each candidate objective vector y produced at step k is assigned to the block whose seed z^j is nearest to y in the subspace orthogonal to the CHIM normal n, and within that block is scored by its progress along n relative to the seed, $s^j(y) := n^\top(y - z^j)$. Each block j retains a single incumbent, that is, the highest-scoring point assigned to it up to step k, and is updated only when a new candidate improves on the current incumbent, so it is monotone non-decreasing in terms of the score. Let $(y^j)_{j=1}^M$ denote the M current block incumbents.

We then compute the crowding distance d^j for each point. Since boundary points are assigned infinite crowding distance, we replace each infinite value by the largest finite crowding distance,

$$\widetilde{d}^j := \begin{cases} d^j, & d^j < \infty, \\ \max\limits_{\ell:\, d^\ell < \infty} d^\ell, & d^j = \infty. \end{cases}$$

We define $b^j := \frac{\widetilde{d}^j}{\sum_i \widetilde{d}^i}$. The resulting next proxy is $\mu_{t_{k+1}} := \sum_{j=1}^M b^j \delta_{y^j}$.

A.3.3 Loss functional

We derive the implementation details of the loss functional in Eq. (7),

$$J_k(q) = W_2^2(f_\# q, \mu_{t_k}) + \gamma L_n(f_\# q, \mu_{t_k}).$$

Directional transport. We give details on the computation of the normal n in

$$L_n(f_\# q, \mu_{t_k}) = -\big(\mathbb{E}_{Y \sim f_\# q}[Y] - \mathbb{E}_{Y \sim \mu_{t_k}}[Y]\big)^\top n.$$

We compute the CHIM (Convex Hull of Individual Minima) from the individual objective minima in the offline dataset: for $r = 1, \dots, m$, let $x_{\text{IM}}^r \in \arg\min_{x \in \mathcal{D}_{\text{off}}} f_r(x)$ and $z^r := f(x_{\text{IM}}^r) \in \mathbb{R}^m$. The CHIM is $\mathcal{H} := \text{conv}\{z^1, \dots, z^m\}$, and n is a unit normal to its affine hull. Let

$$V := \begin{bmatrix} (z^2 - z^1)^\top \\ \vdots \\ (z^m - z^1)^\top \end{bmatrix} \in \mathbb{R}^{(m-1)\times m}.$$

A normal to $\text{aff}(\mathcal{H})$ satisfies $Vn = 0$. If $z^1, \dots, z^m$ are affinely independent, then $\text{rank}(V) = m-1$ and $\ker(V)$ is one-dimensional; we take $\widetilde{n}$ to be the right singular vector of V spanning $\ker(V)$. For

$m = 2$ this reduces to $\widetilde{n} = (-v_2, v_1)^\top$ with $v := z^2 - z^1$, and for $m = 3$ to $\widetilde{n} = (z^2 - z^1) \times (z^3 - z^1)$. Finally, we normalize $\widetilde{n}$ and resolve its sign by orienting it toward decreasing objectives:

$$\bar{n} := \frac{\widetilde{n}}{\|\widetilde{n}\|}, \qquad n := \begin{cases} \bar{n}, & \bar{n}^\top \mathbf{1}_m \le 0, \\ -\bar{n}, & \text{otherwise.} \end{cases} \tag{19}$$

Eq. (19) requires affinely independent minima, and it guarantees $n^\top \mathbf{1}_m \le 0$, i.e., descent on average, but not $n_r \le 0$ for every $r = 1, \dots, m$. For $m \ge 3$, the exact normal of $\mathrm{aff}(\mathcal{H})$ may have a positive component on asymmetric fronts, a property already noted by Das & Dennis (1998). If either the minima are affinely dependent, $\mathrm{rank}(V) < m - 1$, so that the normal is not unique, or n has a positive component, we use instead the quasi-normal of Das & Dennis (1998),

$$n := -\frac{\Phi \mathbf{1}_m}{\|\Phi \mathbf{1}_m\|}, \qquad \Phi := [\, z^1 - z^{\mathrm{ideal}} \ \cdots \ z^m - z^{\mathrm{ideal}} \,], \qquad z^{\mathrm{ideal}} := (z_1^1, \dots, z_m^m)^\top. \tag{20}$$

The quasi-normal does not depend on V, hence is defined for any configuration of the minima other than $z^1 = \cdots = z^m = z^{\mathrm{ideal}}$, and it is componentwise non-positive since every column of Φ is. The direction n is computed once from the offline dataset and kept fixed throughout guided sampling.

Wasserstein matching. We approximate the discrete optimal transport problem defining $W_2^2(f_{\#}q, \mu_{t_k})$ by an entropically regularized optimal transport problem. For candidate terminal particles $X := (x_1^i)_{i=1}^N$, the terminal design-space distribution is represented by the empirical distribution $q := \frac{1}{N} \sum_{i=1}^N \delta_{x_1^i}$. The particle-level optimization acts on the particle locations x_1^i, while each particle retains mass $\frac{1}{N}$. The induced objective-space distribution is $f_{\#}q = \frac{1}{N} \sum_{i=1}^N \delta_{f(x_1^i)}$. Thus, the source marginal in the optimal transport problem is $a := \frac{1}{N} \mathbf{1}_N$. In contrast, the target marginal $b_{t_k} := (b_{t_k}^j)_{j=1}^M$ is determined by the crowding-based proxy-update operator. For simplicity, at a fixed step k, we write $y := y_{t_k}$ and $b := b_{t_k}$.

We define the quadratic cost matrix $C(X) \in \mathbb{R}^{N \times M}$ by $C_{ij}(X) := \|f(x_1^i) - y^j\|^2$. The set of admissible couplings is $\mathcal{U}(a, b) := \{P \in \mathbb{R}_+^{N \times M} : P \mathbf{1}_M = a,\ P^\top \mathbf{1}_N = b\}$. The entropically regularized optimal transport problem is

$$\mathcal{W}_\varepsilon(X) := \min_{P \in \mathcal{U}(a,b)} F_\varepsilon(X, P), \quad F_\varepsilon(X, P) := \langle P, C(X) \rangle + \varepsilon \sum_{i=1}^N \sum_{j=1}^M P_{ij}(\log P_{ij} - 1). \tag{21}$$

For $\varepsilon > 0$, $F_\varepsilon(X, \cdot)$ is strictly convex and therefore admits a unique minimizer $P_\varepsilon^\star(X)$ (Peyré & Cuturi, 2019). Hence, Danskin's theorem yields

$$\nabla_X \mathcal{W}_\varepsilon(X) = \nabla_X F_\varepsilon\big(X, P_\varepsilon^\star(X)\big),$$

where $P_\varepsilon^\star(X)$ is held fixed in the derivative on the right-hand side. Thus, gradients with respect to the terminal particles can be computed without differentiating through the optimal coupling.

In practice, we approximate the coupling using the log-domain Sinkhorn solver implemented in the POT library (Flamary et al., 2021), using $\varepsilon = 10^{-3}$. Given the resulting coupling P, we optimize the weighted quadratic transport cost

$$\langle P, C(X) \rangle = \sum_{i=1}^N \sum_{j=1}^M P_{ij} \left\| f(x_1^i) - y^j \right\|^2$$

with respect to the terminal particles X.

A.4 Implementation details

We provide details on the implementation of ParetoTransport and the datasets used in our experiments.

A.4.1 Training Details

We min-max normalize the design space, using the statistics of the offline dataset that are fixed throughout training and evaluation.

For the surrogate models, we adopt the Vanilla Multiple-Models configuration of Off-MOO-Bench (Xue et al., 2024) and train one surrogate per objective. Each surrogate is a three-layer MLP with two 2048-dimensional hidden layers and ReLU activations, trained on the offline dataset with the mean squared error loss and Adam (initial learning rate 10^{-3}, decay 0.98 per epoch) for 200 epochs with batch size 128.

For the flow-matching model, we use an architecture based on ParetoFlow (Yuan et al., 2025): a four-layer MLP with three hidden layers and SeLU activations. We use a hidden width of 256. The model is trained for up to 1000 epochs with early stopping (patience 20) using Adam (learning rate 10^{-3}) and batch size 512. We use the affine scheduler $(\alpha_t, \beta_t) := (t, 1-t)$ and define p_0 as the uniform distribution over $[0,1]^d$.

At inference, we integrate the learned velocity field with explicit Euler on a uniform grid of $K = 100$ points and generate a population of $N = 256$ designs. Following Yuan et al. (2025), we set $t_{\text{guided}} := 0.8$. At each guided step, we run 10 Adam steps with learning rate 0.01 for the inner optimization problem. We set $\gamma = 0.5$ and $\lambda = 10^{-5}$.

The proxy is initialized with a support of $M = 256$ points selected from the offline dataset and updated after each guided step (Appendix A.3.2). If $|D(\text{best})| > M$, we keep the M points of $D(\text{best})$ with the largest crowding distance; if $|D(\text{best})| < M$, we extend $D(\text{best})$ with points from successive non-domination ranks until M points are reached, so that the initial proxy is then supported on this extended set rather than on $D(\text{best})$ alone.

A.4.2 MOO METRICS

We provide details on the evaluation metrics. All metrics are computed in the min–max normalized objective space, using the statistics of the offline dataset that are fixed throughout training and evaluation.

Hypervolume (HV) (Zitzler & Thiele, 1999) measures m-dimensional volume of the objective-space region dominated by the solution set $A \subset \mathbb{R}^m$ and bounded by a reference point $r \in \mathbb{R}^m$. The reference point is selected such that $a_i \le r_i$ for all $a \in A$ and $i = 1, \dots, m$. The hypervolume is defined as

$$\mathrm{HV}(A, r) := \lambda\left(\bigcup_{a \in A} [a_1, r_1] \times [a_2, r_2] \times \cdots \times [a_m, r_m] \right), \tag{22}$$

where λ denotes the Lebesgue measure in $\mathbb{R}^m$. Generational Distance (GD) (Van Veldhuizen & Lamont, 2000; Schütze et al., 2012) measures the root mean squared distance from points in the solution set A to their closest points in a discrete approximation Z of the Pareto front,

$$\mathrm{GD}(A, Z) := \sqrt{\frac{1}{|A|} \sum_{a \in A} d(a, Z)^2}. \tag{23}$$

Analogously, the Inverted Generational Distance (IGD) (Coello Coello & Reyes Sierra, 2004; Schütze et al., 2012) measures the root mean squared distance from points in a discrete approximation Z of the Pareto front to their closest points in the solution set A, $\mathrm{IGD}(A, Z) := \mathrm{GD}(Z, A)$. HV can be strongly influenced by individual solutions, whereas GD and IGD aggregate distances across the solution set and Pareto front approximation, respectively, providing complementary measures of convergence and coverage.

We additionally report the Wasserstein distance. Let μ_A and μ_Z denote the empirical distributions, with uniform weights, of the solution set A and of the reference set Z,

$$\mu_A \;=\; \frac{1}{|A|} \sum_{a \in A} \delta_a, \qquad \mu_Z \;=\; \frac{1}{|Z|} \sum_{z \in Z} \delta_z. \tag{24}$$

With the cost matrix $C_{ij} = \|a^i - z^j\|^2$, we solve the entropically regularized optimal transport problem in Eq. (21) using the log-domain Sinkhorn solver of the POT library (Flamary et al., 2021) with $\varepsilon = 10^{-3}$, and report the resulting transport cost $\sqrt{\langle P^\star, C \rangle}$.

A.4.3 DATASET DETAILS

We provide further details on the datasets[6] in Tables 22 and 23. We use the original reference points provided in the code repository of Xue et al. (2024).

Table 22: Synthetic tasks; dataset information and reference point for HV computation.

Name	d	m	Pareto Front Shape	Reference Point
DTLZ1	7	3	Linear	(507.46516068, 502.09050332, 516.68985527)
DTLZ2	10	3	Concave	(2.51925352, 2.53011955, 2.66682285)
DTLZ3	10	3	Concave	(1548.83497615, 1459.58638514, 1518.61926977)
DTLZ4	10	3	Concave	(2.75747516, 2.57674881, 2.52558196)
DTLZ5	10	3	Concave	(2.40756462, 2.36937324, 2.45008585)
DTLZ6	10	3	Concave	(8.90471718, 8.89404128, 8.88893072)
DTLZ7	10	3	Disconnected	(0.99999540, 0.99999772, 30.79227037)
ZDT1	30	2	Convex	(0.99999809, 7.82506630)
ZDT2	30	2	Concave	(0.99999706, 9.74316166)
ZDT3	30	2	Disconnected	(0.99999954, 9.83643758)
ZDT4	10	2	Convex	(0.99980978, 273.25533168)
ZDT6	10	2	Concave	(1.00000000, 9.34365208)

Table 23: Real-engineering tasks; dataset information and reference point for HV computation.

Name	d	m	Pareto Front Shape	Reference Point
RE21 (Four bar truss design)	4	2	Convex	(2971.23988, 0.0485512338)
RE22 (Reinforced concrete beam design)	3	2	Mixed	(829.079443, 2407217.25)
RE23 (Pressure vessel design)	4	2	Mixed, Disconnected	(713710.875, 1288669.78054)
RE24 (Hatch cover design)	2	2	Convex	(5997.8316325, 43.67584229)
RE25 (Coil compression spring design)	3	2	Mixed, Disconnected	(124.795202, 10038735.0)
RE31 (Two bar truss design)	3	3	Unknown	(808.852742, 6893375.82, 6793450.00)
RE32 (Welded beam design)	4	3	Unknown	(290.661885, 16552.4628, 388265024.0)
RE33 (Disc brake design)	4	3	Unknown	(8.01164324, 8.83604223, 2343.29711914)
RE34 (Vehicle crashworthiness design)	5	3	Unknown	(1702.51811, 11.6807224, 0.263918844)
RE35 (Speed reducer design)	7	3	Unknown	(7050.78959905, 1696.66697789, 397.83456421)
RE36 (Gear train design)	4	3	Concave, Disconnected	(10.21185714, 60.0, 0.97335988)
RE37 (Rocket injector design)	4	3	Unknown	(0.98949120096, 0.956587924661, 0.987530948586)

A.5 THEORETICAL RESULTS

Corollary 1. *For $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^m)$, $\mathrm{GD}(\mu\|\nu) \le W_2(\mu,\nu)$ and $\mathrm{IGD}(\mu\|\nu) \le W_2(\mu,\nu)$.*

Proof. Let $\pi \in \Gamma(\mu,\nu)$. Since $y \in \operatorname{supp}\nu$ for π-almost every (x,y), it follows that $d(x,\operatorname{supp}\nu) \le \|x-y\|$ for π-a.e. (x,y). Therefore,

$$\mathrm{GD}^2(\mu\|\nu) \le \int_{\mathbb{R}^m\times\mathbb{R}^m} \|x-y\|^2 \, d\pi(x,y).$$

Taking the infimum over $\pi \in \Gamma(\mu,\nu)$ yields $\mathrm{GD}(\mu\|\nu) \le W_2(\mu,\nu)$. Since $\mathrm{IGD}(\mu\|\nu) = \mathrm{GD}(\nu\|\mu)$ and W_2 is symmetric, the IGD bound follows. □

Remark 2 (One-sided optimal transport interpretation)**.** *GD and IGD admit a one-sided optimal-transport interpretation in which one marginal is relaxed subject to a hard support constraint. In contrast, $W_2(\mu,\nu)$ fixes both marginals and therefore also accounts for probability-mass allocation. Indeed, as a direct consequence of Lemma 31 of Vayer & Gribonval (2023), attributed therein to Cañas & Rosasco (2012), we have*

$$\mathrm{GD}^2(\mu\|\nu) = \min_{\rho\in\mathcal{P}_2(\operatorname{supp}\nu)} W_2^2(\mu,\rho), \qquad \mathrm{IGD}^2(\mu\|\nu) = \min_{\rho\in\mathcal{P}_2(\operatorname{supp}\mu)} W_2^2(\nu,\rho).$$

Thus, Corollary 1 follows also by evaluating the respective minima at $\rho=\nu$ and $\rho=\mu$.

[6] We focus on tasks with $m \le 3$, and leave the extension to the many-objective setting ($m \ge 4$) for future work. This choice is motivated by the complex structure of high-dimensional Pareto fronts.

Remark 3 (Uniform reference distribution). *Let $\mathcal{M}^* \subset \mathbb{R}^m$ be nonempty and compact, let d denote its Hausdorff dimension (generically $d = m - 1$), and assume $0 < \mathcal{H}^d(\mathcal{M}^*) < \infty$, where $\mathcal{H}^d$ is the d-dimensional Hausdorff measure. We define the uniform reference distribution on $\mathcal{M}^*$ as the normalized d-dimensional Hausdorff measure,*

$$\mu^* := \frac{\mathcal{H}^d \llcorner \mathcal{M}^*}{\mathcal{H}^d(\mathcal{M}^*)}. \tag{25}$$

Since $\mathcal{M}^*$ is compact, $\mu^* \in \mathcal{P}_2(\mathbb{R}^m)$. Eq. (25) requires neither smoothness nor connectedness of $\mathcal{M}^*$: each connected component of a disconnected front receives mass in proportion to its d-dimensional size, and the dimension adapts to degenerate fronts (a front that is a curve in $\mathbb{R}^3$ has $d = 1$). If $\mathcal{M}^*$ is finite, then $d = 0$, $\mathcal{H}^0$ is the counting measure, and Eq. (25) is the discrete uniform distribution on $\mathcal{M}^*$; the evaluation target μ_Z of Eq. (24) is Eq. (25) applied to the reference set Z.

Technical assumptions:

Assumption 1 (Projection condition). *The Pareto front $M^* \subset \mathbb{R}^m$ is nonempty and closed. There exists a set $U \supset M^*$ such that every $y \in U$ has a unique nearest point in M^*, denoted by $\Pi_{M^*}(y) := \operatorname{argmin}_{z \in M^*} \|y - z\|$.*

Assumption 2 (Moment and concentration condition). *The offline front and Pareto front distributions satisfy $\mu_0, \mu^* \in \mathcal{P}_2(\mathbb{R}^m)$ and $\mu_0(U) = 1$.*

Assumption 3 (Invariance under transport). *For a fixed $\eta \in (0, 1]$, define $T_\eta(y) := (1 - \eta)y + \eta\Pi_{M^*}(y)$, $y \in U$. T_η satisfies $T_\eta(U) \subseteq U$.*

Assumption 1 holds with $U = \mathbb{R}^m$ if $\mathcal{M}^*$ is convex. In general, take U to be the set of all points with a unique nearest point in $\mathcal{M}^*$; it contains $\mathcal{M}^*$, requires no convexity, and need not be connected. The excluded set of points with several nearest points has Lebesgue measure zero for any closed $\mathcal{M}^*$; hence the condition $\mu_0(U) = 1$ in Assumption 2 fails only if an offline point lies exactly in this null set. For U the set of all points with a unique nearest point in $\mathcal{M}^*$, Assumption 3 follows from Assumption 1: moving $y \in U$ toward its unique nearest point keeps that point the unique nearest one, so $T_\eta(y) \in U$ and hence $T_\eta(U) \subseteq U$.

Theorem 1. *Let $(\mu_k)_{k \geq 0}$ denote the sequence of intermediate proxy distributions defined by the iterative procedure in Section 5. If Step (i) achieves exact matching, $W_2(f_\# q_{k+1}, \mu_k) = 0$ for all k, and the technical Assumptions (1)–(3) hold, then $W_2(\mu_k, \bar{\mu}) = (1 - \eta)^k W_2(\mu_0, \bar{\mu})$, and consequently*

$$W_2(\mu_k, \mu^*) \leq (1 - \eta)^k W_2(\mu_0, \bar{\mu}) + W_2(\bar{\mu}, \mu^*). \tag{26}$$

Proof. Since W_2 is a metric on $\mathcal{P}_2(\mathbb{R}^m)$, exact matching in Step (i) implies $f_\# q_{k+1} = \mu_k$, and hence by Step (ii), $\mu_{k+1} = (T_\eta)_\# \mu_k$. In particular, $\mu_k = (T_\eta^k)_\# \mu_0$.

Step 1: We first establish $T_\eta^k = (1 - \xi_k)\,\mathrm{Id} + \xi_k \Pi_{M^*}$ for $\xi_k := 1 - (1 - \eta)^k$. By Assumptions 1 and 3,

$$\Pi_{M^*}(T_\eta(y)) = \Pi_{M^*}(y), \quad y \in U. \tag{27}$$

Indeed, fix $y \in U$, let $z := \Pi_{M^*}(y)$ and $r := \|y - z\| = d(y, M^*)$, and set $w := \Pi_{M^*}(T_\eta(y))$. The latter is well defined by Assumption 3. Since $w \in M^*$, $r \leq \|y - w\|$. Since w is a nearest point to $T_\eta(y)$ and $z \in M^*$,

$$\|y - w\| \leq \|y - T_\eta(y)\| + \|T_\eta(y) - w\| \leq \eta r + (1 - \eta)r = r.$$

Thus $\|y - w\| = r$, so w is also a nearest point in M^* to y. By uniqueness in Assumption 1, $w = z$, proving Eq. (27). By Assumption 3, the identity Eq. (27) may be iterated. A direct induction therefore gives

$$T_\eta^k(y) = (1 - \xi_k)y + \xi_k \Pi_{M^*}(y), \qquad y \in U. \tag{28}$$

Step 2: Assumptions 2 and 3, together with $\mu_k = (T_\eta^k)_\# \mu_0$, imply $\mu_k(U) = 1$ for all k. Moreover, all iterates μ_k belong to $\mathcal{P}_2(\mathbb{R}^m)$. Indeed, fixing any $z_0 \in M^*$, for $y \in U$,

$$\|T_\eta(y)\| \leq \|y\| + \|y - T_\eta(y)\| = \|y\| + \eta\|y - \Pi_{\mathcal{M}^*}(y)\| \leq (1 + \eta)\|y\| + \eta\|z_0\|,$$

so finite second moments are preserved by $(T_\eta)_\#$.

Step 3: We now relate the iteration to a Wasserstein geodesic. Since M^* is nonempty and closed by Assumption 1, there exists a Borel measurable nearest-point selection $P_{M^*} : \mathbb{R}^m \to M^*$. On U the nearest point is unique, hence $P_{M^*}(y) = \Pi_{M^*}(y)$, $y \in U$. Since $\mu_0(U) = 1$ by Assumption 2, $(P_{M^*})_{\#}\mu_0 = (\Pi_{M^*})_{\#}\mu_0 = \bar{\mu}$.

By Lemma 31 of Vayer & Gribonval (2023), it holds that

$$W_2^2(\mu_0, \bar{\mu}) = \int_{\mathbb{R}^m} \|y - P_{M^*}(y)\|^2 \, d\mu_0(y), \tag{29}$$

and the coupling $\pi := (\mathrm{Id}, P_{M^*})_{\#}\mu_0 \in \Gamma(\mu_0, \bar{\mu})$ is an optimal transport plan between μ_0 and $\bar{\mu}$. Since $\mu_0 \in \mathcal{P}_2(\mathbb{R}^m)$ by Assumption 2, and $\bar{\mu} \in \mathcal{P}_2(\mathbb{R}^m)$, we may apply the displacement-interpolation theorem 5.27 of Santambrogio (2015) to π. Thus

$$\mu^{(s)} := \big((1-s)\,\mathrm{pr}_1 + s\,\mathrm{pr}_2\big)_{\#}\pi, \qquad s \in [0,1], \tag{30}$$

is a constant-speed W_2-geodesic from μ_0 to $\bar{\mu}$, that is,

$$W_2(\mu^{(s)}, \bar{\mu}) = (1-s) W_2(\mu_0, \bar{\mu}), \qquad s \in [0,1]. \tag{31}$$

Using the Monge form of π together with $(P_{M^*})_{\#}\mu_0 = (\Pi_{M^*})_{\#}\mu_0$ and $\mu_0(U) = 1$, we may equivalently write

$$\mu^{(s)} = \big((1-s)\,\mathrm{Id} + s\Pi_{M^*}\big)_{\#}\mu_0. \tag{32}$$

Combining $\mu_k = (T_\eta^k)_{\#}\mu_0$, Eq. (28), and Eq. (32) yields $\mu_k = \mu^{(\xi_k)}$. Therefore, by Eq. (31),

$$W_2(\mu_k, \bar{\mu}) = W_2(\mu^{(\xi_k)}, \bar{\mu}) = (1-\xi_k) W_2(\mu_0, \bar{\mu}) = (1-\eta)^k W_2(\mu_0, \bar{\mu}).$$

Finally, $\mu^* \in \mathcal{P}_2(\mathbb{R}^m)$ by Assumption 2, so the triangle inequality for W_2 gives

$$W_2(\mu_k, \mu^*) \le W_2(\mu_k, \bar{\mu}) + W_2(\bar{\mu}, \mu^*) = (1-\eta)^k W_2(\mu_0, \bar{\mu}) + W_2(\bar{\mu}, \mu^*),$$

which proves the claim. □

Remark 4 (Relation to the offline-frontier shift)**.** *Holly et al. (2026) quantify the displacement of an objective-space distribution μ from the Pareto front by $s(\mu) := \int \|y - \Pi_{\mathcal{M}^*}(y)\|^2 \, d\mu(y)$. Since $\|y - \Pi_{\mathcal{M}^*}(y)\| = d(y, \mathcal{M}^*)$ and $\operatorname{supp}\mu^* = \mathcal{M}^*$, both $s(\mu_0)$ and the right-hand side of Eq. (29) equal $\int d(y, \mathcal{M}^*)^2 \, d\mu_0(y) = \mathrm{GD}^2(\mu_0 \| \mu^*)$; hence $W_2(\mu_0, \bar{\mu}) = \mathrm{GD}(\mu_0 \| \mu^*) = \sqrt{s(\mu_0)}$. By Eq. (27), $\bar{\mu} = (\Pi_{\mathcal{M}^*})_{\#}\mu_k$ for every k, so the same identity gives $W_2(\mu_k, \bar{\mu}) = \sqrt{s(\mu_k)}$, and Theorem 1 states that the iterative procedure contracts the offline-frontier shift geometrically: $s(\mu_k) = (1-\eta)^{2k} s(\mu_0)$.*

A.6 Additional results

We report complete per-task results. For each task, methods are ranked in ascending (GD, IGD, W_2) or descending (HV) order of their mean metric value, with any method for which the metric is undefined (NaN or $\pm\infty$) assigned the worst possible rank for that task. Ties are broken using the standard fractional (average) rank. Avg. Rank is the mean of these per-task ranks across all tasks in a benchmark family.

Table 24: HV results on ZDT (↑). Bold = best per column; underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	Avg. Rank
D(best)	4.168	4.677	5.147	5.457	4.608	–
MultiHead-Vanilla	4.823 ± 0.030	5.612 ± 0.098	5.637 ± 0.131	4.938 ± 0.237	4.774 ± 0.003	4.100
MultiHead-PcGrad	4.831 ± 0.028	5.605 ± 0.047	5.524 ± 0.048	3.239 ± 0.678	4.776 ± 0.013	6.200
MultiHead-GradNorm	4.741 ± 0.034	5.483 ± 0.115	5.589 ± 0.192	3.489 ± 0.665	4.709 ± 0.025	8.100
MultipleModels-Vanilla	4.812 ± 0.028	5.600 ± 0.043	5.536 ± 0.140	4.991 ± 0.067	4.774 ± 0.004	5.900
MultipleModels-COM	4.679 ± 0.112	5.093 ± 0.131	5.530 ± 0.072	5.260 ± 0.193	3.435 ± 0.750	8.800
MultipleModels-IOM	4.763 ± 0.033	5.530 ± 0.101	5.681 ± 0.038	5.338 ± 0.174	4.777 ± 0.005	4.000
MultipleModels-ICT	4.748 ± 0.004	5.428 ± 0.004	5.560 ± 0.049	4.830 ± 0.243	4.723 ± 0.051	7.600
MultipleModels-RoMA	4.817 ± 0.029	5.496 ± 0.067	**5.894 ± 0.027**	3.679 ± 0.049	2.597 ± 0.027	7.000
MultipleModels-Trimentoring	4.821 ± 0.002	5.548 ± 0.041	4.569 ± 0.083	4.557 ± 0.556	4.045 ± 0.122	8.000
ParetoFlow	4.136 ± 0.030	4.695 ± 0.071	5.171 ± 0.061	5.104 ± 0.125	4.683 ± 0.068	9.600
PGD	4.567 ± 0.045	5.367 ± 0.064	5.540 ± 0.066	4.999 ± 0.085	**4.823 ± 0.004**	6.800
ParetoTransport (ours)	**4.837 ± 0.022**	**5.654 ± 0.022**	5.589 ± 0.069	**5.483 ± 0.004**	4.822 ± 0.007	**1.900**

Table 25: GD results on ZDT (↓). Bold = best per column; underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	Avg. Rank
D(best)	0.378	0.433	0.382	0.051	0.064	–
MultiHead-Vanilla	0.032 ± 0.017	**0.043 ± 0.048**	0.154 ± 0.031	0.597 ± 0.108	0.113 ± 0.016	4.000
MultiHead-PcGrad	**0.011 ± 0.002**	0.091 ± 0.048	0.255 ± 0.018	0.935 ± 0.221	0.074 ± 0.120	5.400
MultiHead-GradNorm	0.098 ± 0.084	0.103 ± 0.058	0.169 ± 0.010	0.960 ± 0.107	0.098 ± 0.053	7.100
MultipleModels-Vanilla	0.042 ± 0.019	0.074 ± 0.017	0.190 ± 0.021	0.494 ± 0.025	0.096 ± 0.011	3.800
MultipleModels-COM	0.072 ± 0.038	0.237 ± 0.072	0.234 ± 0.018	0.413 ± 0.077	0.301 ± 0.092	7.800
MultipleModels-IOM	0.084 ± 0.013	0.109 ± 0.037	0.149 ± 0.014	0.411 ± 0.037	0.112 ± 0.020	5.400
MultipleModels-ICT	0.049 ± 0.011	0.103 ± 0.009	0.202 ± 0.004	0.646 ± 0.075	0.252 ± 0.063	6.800
MultipleModels-RoMA	0.043 ± 0.018	0.113 ± 0.009	**0.086 ± 0.011**	0.972 ± 0.022	0.614 ± 0.028	7.400
MultipleModels-Trimentoring	0.049 ± 0.008	0.185 ± 0.032	0.702 ± 0.041	0.851 ± 0.216	0.513 ± 0.023	9.500
ParetoFlow	0.363 ± 0.019	0.455 ± 0.010	0.354 ± 0.019	0.395 ± 0.008	0.087 ± 0.024	8.000
PGD	0.275 ± 0.041	0.184 ± 0.032	0.247 ± 0.010	0.565 ± 0.006	0.515 ± 0.041	9.200
ParetoTransport (ours)	0.046 ± 0.045	0.076 ± 0.021	0.235 ± 0.014	**0.333 ± 0.034**	**0.016 ± 0.002**	**3.600**

Table 26: IGD results on ZDT (↓). Bold = best per column; underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	Avg. Rank
D(best)	0.359	0.441	0.367	0.068	0.071	–
MultiHead-Vanilla	0.049 ± 0.076	0.094 ± 0.169	0.162 ± 0.108	0.607 ± 0.060	0.022 ± 0.002	5.400
MultiHead-PcGrad	0.036 ± 0.057	0.057 ± 0.055	0.194 ± 0.027	0.939 ± 0.231	0.020 ± 0.003	5.400
MultiHead-GradNorm	0.090 ± 0.044	0.100 ± 0.052	0.223 ± 0.167	0.879 ± 0.061	0.057 ± 0.016	8.400
MultipleModels-Vanilla	0.058 ± 0.060	0.047 ± 0.020	0.232 ± 0.159	0.584 ± 0.020	0.022 ± 0.001	5.600
MultipleModels-COM	0.159 ± 0.159	0.288 ± 0.046	0.196 ± 0.051	0.423 ± 0.152	0.317 ± 0.132	8.200
MultipleModels-IOM	0.059 ± 0.020	0.081 ± 0.034	0.098 ± 0.014	0.524 ± 0.031	0.022 ± 0.001	4.200
MultipleModels-ICT	0.072 ± 0.011	0.489 ± 0.039	0.147 ± 0.009	0.622 ± 0.087	0.155 ± 0.185	7.800
MultipleModels-RoMA	0.031 ± 0.024	0.420 ± 0.161	**0.017 ± 0.005**	1.024 ± 0.014	0.505 ± 0.002	7.600
MultipleModels-Trimentoring	**0.022 ± 0.002**	0.084 ± 0.027	0.614 ± 0.035	0.699 ± 0.146	0.420 ± 0.049	7.600
ParetoFlow	0.342 ± 0.010	0.465 ± 0.023	0.336 ± 0.026	0.541 ± 0.019	0.224 ± 0.025	9.600
PGD	0.155 ± 0.036	0.138 ± 0.019	0.173 ± 0.042	**0.312 ± 0.033**	0.023 ± 0.010	6.200
ParetoTransport (ours)	0.030 ± 0.040	**0.028 ± 0.010**	0.154 ± 0.021	0.329 ± 0.054	**0.017 ± 0.002**	**2.000**

Table 27: W_2 results on ZDT (↓). Bold = best per column; underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	Avg. Rank
D(best)	0.385	0.475	0.405	0.433	0.171	–
MultiHead-Vanilla	0.194 ± 0.144	**0.130 ± 0.219**	0.306 ± 0.149	0.802 ± 0.084	0.195 ± 0.036	4.000
MultiHead-PcGrad	**0.088 ± 0.106**	0.196 ± 0.175	0.290 ± 0.025	1.085 ± 0.203	**0.087 ± 0.120**	4.200
MultiHead-GradNorm	0.222 ± 0.133	0.221 ± 0.057	0.343 ± 0.106	1.038 ± 0.056	0.259 ± 0.126	7.000
MultipleModels-Vanilla	0.328 ± 0.081	0.224 ± 0.128	0.375 ± 0.106	0.724 ± 0.019	0.186 ± 0.022	6.400
MultipleModels-COM	0.363 ± 0.154	0.440 ± 0.066	0.342 ± 0.052	0.629 ± 0.091	0.442 ± 0.176	7.600
MultipleModels-IOM	0.224 ± 0.049	0.134 ± 0.052	0.175 ± 0.017	0.662 ± 0.024	0.258 ± 0.036	4.200
MultipleModels-ICT	0.332 ± 0.045	0.522 ± 0.002	0.261 ± 0.008	0.836 ± 0.062	0.395 ± 0.144	7.600
MultipleModels-RoMA	0.089 ± 0.033	0.507 ± 0.040	**0.137 ± 0.033**	1.122 ± 0.019	0.776 ± 0.022	7.400
MultipleModels-Trimentoring	0.218 ± 0.027	0.448 ± 0.061	0.731 ± 0.048	0.960 ± 0.153	0.657 ± 0.021	9.200
ParetoFlow	0.491 ± 0.012	0.633 ± 0.006	0.471 ± 0.014	0.650 ± 0.005	0.417 ± 0.004	9.400
PGD	0.333 ± 0.058	0.344 ± 0.022	0.447 ± 0.045	0.579 ± 0.007	0.591 ± 0.032	7.800
ParetoTransport (ours)	0.210 ± 0.112	0.191 ± 0.063	0.328 ± 0.029	**0.550 ± 0.012**	0.108 ± 0.017	**3.200**

Table 28: HV results on DTLZ (↑). Bold = best per column; underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7	Avg. Rank
D(best)	10.614	12.427	9.897	17.503	10.692	10.671	8.992	–
MultiHead-Vanilla	10.643 ± 0.002	12.444 ± 0.000	9.882 ± 0.004	17.567 ± 0.091	10.758 ± 0.001	10.900 ± 0.019	10.715 ± 0.027	3.714
MultiHead-PcGrad	10.644 ± 0.001	12.443 ± 0.001	9.895 ± 0.004	17.533 ± 0.014	**10.759 ± 0.001**	10.918 ± 0.011	10.598 ± 0.028	3.071
MultiHead-GradNorm	10.537 ± 0.239	12.439 ± 0.001	9.657 ± 0.307	17.451 ± 0.190	9.833 ± 2.028	10.848 ± 0.060	9.654 ± 0.735	9.000
MultipleModels-Vanilla	10.644 ± 0.001	12.445 ± 0.000	9.886 ± 0.003	17.046 ± 0.179	10.758 ± 0.000	10.928 ± 0.008	10.535 ± 0.079	4.357
MultipleModels-COM	10.642 ± 0.002	12.433 ± 0.001	9.877 ± 0.006	17.077 ± 0.381	10.747 ± 0.003	10.786 ± 0.017	9.809 ± 0.273	7.643
MultipleModels-IOM	10.643 ± 0.002	12.443 ± 0.001	**9.898 ± 0.000**	16.937 ± 0.203	10.757 ± 0.001	10.886 ± 0.008	10.544 ± 0.126	5.214
MultipleModels-ICT	10.644 ± 0.001	12.429 ± 0.003	9.855 ± 0.050	17.109 ± 0.251	10.688 ± 0.011	10.781 ± 0.022	10.342 ± 0.092	7.143
MultipleModels-RoMA	10.641 ± 0.002	12.349 ± 0.014	9.878 ± 0.002	15.201 ± 0.490	10.594 ± 0.033	10.742 ± 0.015	10.596 ± 0.036	8.714
MultipleModels-Trimentoring	10.642 ± 0.007	11.854 ± 0.892	9.326 ± 0.020	17.107 ± 0.040	10.650 ± 0.004	10.778 ± 0.049	10.392 ± 0.423	9.071
ParetoFlow	10.626 ± 0.006	12.253 ± 0.075	9.792 ± 0.054	17.508 ± 0.005	10.546 ± 0.046	10.686 ± 0.039	8.875 ± 0.099	9.857
PGD	10.645 ± 0.001	12.387 ± 0.024	9.889 ± 0.002	14.504 ± 0.074	10.369 ± 0.357	10.336 ± 0.204	9.767 ± 0.141	8.571
ParetoTransport (ours)	**10.647 ± 0.000**	**12.448 ± 0.000**	9.896 ± 0.002	**17.612 ± 0.010**	10.757 ± 0.000	**11.045 ± 0.009**	**10.749 ± 0.062**	**1.643**

Table 29: GD results on DTLZ (↓). Bold = best per column; underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7	Avg. Rank
D(best)	0.601	0.102	0.341	0.185	0.131	0.759	0.490	–
MultiHead-Vanilla	0.469 ± 0.021	0.066 ± 0.014	0.531 ± 0.035	0.275 ± 0.037	0.064 ± 0.006	0.555 ± 0.027	0.190 ± 0.022	5.929
MultiHead-PcGrad	0.414 ± 0.032	0.066 ± 0.015	0.435 ± 0.020	0.321 ± 0.048	0.049 ± 0.007	0.521 ± 0.036	0.078 ± 0.048	4.214
MultiHead-GradNorm	0.402 ± 0.029	0.067 ± 0.024	0.401 ± 0.116	0.307 ± 0.034	0.244 ± 0.234	0.596 ± 0.082	0.209 ± 0.065	5.357
MultipleModels-Vanilla	0.407 ± 0.029	0.035 ± 0.009	0.486 ± 0.029	0.557 ± 0.102	**0.011 ± 0.005**	0.522 ± 0.010	0.162 ± 0.020	4.143
MultipleModels-COM	0.459 ± 0.016	0.162 ± 0.023	0.527 ± 0.017	0.285 ± 0.036	0.145 ± 0.027	0.684 ± 0.008	0.498 ± 0.021	8.571
MultipleModels-IOM	0.436 ± 0.021	0.078 ± 0.020	0.424 ± 0.009	0.278 ± 0.034	0.090 ± 0.022	0.606 ± 0.012	0.305 ± 0.053	5.714
MultipleModels-ICT	0.450 ± 0.019	0.279 ± 0.012	0.493 ± 0.027	0.614 ± 0.036	0.174 ± 0.019	0.692 ± 0.023	0.426 ± 0.068	9.286
MultipleModels-RoMA	0.446 ± 0.009	0.303 ± 0.040	0.561 ± 0.023	0.707 ± 0.050	0.273 ± 0.042	0.724 ± 0.008	0.397 ± 0.030	10.857
MultipleModels-Trimentoring	**0.250 ± 0.095**	0.275 ± 0.058	**0.175 ± 0.008**	0.594 ± 0.040	0.206 ± 0.022	0.711 ± 0.061	0.288 ± 0.136	6.714
ParetoFlow	0.510 ± 0.023	0.130 ± 0.015	0.513 ± 0.023	0.193 ± 0.004	0.150 ± 0.010	0.725 ± 0.040	0.479 ± 0.016	8.429
PGD	0.402 ± 0.011	0.205 ± 0.051	0.523 ± 0.011	0.212 ± 0.022	0.252 ± 0.027	0.667 ± 0.006	0.359 ± 0.028	7.214
ParetoTransport (ours)	0.362 ± 0.015	**0.023 ± 0.002**	0.421 ± 0.006	**0.098 ± 0.016**	0.048 ± 0.006	**0.346 ± 0.009**	**0.058 ± 0.013**	**1.571**

Table 30: IGD results on DTLZ (↓). Bold = best per column; underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7	Avg. Rank
D(best)	0.259	0.059	0.083	0.209	0.042	0.684	0.390	–
MultiHead-Vanilla	0.132 ± 0.038	0.024 ± 0.001	0.215 ± 0.020	0.158 ± 0.056	**0.003 ± 0.000**	0.467 ± 0.016	**0.076 ± 0.040**	5.214
MultiHead-PcGrad	0.114 ± 0.018	0.025 ± 0.002	0.103 ± 0.039	0.158 ± 0.006	**0.003 ± 0.000**	0.448 ± 0.020	0.353 ± 0.084	4.571
MultiHead-GradNorm	0.108 ± 0.027	0.032 ± 0.001	0.150 ± 0.043	0.211 ± 0.042	0.161 ± 0.334	0.494 ± 0.058	0.555 ± 0.272	7.714
MultipleModels-Vanilla	0.101 ± 0.027	0.023 ± 0.001	0.185 ± 0.025	0.407 ± 0.096	**0.003 ± 0.000**	0.453 ± 0.009	0.247 ± 0.127	5.714
MultipleModels-COM	0.125 ± 0.016	0.042 ± 0.003	0.183 ± 0.020	0.318 ± 0.065	0.008 ± 0.001	0.528 ± 0.020	0.225 ± 0.039	7.143
MultipleModels-IOM	0.128 ± 0.023	0.027 ± 0.002	**0.072 ± 0.015**	0.336 ± 0.030	**0.003 ± 0.000**	0.483 ± 0.014	0.158 ± 0.108	5.357
MultipleModels-ICT	0.095 ± 0.015	0.061 ± 0.007	0.185 ± 0.028	0.312 ± 0.009	0.076 ± 0.014	0.571 ± 0.014	0.309 ± 0.024	7.643
MultipleModels-RoMA	0.130 ± 0.032	0.183 ± 0.015	0.215 ± 0.055	0.779 ± 0.048	0.101 ± 0.020	0.616 ± 0.017	0.097 ± 0.007	9.786
MultipleModels-Trimentoring	0.076 ± 0.006	0.196 ± 0.114	0.124 ± 0.004	0.362 ± 0.066	0.160 ± 0.000	0.570 ± 0.038	0.126 ± 0.090	7.286
ParetoFlow	0.087 ± 0.041	0.121 ± 0.009	0.217 ± 0.045	0.208 ± 0.001	0.077 ± 0.009	0.608 ± 0.040	0.406 ± 0.022	8.571
PGD	0.083 ± 0.029	0.120 ± 0.031	0.152 ± 0.023	0.457 ± 0.005	0.123 ± 0.016	0.427 ± 0.043	0.268 ± 0.034	7.000
ParetoTransport (ours)	**0.067 ± 0.017**	**0.018 ± 0.001**	0.118 ± 0.020	**0.150 ± 0.009**	0.004 ± 0.000	**0.283 ± 0.000**	0.093 ± 0.063	**2.000**

Table 31: W_2 results on DTLZ (↓). Bold = best per column; underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7	Avg. Rank
D(best)	0.602	0.113	0.341	0.461	0.199	0.764	0.508	–
MultiHead-Vanilla	0.469 ± 0.021	0.107 ± 0.022	0.531 ± 0.035	0.351 ± 0.051	0.110 ± 0.012	0.558 ± 0.026	0.294 ± 0.038	5.714
MultiHead-PcGrad	0.415 ± 0.032	0.100 ± 0.017	0.435 ± 0.020	0.380 ± 0.048	0.096 ± 0.012	0.528 ± 0.030	0.415 ± 0.083	4.286
MultiHead-GradNorm	0.402 ± 0.029	0.095 ± 0.026	0.402 ± 0.116	0.423 ± 0.038	0.323 ± 0.297	0.607 ± 0.082	0.683 ± 0.097	5.714
MultipleModels-Vanilla	0.408 ± 0.029	0.063 ± 0.013	0.487 ± 0.029	0.659 ± 0.106	**0.037 ± 0.013**	0.526 ± 0.010	0.412 ± 0.053	4.286
MultipleModels-COM	0.459 ± 0.016	0.179 ± 0.023	0.527 ± 0.017	0.528 ± 0.055	0.188 ± 0.026	0.688 ± 0.008	0.526 ± 0.021	7.714
MultipleModels-IOM	0.436 ± 0.021	0.101 ± 0.021	0.424 ± 0.009	0.541 ± 0.025	0.125 ± 0.023	0.609 ± 0.013	0.398 ± 0.091	5.429
MultipleModels-ICT	0.451 ± 0.019	0.312 ± 0.012	0.493 ± 0.027	0.756 ± 0.059	0.255 ± 0.020	0.697 ± 0.024	0.567 ± 0.025	9.286
MultipleModels-RoMA	0.446 ± 0.009	0.342 ± 0.036	0.561 ± 0.023	0.931 ± 0.029	0.282 ± 0.041	0.726 ± 0.009	0.498 ± 0.036	10.000
MultipleModels-Trimentoring	**0.250 ± 0.095**	0.419 ± 0.119	**0.175 ± 0.008**	0.693 ± 0.045	0.319 ± 0.028	0.714 ± 0.061	0.349 ± 0.123	6.714
ParetoFlow	0.510 ± 0.023	0.200 ± 0.020	0.513 ± 0.023	0.463 ± 0.003	0.196 ± 0.018	0.732 ± 0.040	0.543 ± 0.012	8.714
PGD	0.403 ± 0.011	0.255 ± 0.054	0.523 ± 0.011	0.566 ± 0.016	0.405 ± 0.048	0.689 ± 0.005	0.544 ± 0.037	8.571
ParetoTransport (ours)	0.363 ± 0.015	**0.039 ± 0.004**	0.421 ± 0.006	**0.323 ± 0.023**	0.066 ± 0.011	**0.352 ± 0.009**	**0.205 ± 0.096**	**1.571**

Table 32: HV results on RE (↑). Bold = best per column; underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33	RE34	RE35	RE36	RE37	Avg. Rank
D(best)	4.176	4.806	4.751	4.595	4.794	10.607	10.611	10.533	9.652	10.217	7.760	5.916	–
MultiHead-Vanilla	**4.598 ± 0.000**	**4.840 ± 0.000**	**4.840 ± 0.000**	4.075 ± 0.815	4.540 ± 0.287	10.647 ± 0.001	10.632 ± 0.009	10.619 ± 0.007	10.109 ± 0.004	10.556 ± 0.024	10.201 ± 0.074	**6.740 ± 0.003**	4.833
MultiHead-PcGrad	**4.598 ± 0.001**	4.828 ± 0.023	4.763 ± 0.170	4.544 ± 0.231	4.739 ± 0.206	**10.648 ± 0.000**	10.628 ± 0.002	10.588 ± 0.025	10.101 ± 0.007	10.498 ± 0.085	10.189 ± 0.070	6.650 ± 0.075	7.000
MultiHead-GradNorm	4.120 ± 0.606	4.721 ± 0.218	2.714 ± 0.067	4.333 ± 0.094	4.689 ± 0.295	10.228 ± 0.577	10.505 ± 0.190	6.970 ± 0.349	8.408 ± 2.057	8.225 ± 3.414	8.496 ± 1.387	5.898 ± 1.595	11.500
MultipleModels-Vanilla	**4.598 ± 0.001**	**4.840 ± 0.000**	4.831 ± 0.018	4.831 ± 0.010	4.838 ± 0.002	**10.648 ± 0.000**	10.639 ± 0.002	10.618 ± 0.004	**10.112 ± 0.003**	10.547 ± 0.035	10.230 ± 0.091	6.739 ± 0.006	**3.417**
MultipleModels-COM	4.238 ± 0.012	4.809 ± 0.012	4.817 ± 0.033	4.658 ± 0.071	4.803 ± 0.004	10.595 ± 0.005	10.617 ± 0.004	10.611 ± 0.008	9.858 ± 0.079	10.471 ± 0.019	9.540 ± 0.049	6.245 ± 0.071	8.750
MultipleModels-IOM	4.594 ± 0.007	**4.840 ± 0.000**	4.813 ± 0.034	4.789 ± 0.104	**4.841 ± 0.000**	**10.648 ± 0.000**	**10.648 ± 0.000**	10.620 ± 0.001	10.105 ± 0.005	10.569 ± 0.009	10.226 ± 0.138	6.723 ± 0.008	3.708
MultipleModels-ICT	**4.598 ± 0.001**	**4.840 ± 0.000**	4.455 ± 0.071	4.830 ± 0.011	4.834 ± 0.003	**10.648 ± 0.000**	10.639 ± 0.003	10.621 ± 0.004	10.101 ± 0.003	10.490 ± 0.006	**10.312 ± 0.012**	6.732 ± 0.007	4.417
MultipleModels-RoMA	4.566 ± 0.003	4.543 ± 0.664	4.836 ± 0.006	3.954 ± 1.200	4.832 ± 0.010	10.644 ± 0.004	10.635 ± 0.005	10.592 ± 0.021	9.917 ± 0.010	10.533 ± 0.022	9.490 ± 0.442	6.611 ± 0.054	7.667
MultipleModels-Trimentoring	4.595 ± 0.001	**4.840 ± 0.000**	**4.840 ± 0.000**	4.530 ± 0.406	4.807 ± 0.002	**10.648 ± 0.000**	10.631 ± 0.010	**10.626 ± 0.000**	10.093 ± 0.004	10.521 ± 0.020	7.270 ± 2.131	6.718 ± 0.017	5.708
ParetoFlow	4.133 ± 0.093	4.815 ± 0.028	4.800 ± 0.022	4.832 ± 0.005	4.823 ± 0.005	10.528 ± 0.103	10.630 ± 0.012	10.340 ± 0.087	9.693 ± 0.037	10.410 ± 0.044	8.785 ± 0.221	5.595 ± 0.405	9.333
PGD	4.479 ± 0.046	4.836 ± 0.003	4.837 ± 0.003	**4.834 ± 0.001**	4.839 ± 0.000	10.613 ± 0.021	10.646 ± 0.001	10.430 ± 0.114	9.444 ± 0.164	10.248 ± 0.110	9.426 ± 0.223	6.182 ± 0.131	7.083
ParetoTransport (ours)	4.577 ± 0.007	**4.840 ± 0.000**	4.839 ± 0.000	4.833 ± 0.002	4.838 ± 0.001	10.614 ± 0.010	10.632 ± 0.007	**10.626 ± 0.001**	10.091 ± 0.018	**10.584 ± 0.009**	10.186 ± 0.139	6.681 ± 0.018	4.583

Table 33: GD results on RE (↓). Bold = best per column; underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33	RE34	RE35	RE36	RE37	Avg. Rank
D(best)	0.111	0.001	0.034	0.040	0.030	0.017	0.051	0.007	0.065	0.077	0.377	0.093	–
MultiHead-Vanilla	**0.001 ± 0.000**	272.464 ± 557.607	0.219 ± 0.071	0.460 ± 0.364	0.461 ± 0.059	0.054 ± 0.023	0.006 ± 0.010	0.093 ± 0.033	**0.007 ± 0.001**	**0.005 ± 0.003**	0.160 ± 0.031	0.016 ± 0.001	6.375
MultiHead-PcGrad	0.002 ± 0.000	96.838 ± 169.131	0.217 ± 0.125	0.251 ± 0.320	0.255 ± 0.254	0.039 ± 0.028	0.154 ± 0.344	0.084 ± 0.029	0.029 ± 0.010	0.095 ± 0.088	0.177 ± 0.035	**0.013 ± 0.001**	7.250
MultiHead-GradNorm	0.081 ± 0.157	0.919 ± 2.027	0.020 ± 0.020	0.240 ± 0.061	0.211 ± 0.144	0.019 ± 0.018	0.133 ± 0.170	0.335 ± 0.055	0.160 ± 0.082	0.242 ± 0.338	0.198 ± 0.169	0.081 ± 0.146	7.917
MultipleModels-Vanilla	**0.001 ± 0.000**	0.084 ± 0.179	0.041 ± 0.049	0.443 ± 0.066	0.153 ± 0.112	0.040 ± 0.020	0.190 ± 0.184	0.102 ± 0.038	**0.007 ± 0.001**	0.039 ± 0.081	0.143 ± 0.026	0.016 ± 0.002	5.708
MultipleModels-COM	0.062 ± 0.002	6.222 ± 13.724	0.023 ± 0.009	**0.035 ± 0.014**	0.125 ± 0.043	**0.015 ± 0.013**	0.039 ± 0.049	0.012 ± 0.004	0.041 ± 0.005	0.035 ± 0.012	0.162 ± 0.021	0.054 ± 0.003	5.292
MultipleModels-IOM	0.006 ± 0.005	3.791 ± 4.118	0.043 ± 0.031	0.118 ± 0.055	4.658 ± 10.412	0.037 ± 0.019	0.211 ± 0.058	0.053 ± 0.036	0.010 ± 0.002	0.072 ± 0.005	**0.092 ± 0.016**	0.023 ± 0.003	6.500
MultipleModels-ICT	**0.001 ± 0.000**	17.170 ± 15.728	**0.006 ± 0.000**	0.437 ± 0.033	0.284 ± 0.025	0.050 ± 0.006	**0.001 ± 0.000**	0.057 ± 0.042	0.022 ± 0.002	0.015 ± 0.002	0.106 ± 0.009	0.019 ± 0.002	5.292
MultipleModels-RoMA	0.042 ± 0.008	198.883 ± 411.102	0.087 ± 0.187	0.648 ± 0.260	0.260 ± 0.221	**0.015 ± 0.007**	0.355 ± 0.065	0.105 ± 0.033	0.105 ± 0.006	0.026 ± 0.006	0.574 ± 0.052	0.058 ± 0.002	9.250
MultipleModels-Trimentoring	0.003 ± 0.001	0.009 ± 0.021	0.072 ± 0.027	0.367 ± 0.216	**0.029 ± 0.006**	0.071 ± 0.010	0.352 ± 0.073	0.009 ± 0.003	0.013 ± 0.002	0.019 ± 0.009	0.623 ± 0.268	0.017 ± 0.002	6.083
ParetoFlow	0.042 ± 0.014	**0.000 ± 0.000**	0.073 ± 0.006	0.117 ± 0.004	0.046 ± 0.006	0.024 ± 0.023	0.058 ± 0.005	**0.007 ± 0.000**	0.058 ± 0.007	0.065 ± 0.004	0.278 ± 0.022	0.118 ± 0.009	6.042
PGD	0.115 ± 0.003	–[7]	0.050 ± 0.004	0.470 ± 0.060	0.035 ± 0.016	0.193 ± 0.074	0.084 ± 0.015	0.017 ± 0.003	0.101 ± 0.009	0.011 ± 0.009	0.469 ± 0.013	0.119 ± 0.008	8.417
ParetoTransport (ours)	**0.001 ± 0.000**	0.018 ± 0.017	0.007 ± 0.006	0.076 ± 0.018	0.030 ± 0.013	0.137 ± 0.062	0.123 ± 0.071	0.023 ± 0.008	0.008 ± 0.001	0.007 ± 0.010	0.138 ± 0.036	0.018 ± 0.002	**3.875**

Table 34: IGD results on RE (↓). Bold = best per column; underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33	RE34	RE35	RE36	RE37	Avg. Rank
D(best)	0.108	0.002	0.046	0.063	0.018	0.402	0.064	0.058	0.070	0.084	0.435	0.121	–
MultiHead-Vanilla	**0.002 ± 0.000**	0.067 ± 0.011	0.011 ± 0.004	0.361 ± 0.331	0.145 ± 0.125	0.237 ± 0.086	0.254 ± 0.123	0.060 ± 0.025	0.023 ± 0.002	0.140 ± 0.100	0.035 ± 0.013	0.040 ± 0.001	5.833
MultiHead-PcGrad	0.003 ± 0.000	0.138 ± 0.084	0.081 ± 0.101	0.127 ± 0.111	0.045 ± 0.093	0.243 ± 0.078	0.312 ± 0.006	0.048 ± 0.010	0.028 ± 0.005	0.138 ± 0.129	0.052 ± 0.026	0.057 ± 0.007	7.333
MultiHead-GradNorm	0.121 ± 0.153	0.150 ± 0.054	0.566 ± 0.030	0.226 ± 0.042	0.072 ± 0.131	0.182 ± 0.038	0.254 ± 0.130	0.509 ± 0.087	0.215 ± 0.206	0.585 ± 0.359	0.237 ± 0.152	0.197 ± 0.176	10.375
MultipleModels-Vanilla	**0.002 ± 0.000**	0.100 ± 0.013	**0.004 ± 0.002**	0.024 ± 0.030	0.001 ± 0.001	0.283 ± 0.091	0.308 ± 0.008	0.094 ± 0.045	0.024 ± 0.001	0.136 ± 0.110	0.042 ± 0.026	0.040 ± 0.002	5.333
MultipleModels-COM	0.056 ± 0.002	0.055 ± 0.043	0.397 ± 0.012	0.038 ± 0.026	0.016 ± 0.002	0.345 ± 0.055	**0.054 ± 0.042**	0.077 ± 0.063	0.044 ± 0.003	0.027 ± 0.002	0.079 ± 0.011	0.059 ± 0.003	7.250
MultipleModels-IOM	0.003 ± 0.000	0.038 ± 0.041	0.187 ± 0.197	0.013 ± 0.026	0.002 ± 0.001	**0.173 ± 0.020**	0.066 ± 0.028	**0.030 ± 0.003**	0.024 ± 0.002	**0.020 ± 0.003**	0.022 ± 0.008	**0.039 ± 0.001**	**3.375**
MultipleModels-ICT	**0.002 ± 0.000**	0.115 ± 0.054	0.042 ± 0.010	0.020 ± 0.034	0.009 ± 0.006	0.218 ± 0.070	0.305 ± 0.002	0.145 ± 0.096	0.031 ± 0.003	0.152 ± 0.010	**0.020 ± 0.004**	0.040 ± 0.002	5.875
MultipleModels-RoMA	0.022 ± 0.002	0.193 ± 0.300	0.092 ± 0.173	0.391 ± 0.512	0.004 ± 0.004	0.325 ± 0.047	0.055 ± 0.009	0.038 ± 0.008	0.086 ± 0.003	0.029 ± 0.004	0.199 ± 0.067	0.052 ± 0.002	7.542
MultipleModels-Trimentoring	0.003 ± 0.001	0.147 ± 0.046	0.007 ± 0.002	0.144 ± 0.176	0.015 ± 0.001	0.276 ± 0.049	0.121 ± 0.022	0.172 ± 0.019	0.052 ± 0.014	0.028 ± 0.006	0.469 ± 0.340	0.041 ± 0.002	7.083
ParetoFlow	0.115 ± 0.024	0.010 ± 0.009	0.042 ± 0.002	0.007 ± 0.005	0.006 ± 0.002	0.395 ± 0.033	0.062 ± 0.001	0.042 ± 0.007	0.066 ± 0.004	0.062 ± 0.018	0.240 ± 0.022	0.173 ± 0.026	6.917
PGD	0.041 ± 0.005	**0.001 ± 0.000**	0.020 ± 0.002	0.007 ± 0.003	**0.000 ± 0.000**	0.287 ± 0.004	0.072 ± 0.016	0.031 ± 0.008	0.079 ± 0.009	0.268 ± 0.043	0.198 ± 0.032	0.093 ± 0.005	6.000
ParetoTransport (ours)	0.006 ± 0.001	0.018 ± 0.007	0.083 ± 0.082	**0.006 ± 0.003**	**0.000 ± 0.000**	0.325 ± 0.009	0.133 ± 0.091	0.060 ± 0.014	**0.022 ± 0.002**	0.052 ± 0.048	0.042 ± 0.030	0.054 ± 0.003	5.083

A.7 EXTENDED RELATED WORK

Forward methods for offline optimization. In offline optimization, a surrogate is trained on the offline dataset and optimized as a proxy for the unknown objective (Xue et al., 2024; Kim et al., 2026). Surrogate-based optimization of expensive black-box objective functions originates with kriging-based expected improvement (Jones et al., 1998); in evolutionary computation, surrogates are used in offline data-driven single-objective optimization (Jin, 2011; Wang et al., 2019) and in

[7] RE22 contains divisions by x_2, whose domain includes $x_2 = 0$. Evaluations at this boundary can therefore produce arbitrarily large or non-finite objective values. This affects GD and W_2, which average distances over the solution set and are therefore dominated by a single divergent point, whereas HV and IGD are unaffected: a divergent point dominates no additional volume and is never the nearest solution to a point on the Pareto front.

Table 35: W_2 results on RE (↓). Bold = best per column; underline = second best. Each algorithm is run for 5 seeds and evaluated on 256 designs.

Method	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33	RE34	RE35	RE36	RE37	Avg. Rank
D(best)	0.120	0.032	0.350	0.063	0.064	0.452	0.296	0.172	0.252	0.191	0.479	0.237	–
MultiHead-Vanilla	0.009 ± 0.004	272.533 ± 557.568	0.323 ± 0.079	0.479 ± 0.366	0.465 ± 0.054	0.348 ± 0.064	0.266 ± 0.103	0.259 ± 0.041	0.148 ± 0.010	0.201 ± 0.101	0.210 ± 0.020	0.108 ± 0.006	5.667
MultiHead-PcGrad	0.014 ± 0.005	96.952 ± 169.051	0.348 ± 0.126	0.267 ± 0.320	0.263 ± 0.261	0.386 ± 0.047	0.445 ± 0.293	0.237 ± 0.020	0.174 ± 0.014	0.325 ± 0.164	0.222 ± 0.030	0.119 ± 0.014	7.000
MultiHead-GradNorm	0.226 ± 0.172	1.045 ± 1.962	0.571 ± 0.028	0.258 ± 0.060	0.236 ± 0.172	0.401 ± 0.064	0.359 ± 0.162	0.656 ± 0.175	0.332 ± 0.119	0.690 ± 0.388	0.444 ± 0.071	0.275 ± 0.161	9.042
MultipleModels-Vanilla	**0.007 ± 0.002**	0.233 ± 0.159	0.212 ± 0.064	0.455 ± 0.067	0.246 ± 0.150	0.401 ± 0.051	0.427 ± 0.114	0.293 ± 0.043	0.151 ± 0.011	0.244 ± 0.127	0.217 ± 0.023	**0.102 ± 0.001**	5.500
MultipleModels-COM	0.069 ± 0.003	6.319 ± 13.674	0.404 ± 0.004	**0.055 ± 0.018**	0.224 ± 0.070	0.440 ± 0.017	**0.174 ± 0.074**	0.197 ± 0.062	0.219 ± 0.012	**0.115 ± 0.011**	0.229 ± 0.017	0.163 ± 0.007	5.917
MultipleModels-IOM	0.015 ± 0.007	3.876 ± 4.104	0.292 ± 0.160	0.134 ± 0.052	4.673 ± 10.441	**0.311 ± 0.074**	0.402 ± 0.056	0.172 ± 0.034	0.165 ± 0.016	0.209 ± 0.010	**0.142 ± 0.022**	0.124 ± 0.003	5.000
MultipleModels-ICT	**0.007 ± 0.001**	17.261 ± 15.678	0.337 ± 0.017	0.451 ± 0.032	0.396 ± 0.030	0.371 ± 0.059	0.314 ± 0.000	0.267 ± 0.106	**0.146 ± 0.012**	0.266 ± 0.044	0.179 ± 0.009	0.107 ± 0.004	5.750
MultipleModels-RoMA	0.064 ± 0.013	198.919 ± 411.081	0.309 ± 0.169	0.667 ± 0.262	0.287 ± 0.213	0.424 ± 0.018	0.456 ± 0.068	0.258 ± 0.051	0.305 ± 0.007	0.194 ± 0.059	0.670 ± 0.061	0.191 ± 0.007	8.583
MultipleModels-Trimentoring	**0.007 ± 0.001**	0.190 ± 0.012	**0.164 ± 0.021**	0.379 ± 0.218	0.112 ± 0.003	0.447 ± 0.018	0.512 ± 0.127	0.304 ± 0.009	0.190 ± 0.040	0.149 ± 0.020	0.729 ± 0.333	**0.102 ± 0.010**	6.042
ParetoFlow	0.187 ± 0.004	**0.034 ± 0.015**	0.360 ± 0.002	0.137 ± 0.004	0.076 ± 0.006	0.433 ± 0.006	0.313 ± 0.006	**0.129 ± 0.004**	0.191 ± 0.035	0.233 ± 0.009	0.413 ± 0.031	0.327 ± 0.040	6.500
PGD	0.147 ± 0.034	–[7]	0.228 ± 0.067	0.483 ± 0.061	0.046 ± 0.014	0.466 ± 0.034	0.271 ± 0.009	0.226 ± 0.037	0.273 ± 0.029	0.460 ± 0.044	0.533 ± 0.012	0.248 ± 0.016	8.208
ParetoTransport (ours)	0.065 ± 0.008	0.139 ± 0.016	0.249 ± 0.095	0.096 ± 0.018	**0.036 ± 0.016**	0.466 ± 0.072	0.298 ± 0.018	0.224 ± 0.050	0.152 ± 0.010	0.214 ± 0.035	0.211 ± 0.045	0.138 ± 0.011	**4.792**

multi-objective evolutionary algorithms (Knowles, 2006; Yang et al., 2020). In single-objective settings, the surrogate model is typically optimized by gradient-based methods, with regularization against over-optimization by conservative objective models (Trabucco et al., 2021), invariant representations (Qi et al., 2022), co-teaching (Yuan et al., 2023), robust model adaptation (Yu et al., 2021), or parallel mentoring (Chen et al., 2023). For multi-objective settings, Off-MOO-Bench (Xue et al., 2024) combines surrogate learning from these single-objective methods with evolutionary search, training either one single-head surrogate per objective with one of the methods above or one multi-head surrogate with multi-task learning methods (Yu et al., 2020; Chen et al., 2018), and searching them with NSGA-II (Deb et al., 2002a). Among classical methods, normal-boundary intersection (Das & Dennis, 1998) obtains an evenly distributed set of Pareto-optimal points by solving, from evenly spaced points on the CHIM (Convex Hull of Individual Minima), one subproblem per point that advances toward the front along the CHIM normal. We adopt this normal as the fixed global direction of our directional transport term, where it moves the induced distribution as a whole rather than individual points (Section 5.2, Appendix A.3.3).

Inverse methods for offline optimization. Inverse methods invert the direction of the problem, producing designs from objective information rather than searching for them. In single-objective offline optimization, model inversion networks learn a conditional generator from objective values to designs (Kumar & Levine, 2020); conditioning by adaptive sampling iteratively re-weights a generative model toward the target region (Brookes et al., 2019); the diffusion model DDOM is trained with an objective-weighted loss and sampled with classifier-free guidance toward high-performing designs (Krishnamoorthy et al., 2023); and BONET synthesizes trajectories of offline points that mimic the runs of an online optimizer and trains on them an autoregressive transformer that generates improving trajectories (Mashkaria et al., 2023).

Only recent work applies generative models to the multi-objective setting. In conditional models, objective information enters during training: PCD trains a conditional diffusion model on a Pareto-reweighted dataset and samples it with classifier-free guidance toward prescribed objective targets (Shrestha et al., 2026); SPREAD learns a conditional diffusion model and refines candidates during reverse diffusion using adaptive multi-objective gradient updates and a repulsion kernel to promote diversity (Hotegni & Peitz, 2026). In training-free guidance, the generative model is trained unconditionally on the offline designs and objective information enters only at inference time: ParetoFlow guides a pre-trained flow-matching model with surrogate gradients under scalarization weights (Yuan et al., 2025), and PGD guides a diffusion model with a time-conditioned preference model that predicts Pareto dominance between two designs, with diversity encouraged by labeling non-dominated pairs through their crowding distance (Annadani et al., 2025). In each of these methods, the objective-space distribution of the generated population is determined implicitly, by the conditioning targets, the scalarization weights, the repulsion kernel, or the preference labels, rather than being specified and refined explicitly during generation. ParetoTransport belongs to the training-free family, which separates generative modeling from task-specific optimization and allows the same model to be reused across tasks, and explicitly specifies and refines the objective-space distribution of the generated population.

Training-free guidance of flow and diffusion models. Beyond offline MOO, recent work guides pre-trained generative models toward a task-specific loss at inference time. The approaches differ in how far the loss gradient is propagated through the model. Full-horizon formulations differentiate through the entire sampling process, optimizing the source sample (Ben-Hamu et al., 2024) or the velocity corrections along the path (Liu et al., 2023), or solve the corresponding optimal control problem (Wang et al., 2025). Single-step formulations replace the full-horizon problem by a

sequence of one-step subproblems in which the loss is evaluated at a posterior-mean estimate of the terminal sample. They differ in whether the gradient is propagated through the model once per step (Chung et al., 2023; Webber et al., 2026), or not at all, by updating the terminal estimate directly and mapping it back to the sampling path (Kim et al., 2025a; Li et al., 2026; Pourya et al., 2026). Our sampler follows the latter construction (Appendix A.3.1). All of these methods guide individual trajectories with a per-sample loss; ParetoTransport lifts the construction to a particle system coupled through a loss functional on the induced distribution.