LEARNING WHERE TO STEER: NOISE-SPACE GEOMETRY FOR EFFICIENT OFFLINE MULTI-OBJECTIVE OPTIMIZATION WITH GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline multi-objective optimization (MOO) seeks solutions with better objective trade-offs using only a fixed dataset, without querying the objectives. Diffusion models trained on such data have emerged as a promising approach, but their samples are not inherently better than the data and must be steered toward the Pareto front. Existing methods guide or condition every sampling step. We instead act on the initial noise and leave the sampling process unchanged. Across Off-MOO-Bench, we observe that the objectives, as functions of the noise, are sensitive to only a few directions. We estimate these directions once per task via a Recursive Feature Machine using function values alone, and a small cache serves every trade-off, so each candidate costs one noise displacement and one ODE solve. We prove that this displacement increases the learned scalarized objective in expectation, and that sweeping trade-offs recovers the flow's attainable front up to proxy and steering errors. With additional guidance, for which we introduce novel data-adaptive and Pareto-aware operators, our method attains the best average hypervolume rank among generative methods on 47 tasks, at comparable or lower sampling cost. Steering alone outranks the best prior generative method at a fraction of its sampling cost.

1 INTRODUCTION

In offline multi-objective optimization (MOO), we are usually given a fixed dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ consisting of candidate solutions $X = \{\boldsymbol{x}_i\}_{i=1}^N \subset \mathbb{R}^D$ and their values on m objectives $\boldsymbol{y}_i = \boldsymbol{f}(\boldsymbol{x}_i) \in \mathbb{R}^m$. The goal is to propose new solutions that approximate the Pareto front without querying the true objectives on new data points. Generative modeling has recently emerged as a promising approach to this problem. In particular, a flow or diffusion model trained on $\mathcal{D}$ can generate new solutions that lie within the support of the input data X. However, faithfully modeling the observed data distribution is not sufficient for optimization, because samples drawn from the learned distribution are not inherently biased toward solutions with better objective values than those already present in the dataset. Improving upon the observed Pareto front may require generating out-of-distribution (OOD) solutions beyond the regions covered by the input data X. Thus, the generative process must be explicitly steered toward high-quality regions of the solution space and, in particular, toward the Pareto front.

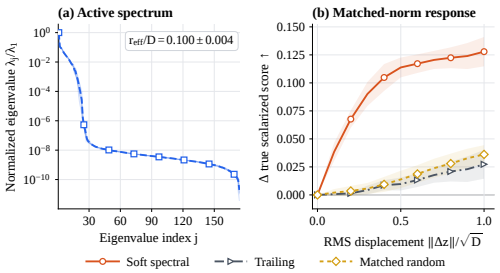

Figure 1: The noise-space objective is low rank. (a) Spectrum of $\boldsymbol{M_w}$ on IN-1K/MOP8. A handful of directions carry nearly all the sensitivity. (b) Displacing along them raises the true score. A trailing or random direction of equal norm does not.

Existing approaches such as ParetoFlow (Yuan et al., 2025) and PGD-MOO (Annadani et al., 2025) achieve this by guiding the sampling trajectory of a flow or diffusion model using proxy-objective gradients or a learned dominance classifier. Such methods require an additional correction or guidance term applied at each integration step during the inference trajectory, increasing inference cost.

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.

Moreover, guidance is often constructed from a denoised estimate of the final sample, and this estimate can be unreliable during the early, highly noisy portion of the trajectory. An alternative approach, PCD (Shrestha et al., 2026), which achieves best prior generative performance, directly trains an objective-conditioned diffusion model and uses classifier-free guidance (CFG) (Ho & Salimans, 2021) at inference time to bias generation toward desirable objective values. This approach avoids the need for an external proxy model, but places a greater burden on the conditional generative model, which must learn a sufficiently accurate relationship between objective values and designs, including in the high-performing regions that are typically sparsely represented in the offline dataset. PCD also reports diminishing returns as the conditioning strength is increased.

Our work. In this paper, we propose *Active-Subspace Noise Steering* (ASNS), a novel and lightweight *noise-space steering* strategy for any ODE-sampled diffusion model. To illustrate the key idea, consider a flow matching model. Recall that the design space of the MOO problem is $\mathbb{R}^D$ and the m objectives are $\boldsymbol{f} : \mathbb{R}^D \to \mathbb{R}^m$. At inference time, sampling begins from a noise vector $\boldsymbol{z} \in \mathbb{R}^D$ and follows the ODE trajectory of the learned flow to produce a clean sample $\boldsymbol{x} \in \mathbb{R}^D$, a new candidate solution. Since the flow dynamics are deterministic, the entire sampling procedure induces a deterministic map $\Phi_\theta : \mathbb{R}^D \to \mathbb{R}^D$, where θ denotes the parameters of the learned flow model. Thus, properties of the final solution are already encoded through its dependence on the initial noise. This motivates our central question: *Can we steer the initial noise efficiently and effectively toward regions that generate solutions near the Pareto front of the MOO problem?*

Interestingly, we find that such noise-space steering can be highly effective. Consider a scalarization of the multi-objective function, $h_{\boldsymbol{w}}(\boldsymbol{x}) := \boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{x})$ (in practice with learned proxies $\hat{\boldsymbol{f}}$), parameterized by a trade-off vector $\boldsymbol{w} \in \Delta^{m-1}$. Through the learned generative map Φ_θ, this induces a scalar-valued function directly on noise space, $H_{\boldsymbol{w}}(\boldsymbol{z}) := h_{\boldsymbol{w}}(\Phi_\theta(\boldsymbol{z}))$. Our key empirical observation is that $H_{\boldsymbol{w}}$ is often sensitive to only a few directions in the high-dimensional noise space $\mathbb{R}^D$. In particular, the average pullback metric $\boldsymbol{M}_{\boldsymbol{w}}$ induced by $H_{\boldsymbol{w}}$ is approximately low-rank across nearly all datasets we consider (see Section 3.1) for details and Figure 1a for an example on IN-1K/MOP8 dataset. This suggests that, although the noise space may be high-dimensional, the objective-relevant variation is concentrated in a low-dimensional subspace. If these important directions can be identified, we can steer the initial noise along them so that the resulting samples move toward the Pareto front.

To identify these objective-relevant directions, we build on the Average Gradient Outer Product (AGOP) and Recursive Feature Machines (RFM) framework of Radhakrishnan et al. (2024), which can extract task-dependent directions effectively using a few examples. RFM+AGOP has previously been used for activation steering in LLMs (Beaglehole et al., 2026b) and for steering the activations of image diffusion models (Wang et al., 2026). Our setting, however, introduces an important additional challenge: rather than steering with respect to a single target function, we must handle an entire family of scalarized objectives $H_{\boldsymbol{w}}$, parameterized by the trade-off vector $\boldsymbol{w}$. In Section 3.2, we show that this family possesses additional structure that allows us to precompute and cache only a small collection of matrices that suffices to support arbitrary trade-offs $\boldsymbol{w} \in \mathbb{R}^m$. At inference time, given a desired trade-off vector $\boldsymbol{w}$, we efficiently construct the corresponding steering directions from these cached matrices and perturb the initial noise along these directions. The resulting "improved" noise vectors are then passed through **the original flow model without further modification**, producing candidate solutions that are biased toward better objective values and closer to the unknown Pareto front. In Section 3.3, we provide theoretical justification for this noise-space steering principle, showing conditions under which moving along the identified objective-relevant directions provably increases the scalarized proxy objective in expectation.

Our noise-space steering strategy is also modular, since it acts only on the initial noise, and can be applied to any pretrained diffusion model, with or without inference-time guidance. In Section 4, we develop complementary guidance operators for flow and diffusion models, which further improve performance when combined with noise steering. We summarize our contributions as follows.

- We introduce ASNS, a novel offline MOO approach that steers only the initial noise, leaving the sampler unmodified. The directions are the leading eigenvectors of the AGOP of the noise-space objective, estimated once per task with an RFM, for every trade-off (Section 3).
- We prove that these directions capture the most squared gradient of any subspace of their dimension, that steering along them increases the learned scalarized objective in expecta-

tion, and that sweeping trade-offs approaches the best front the flow can generate, up to estimation and steering errors (Section 3.3).

- Where a single displacement cannot reach, we introduce two families of guidance operators that use information prior guidance discards and never differentiate through the ODE solve. *Data-adaptive* operators restrict the objective gradient to the learned geometry and the local data manifold, and *Pareto-aware* operators combine the per-objective gradients, not their scalarized sum, into a direction along which no objective decreases to first order (Section 4).
- Across 47 tasks and 14 strong baselines, steering with our guidance operators attains the best average hypervolume rank among generative methods and steering alone outranks the best prior generative method at up to an order of magnitude lower sampling cost. Ablations isolate the chosen directions, rank, weighting, and steering strength (Section 5, § B).

More related work. Surrogate-based methods fit proxies on $\mathcal{D}$ and search them with NSGA-II (Xue et al., 2024; Deb et al., 2002), often regularized against exploitation away from the data (Trabucco et al., 2021; Yu et al., 2021; Qi et al., 2022; Yuan et al., 2023; Chen et al., 2023). Beyond the generative methods above, earlier work models $\mathcal{D}$ for single-objective design (Kumar & Levine, 2020; Krishnamoorthy et al., 2023), SPREAD guides with a common-descent direction (Hotegni & Peitz, 2026), and a recent diagnosis finds generative methods more conservative than evolutionary search when the data lie far from the front (Holly et al., 2026). Diffusion noise has also been optimized per sample and per reward (Wallace et al., 2023; Eyring et al., 2024; Zhou et al., 2025; Ma et al., 2025) or edited along a few patch-PCA components (Wang et al., 2025), whereas we learn objective-sensitive directions once and share them across noise vectors and trade-offs.

2 Preliminaries

Offline multi-objective optimization. We consider the problem

$$\max_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{f}(\boldsymbol{x}) = \big(f_1(\boldsymbol{x}), \dots, f_m(\boldsymbol{x})\big), \tag{1}$$

where $\mathcal{X} \subseteq \mathbb{R}^D$ is a compact design space of candidate solutions and $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^m$ is a vector of $m \geq 2$ objectives, all maximized without loss of generality. No queries to $\boldsymbol{f}$ are permitted. We observe only a fixed dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ with $\boldsymbol{y}_i = \boldsymbol{f}(\boldsymbol{x}_i)$, from which we train differentiable proxies $\hat{\boldsymbol{f}} = (\hat{f}_1, \dots, \hat{f}_m)$. The true $\boldsymbol{f}$ is reserved for final evaluation.

Pareto concepts and scalarization. For $\boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^m$, $\boldsymbol{y}$ *dominates* $\boldsymbol{y}'$ ($\boldsymbol{y} \succ \boldsymbol{y}'$) if $y_a \geq y'_a$ for all a with strict inequality for some a. A solution is *Pareto optimal* if no feasible solution dominates it. The Pareto set $\mathcal{P}^\star$ collects them and the Pareto front is its image $\mathcal{F}^\star = \boldsymbol{f}(\mathcal{P}^\star)$. The goal is a finite set $\mathcal{S}$ whose image is close to $\mathcal{F}^\star$ and spreads across it, measured jointly by hypervolume. We scalarize the objectives by a preference weight $\boldsymbol{w}$ on the simplex Δ^{m-1}. Throughout we use the weighted sum, and sweep a set of weights to cover the front, though our development is agnostic to the choice of scalarization.

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^\top \hat{\boldsymbol{f}}(\boldsymbol{x}) = \sum_{a=1}^{m} w_a \, \hat{f}_a(\boldsymbol{x}), \tag{2}$$

Flow models and guidance. Our method applies to any generative model whose sampler is a deterministic map from Gaussian noise to solutions, which includes flow matching models and diffusion models sampled through their probability flow ODE. Such a model has a velocity field $\boldsymbol{v}_\theta$, and sampling integrates

$$\frac{d\boldsymbol{x}_t}{dt} = \boldsymbol{v}_\theta(\boldsymbol{x}_t, t), \qquad \boldsymbol{x}_0 = \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D), \tag{3}$$

which defines a map $\Phi_\theta : \boldsymbol{z} \mapsto \boldsymbol{x}_1$ from noise to solutions. In our experiments we use a rectified flow (Liu et al., 2023), trained to match $\boldsymbol{x} - \boldsymbol{z}$ along $\boldsymbol{x}_t = (1-t)\boldsymbol{z} + t\boldsymbol{x}$ for $\boldsymbol{x} \sim p_{\text{data}}$. Guidance adds a correction to the velocity at every step,

$$\tilde{\boldsymbol{v}}(\boldsymbol{x}_t, t) = \boldsymbol{v}_\theta(\boldsymbol{x}_t, t) + \eta(t)\, \nabla_{\boldsymbol{x}_t} h_{\boldsymbol{w}}\big(\hat{\boldsymbol{x}}_1(\boldsymbol{x}_t, t)\big), \tag{4}$$

evaluated through the one-step denoised estimate $\hat{\boldsymbol{x}}_1$ because the proxy is trained on clean solutions. For a rectified flow $\hat{\boldsymbol{x}}_1(\boldsymbol{x}_t, t) = \boldsymbol{x}_t + (1-t)\, \boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$, which is accurate near $t = 1$ and unreliable at small t. Prior guided methods for offline MOO follow this scheme (§ 1). Our method acts on the initial noise $\boldsymbol{x}_0 = \boldsymbol{z}$ and integrates Eq. (3) unmodified, and uses Eq. (4) only late in the trajectory.

Algorithm 1 Active-Subspace Noise Steering (ASNS)

Require: A flow map Φ_θ, proxies $\hat{\boldsymbol{f}} = (\hat{f}_1, \dots, \hat{f}_m)$, weight $\boldsymbol{w} \in \Delta^{m-1}$, sample budget n, steering strength γ, rank tolerance $\varepsilon_{\text{rank}}$, *optional* guidance operator $\mathcal{G}_{\boldsymbol{w}}$ with onset t_{start} and strength $\eta(t)$
Ensure: Steered solution $\hat{\boldsymbol{x}} = \Phi_\theta(\boldsymbol{z}_{\boldsymbol{w}})$: an unmodified-flow sample steered toward the $\boldsymbol{w}$-optimum of $H_{\boldsymbol{w}}$

Precomputation *(once, independent of $\boldsymbol{w}$)* ▷ cache the noise-space geometry
  draw $\boldsymbol{z}_1, \dots, \boldsymbol{z}_n \sim \mathcal{N}$; decode $\boldsymbol{x}_i \leftarrow \Phi_\theta(\boldsymbol{z}_i)$; score $\hat{\boldsymbol{y}}_i \leftarrow \hat{\boldsymbol{f}}(\boldsymbol{x}_i)$
  Fit RFM surrogates $\phi_a \approx g_a$ for each objective on $\{(\boldsymbol{z}_i, \hat{y}_{i,a})\}_{i=1}^n$, $a = 1, \dots, m$ ▷ Eq. (7)
  Cache $\boldsymbol{C}_{ab} \leftarrow \frac{1}{n}\sum_i \nabla\phi_a(\boldsymbol{z}_i)\nabla\phi_b(\boldsymbol{z}_i)^\top$, $1 \le a \le b \le m$
Steering *(per weight $\boldsymbol{w}$)*
  $\widehat{\boldsymbol{M}}_{\boldsymbol{w}} \leftarrow \sum_{a,b} w_a w_b \boldsymbol{C}_{ab}$
  Eigendecompose $\widehat{\boldsymbol{M}}_{\boldsymbol{w}} = \sum_j \widehat{\lambda}_j^{(\boldsymbol{w})} \widehat{\boldsymbol{u}}_{\boldsymbol{w}}^{(j)} \widehat{\boldsymbol{u}}_{\boldsymbol{w}}^{(j)\top}$ $\lambda_1^{(\boldsymbol{w})} \ge \cdots \ge \lambda_D^{(\boldsymbol{w})}$ ▷ reuse cached blocks, Eq. (8)
  $\beta_j \leftarrow \mathbf{1}[j \le k_{\text{eff}}(\boldsymbol{w})]$ (hard) or $(\widehat{\lambda}_j^{(\boldsymbol{w})}/\widehat{\lambda}_1^{(\boldsymbol{w})})^\alpha$ (soft) ▷ Eq. (9)
  $\sigma_j \leftarrow \operatorname{sgn}\operatorname{corr}\big(\{\langle \boldsymbol{z}_i, \widehat{\boldsymbol{u}}_{\boldsymbol{w}}^{(j)}\rangle\}_i,\ \{\boldsymbol{w}^\top \hat{\boldsymbol{y}}_i\}_i\big)$ ▷ orientation from cached data, no ODE
  draw $\boldsymbol{z} \sim \mathcal{N}$; $\boldsymbol{z}_{\boldsymbol{w}} \leftarrow \boldsymbol{z} + \gamma \sum_j \beta_j \sigma_j \widehat{\boldsymbol{u}}_{\boldsymbol{w}}^{(j)}$ ▷ weighted step, Thm. 1
Decoding: if $\mathcal{G}_{\boldsymbol{w}}$ is given, replace $\boldsymbol{v}_\theta$ in Eq. (3) by $\boldsymbol{v}_\theta + \eta(t)\,\mathcal{G}_{\boldsymbol{w}}(\boldsymbol{x}_t; \{\boldsymbol{C}_{ab}\})$ for $t \ge t_{\text{start}}$ ▷ Sec. 4
return $\hat{\boldsymbol{x}}_{\boldsymbol{w}} \leftarrow \Phi_\theta(\boldsymbol{z}_{\boldsymbol{w}})$ ▷ one ODE solve, unmodified when $\mathcal{G}_{\boldsymbol{w}}$ is absent

2.1 Problem Setting

Composing the proxies with the sampler turns every objective on solutions into an objective on noise. We write $g_a(\boldsymbol{z}) := \hat{f}_a(\Phi_\theta(\boldsymbol{z}))$ for $a = 1, \dots, m$, and for a preference weight $\boldsymbol{w} \in \Delta^{m-1}$

$$H_{\boldsymbol{w}}(\boldsymbol{z}) := h_{\boldsymbol{w}}\big(\Phi_\theta(\boldsymbol{z})\big) = \sum_{a=1}^{m} w_a\, g_a(\boldsymbol{z}), \tag{5}$$

so that $\nabla H_{\boldsymbol{w}} = \sum_a w_a \nabla g_a$. We reparameterize Eq. (1) and seek, for each $\boldsymbol{w}$, a noise vector $\boldsymbol{z}_{\boldsymbol{w}}$ at which $H_{\boldsymbol{w}}$ is large, decoding the candidate $\hat{\boldsymbol{x}}_{\boldsymbol{w}} = \Phi_\theta(\boldsymbol{z}_{\boldsymbol{w}})$ by Eq. (3). Sweeping $\boldsymbol{w}$ over a finite weight set with several draws per weight, and retaining the non-dominated candidates under $\hat{\boldsymbol{f}}$, yields the solution set $\mathcal{S}$. The reference for $\mathcal{S}$ is set by the sampler. It is the best front attainable among the solutions Φ_θ decodes, and two approximations separate $\boldsymbol{f}(\mathcal{S})$ from it. The proxies $\hat{\boldsymbol{f}}$ defining $H_{\boldsymbol{w}}$ only estimate $\boldsymbol{f}$, and $\boldsymbol{z}_{\boldsymbol{w}}$ is reached by steering rather than by exact maximization. Appendix C.5 defines this reference formally and bounds the distance. How far the reference lies from the true front $\mathcal{F}^\star$ depends on the flow's generalization beyond its data, and cannot be bounded without querying $\boldsymbol{f}$ (Section 3.3).

3 Noise-Space Steering

3.1 The Objective is Low Rank in Noise Space

The noise-space objective $H_{\boldsymbol{w}}$ is accessible only by evaluating the sampler, so the natural approach is to ascend it from a random $\boldsymbol{z}$, but each step requires

$$\nabla_{\boldsymbol{z}} H_{\boldsymbol{w}}(\boldsymbol{z}) = \boldsymbol{J}_{\Phi_\theta}(\boldsymbol{z})^\top \nabla_{\boldsymbol{x}} h_{\boldsymbol{w}}(\boldsymbol{x}), \qquad \boldsymbol{x} = \Phi_\theta(\boldsymbol{z}), \tag{6}$$

where $\boldsymbol{J}_{\Phi_\theta}(\boldsymbol{z}) \in \mathbb{R}^{D\times D}$ is the Jacobian of the sampler at $\boldsymbol{z}$. This gradient differentiates through the ODE solve and must be recomputed for every noise vector and every weight. Prior guided methods avoid this cost by correcting the velocity at each sampling step. Yet Eq. (6) also suggests why exact gradients may be unnecessary. When the data lie near a k-dimensional manifold, the learned score at low noise levels is approximately orthogonal to that manifold (Stanczuk et al., 2024), so we expect the sampler to contract the remaining $D - k$ noise directions. $\boldsymbol{J}_{\Phi_\theta}(\boldsymbol{z})^\top$ then has about k dominant singular directions, and every gradient concentrates in a k-dimensional subspace. If that subspace is shared across $\boldsymbol{z}$, the averaged metric $\boldsymbol{M}_{\boldsymbol{w}} = \mathbb{E}[\nabla H_{\boldsymbol{w}} \nabla H_{\boldsymbol{w}}^\top]$ has effective rank about k (Eq. 9). We find empirical support for this hypothesis. On every benchmark task the estimated spectrum of $\boldsymbol{M}_{\boldsymbol{w}}$ concentrates on a few directions (Figure 3). Within the tasks of one benchmark family such as C-10/MOP1–9 (Section 5), the effective rank changes little as D grows, which suggests that it depends on the intrinsic dimension of the data instead. Differences appear mainly between

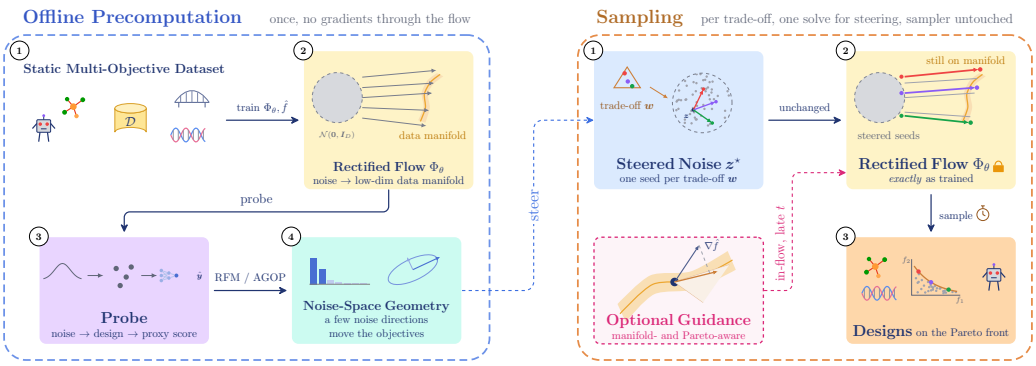

Figure 2: Noise-space RFM steering. *(Left, once per task.)* Random noise is decoded by the flow Φ_θ and scored by the proxies $\hat{\boldsymbol{f}}$, and the RFM recovers from these pairs a metric whose few leading directions are the ones that move the objectives. *(Right, per trade off $\boldsymbol{w}$).* A noise vector is displaced along the cached directions for $\boldsymbol{w}$ and decoded by the unmodified flow in one ODE solve, with optional late guidance (Section 4).

families, and pooling all 47 tasks, which span nearly four orders of magnitude in D, a log-log fit gives $r_{\text{eff}} \propto D^{0.42}$, where r_{eff} is the effective number of directions steering uses (Eq. 13, § B.1). The retained fraction of space therefore shrinks as D grows. The active directions are a property of the sampler and the proxies rather than of any particular $\boldsymbol{z}$, and can be estimated once (Section 3.2). A small rank also matters for steering, since displacing along fewer directions moves the noise less for the same gain (Theorem 1). None of the guarantees below assumes that the spectrum decays. They hold for any rank, with the cost of truncation bounded in Proposition 1.

3.2 Noise-space geometry

Learning the metric. $\boldsymbol{M_w}$ is the average gradient outer product (AGOP) of $H_{\boldsymbol{w}}$ (Radhakrishnan et al., 2024), the second moment of its gradient under the prior, which is prohibitively expensive to form by Eq. (6), so we estimate it from function values. We decode n noise vectors $\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D)$ and score the results with the proxies, obtaining pairs $(\boldsymbol{z}_i, \hat{\boldsymbol{y}}_i)$ with $\hat{\boldsymbol{y}}_i = \hat{\boldsymbol{f}}(\Phi_\theta(\boldsymbol{z}_i))$. For each objective a, following (Radhakrishnan et al., 2024) we fit a so-called Recursive Feature Machine (RFM) to these pairs, a kernel ridge surrogate $\phi_a \approx g_a$ whose Mahalanobis metric $\boldsymbol{M}_a$ estimates the AGOP of g_a and is learned by alternating the fit with the AGOP update,

$$\boldsymbol{M}_a^{(q+1)} \leftarrow \frac{1}{n}\sum_{i=1}^{n} \nabla\phi_a^{(q)}(\boldsymbol{z}_i)\, \nabla\phi_a^{(q)}(\boldsymbol{z}_i)^\top, \qquad \boldsymbol{M}_a^{(0)} = \boldsymbol{I}_D. \tag{7}$$

Kernel machine gradients have a closed form, so the geometry is recovered from input-output pairs alone and Φ_θ is never differentiated. The final surrogate gradients $\nabla\phi_a(\boldsymbol{z}_i)$ are retained. As shown in (Radhakrishnan et al., 2024; Beaglehole et al., 2026a), RFM is well aligned with the AGOP and also particularly effective when there are only a few samples.

One precomputation for all trade off $\boldsymbol{w}$'s. Let $\boldsymbol{G}_i := [\nabla\phi_1(\boldsymbol{z}_i) \cdots \nabla\phi_m(\boldsymbol{z}_i)] \in \mathbb{R}^{D\times m}$ collect the cached gradients at sample i. For any differentiable scalarization $s_{\boldsymbol{w}} : \mathbb{R}^m \to \mathbb{R}$ of the objective values, with $h_{\boldsymbol{w}} = s_{\boldsymbol{w}} \circ \hat{\boldsymbol{f}}$ (the weighted sum Eq. (2) is $s_{\boldsymbol{w}}(\boldsymbol{y}) = \boldsymbol{w}^\top \boldsymbol{y}$), the surrogate $\phi_{\boldsymbol{w}} = s_{\boldsymbol{w}}(\phi_1, \ldots, \phi_m)$ has gradient $\nabla\phi_{\boldsymbol{w}}(\boldsymbol{z}_i) = \boldsymbol{G}_i\boldsymbol{c}_i$ with $\boldsymbol{c}_i := \nabla s_{\boldsymbol{w}}(\phi_1(\boldsymbol{z}_i), \ldots, \phi_m(\boldsymbol{z}_i))$, so the metric of every trade off is assembled from the cache,

$$\widehat{\boldsymbol{M}}_{\boldsymbol{w}} := \frac{1}{n}\sum_i \nabla\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)\nabla\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)^\top = \frac{1}{n}\sum_i \boldsymbol{G}_i\boldsymbol{c}_i\boldsymbol{c}_i^\top\boldsymbol{G}_i^\top = \sum_{a,b=1}^{m} w_a w_b\, \boldsymbol{C}_{ab}, \tag{8}$$

where the last equality holds for the weighted sum Eq. (2), under which $\boldsymbol{c}_i \equiv \boldsymbol{w}$ and $\boldsymbol{C}_{ab} := \frac{1}{n}\sum_i \nabla\phi_a(\boldsymbol{z}_i)\nabla\phi_b(\boldsymbol{z}_i)^\top$, with $\boldsymbol{C}_{aa}$ the converged iterate of Eq. (7). The $m(m+1)/2$ distinct blocks ($\boldsymbol{C}_{ba} = \boldsymbol{C}_{ab}^\top$) are stored once, and a new weight costs one contraction of them and one eigendecomposition. The directions depend on neither the noise vector nor the trajectory, so one precomputation serves every weight, whereas guidance recomputes its signal at every step for every weight.

Active subspace. Compute the eigendecomposition $\boldsymbol{M}_{\boldsymbol{w}} = \sum_j \lambda_j^{(\boldsymbol{w})} \boldsymbol{u}_{\boldsymbol{w}}^{(j)} \boldsymbol{u}_{\boldsymbol{w}}^{(j)\top}$ with $\lambda_1^{(\boldsymbol{w})} \geq \cdots \geq \lambda_D^{(\boldsymbol{w})}$. Each eigenvalue is the mean squared derivative of $H_{\boldsymbol{w}}$ along its eigenvector, so the leading eigenvectors are the noise directions the objective responds to most (Constantine, 2015). We retain the fewest that carry all but an $\varepsilon_{\text{rank}}$ fraction of the total spectrum,

$$k_{\text{eff}}(\boldsymbol{w}) = \min\Big\{r : \textstyle\sum_{j>r} \lambda_j^{(\boldsymbol{w})} < \varepsilon_{\text{rank}} \sum_j \lambda_j^{(\boldsymbol{w})}\Big\}, \tag{9}$$

and the discarded tail is exactly the expected squared gradient outside the retained directions (Proposition 1). We write $\boldsymbol{U}_{\boldsymbol{w}} = [\boldsymbol{u}_{\boldsymbol{w}}^{(1)} \cdots \boldsymbol{u}_{\boldsymbol{w}}^{(r)}]$ with $r = k_{\text{eff}}(\boldsymbol{w})$ for the retained basis, whose span is the active subspace, and the soft weighting of Section 3.3 keeps every direction and weights it by its eigenvalue instead. The same objects computed from $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ carry hats. $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ converges at rate $O(n^{-1/2})$ to the metric of the RFM surrogates, and so do its retained directions under an eigengap, or at $O(n^{-\alpha/2})$ under soft weights. Appendix C.4 bounds the remaining surrogate and proxy errors.

3.3 Noise-space Steering

Given $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D)$ and a weight $\boldsymbol{w}$, we displace the noise within the active subspace to a new noise sample $\boldsymbol{z}_{\boldsymbol{w}}$, and then decode $\boldsymbol{z}_{\boldsymbol{w}}$ through the unmodified ODE to generate a sample $\hat{\boldsymbol{x}}_{\boldsymbol{w}}$:

$$\boldsymbol{z}_{\boldsymbol{w}} = \boldsymbol{z} + \gamma \sum_j \beta_j \, \sigma_j \, \widehat{\boldsymbol{u}}_{\boldsymbol{w}}^{(j)}, \qquad \hat{\boldsymbol{x}}_{\boldsymbol{w}} = \Phi_\theta(\boldsymbol{z}_{\boldsymbol{w}}), \tag{10}$$

with *steering strength* γ and weights $\beta_j = \mathbf{1}[j \leq k_{\text{eff}}(\boldsymbol{w})]$ (hard) or $\beta_j = (\widehat{\lambda}_j^{(\boldsymbol{w})} / \widehat{\lambda}_1^{(\boldsymbol{w})})^\alpha$ (soft). Since the metric carries no orientation, σ_j is the sign of the Pearson correlation between $\langle \boldsymbol{z}_i, \widehat{\boldsymbol{u}}_{\boldsymbol{w}}^{(j)} \rangle$ and $\boldsymbol{w}^\top \hat{\boldsymbol{y}}_i$ over the cache, which by Stein's identity is the population sign of the mean derivative of $H_{\boldsymbol{w}}$ along $\widehat{\boldsymbol{u}}_{\boldsymbol{w}}^{(j)}$, the sign Theorem 1 requires. The following theorem states that, averaged over the noise, this displacement indeed increases the scalarized objective.

Theorem 1 (Steering ascends the scalarized objective)**.** *Let $H_{\boldsymbol{w}}$ be smooth with $\|\nabla^2 H_{\boldsymbol{w}}\|_2 \leq L_{\boldsymbol{w}}$ (Assumption 1). Then,*

$$\mathbb{E}_{\boldsymbol{z}}\big[H_{\boldsymbol{w}}(\boldsymbol{z}_{\boldsymbol{w}}) - H_{\boldsymbol{w}}(\boldsymbol{z})\big] \;\geq\; \gamma \, \langle \boldsymbol{\beta}, |\bar{\boldsymbol{d}}_{\boldsymbol{w}}| \rangle - \tfrac{1}{2} L_{\boldsymbol{w}} \, \gamma^2 \, \|\boldsymbol{\beta}\|_2^2, \tag{11}$$

which is positive for $0 < \gamma < 2\gamma^\star$ and maximal at $\gamma^\star = \langle \boldsymbol{\beta}, |\bar{\boldsymbol{d}}_{\boldsymbol{w}}| \rangle / (L_{\boldsymbol{w}} \|\boldsymbol{\beta}\|_2^2)$, where it equals $\langle \boldsymbol{\beta}, |\bar{\boldsymbol{d}}_{\boldsymbol{w}}| \rangle^2 / (2 L_{\boldsymbol{w}} \|\boldsymbol{\beta}\|_2^2)$. Here $\bar{d}_{\boldsymbol{w},j} := \boldsymbol{u}_{\boldsymbol{w}}^{(j)\top} \mathbb{E}_{\boldsymbol{z}} \nabla H_{\boldsymbol{w}}$, with $\sigma_j = \operatorname{sgn}(\bar{d}_{\boldsymbol{w},j})$ in Eq. (10).

The gain is linear in γ and the curvature penalty quadratic, so a sufficiently small steering strength always ascends (proof in Appendix C.3.2). The subspace enters Eq. (11) through $\bar{\boldsymbol{d}}_{\boldsymbol{w}}$, the mean gradient it retains, and the next result controls what it misses.

Proposition 1 (Optimality of the active subspace)**.** *Among all r dimensional subspaces, $\operatorname{span}(\boldsymbol{U}_{\boldsymbol{w}})$ captures the largest expected squared gradient of $H_{\boldsymbol{w}}$, equal to $\sum_{j \leq r} \lambda_j^{(\boldsymbol{w})}$, and the expected squared gradient it leaves out is exactly $\sum_{j>r} \lambda_j^{(\boldsymbol{w})}$.*

With $\mathbf{P}_{\boldsymbol{w}}^{\perp} := \boldsymbol{I}_D - \boldsymbol{U}_{\boldsymbol{w}} \boldsymbol{U}_{\boldsymbol{w}}^\top$, Jensen's inequality gives $\|\mathbf{P}_{\boldsymbol{w}}^{\perp} \mathbb{E} \nabla H_{\boldsymbol{w}}\|_2^2 \leq \sum_{j>r} \lambda_j^{(\boldsymbol{w})}$ for the mean gradient outside the active subspace, so the rank rule Eq. (9) bounds what Theorem 1 loses to truncation (proof in Appendix C.3.3). Figure 1b is consistent with this. Displacing along $\widehat{\boldsymbol{u}}_{\boldsymbol{w}}^{(1)}$ raises the true objective, while a trailing eigenvector or a random direction of equal norm does not, and the gain saturates beyond a task dependent step.

Steering beyond the support of the data. The displacement has norm $R := \gamma \|\boldsymbol{\beta}\|_2$. Because steering acts only on the noise, R measures how far the steered noise departs from the region the flow was trained on, and we vary it directly (§ B.2.1). As R grows, hypervolume degrades on ZDT4, saturates on the NAS tasks, and keeps increasing on RE36. Offline benchmarks withhold the top solutions from the training set (Xue et al., 2024; Trabucco et al., 2022), so a sampling method needs to reach beyond the data, which generative methods do only conservatively (Holly et al., 2026). Proxies, however, are reliable only near the data (Trabucco et al., 2021; Brookes et al., 2019; Kim et al., 2026), so their predictions far from its support tend to be inaccurate. Theorem 3 (§ C.5) separates the two effects of R. A larger R reaches more of the best front the flow can produce, but the proxies are then evaluated farther from the data, where their error is larger, and which effect dominates depends on the task. Section 4 recovers part of the remaining value without a larger R.

4 GUIDED NOISE-SPACE STEERING

Noise-space-only steering stops improving when the mean gradient leaves the active subspace or when the optimum lies beyond the reach R (Theorem 3). To recover part of this loss, we add guidance late in the same trajectory, replacing the gradient term in Eq. (4) for $t \geq t_{\text{start}}$ with further geometry-adapted projections of the proxy gradient within the RFM+AGOP framework. The RFM geometry is computed once per task, independent of the noise and the trajectory, so each guided step costs one proxy gradient and never differentiates through the ODE solve. Plain guidance follows the scalarized gradient $\boldsymbol{g} = \sum_a w_a \nabla \hat{f}_a$ as is, and we introduce two families of operators that exploit information it discards. *Data-adaptive* operators (RFM-APG, RFM-DAMG) constrain $\boldsymbol{g}$ to the geometry of the objective and of the samples, so a step moves only where the objective responds and the flow can follow. *Pareto-aware* operators (RFM-Cone) replace $\boldsymbol{g}$ by a direction all proxies agree on, so a step does not decrease any objective to first order.

Data adaptive guidance. The geometry of Section 3.2, rebuilt at any t, supplies the first constraint and the candidates sampled in parallel the second. *RFM-APG* repeats the construction once more per task at t_{start}, on intermediate states instead of noise, and projects $\boldsymbol{g}$ onto the resulting active subspace, so the step is restricted to directions in which the remaining flow still moves the objective. *RFM-DAMG* reweights $\boldsymbol{g}$ by the cached spectrum, $\mathbf{R}_{\boldsymbol{w}} = \widehat{\boldsymbol{U}}_{\boldsymbol{w}} \operatorname{diag}(\beta_j) \widehat{\boldsymbol{U}}_{\boldsymbol{w}}^{\top}$, and projects the result onto a tangent space estimated by local PCA over the nearest other candidates at the same t, which come from nearby preferences. When the local rank is not certified by an eigenvalue threshold the projection is skipped and the operator reduces to $\mathbf{R}_{\boldsymbol{w}}\boldsymbol{g}$, so RFM-DAMG exploits a manifold where one exists and reduces to plain guidance where neither a manifold nor a spectral gap exists.

Pareto aware guidance. The per objective proxy gradients are available at every step but scalarized guidance collapses them into one before looking at them, so a step can raise the sum while lowering an objective. *RFM-Cone* uses them jointly. With $\widehat{\boldsymbol{J}}_u$ the matrix of unit normalized proxy gradients at $\hat{\boldsymbol{x}}_1$ and $\boldsymbol{\pi}$ the normalized profile of the current proxy scores, it forms the preference direction $\boldsymbol{u}_{\boldsymbol{w}} = \widehat{\boldsymbol{J}}_u(2\boldsymbol{w} - \boldsymbol{\pi})$, which pulls hardest on the objectives lagging their target weight, and projects it onto the cone $\mathcal{K}(\boldsymbol{x}) = \{\boldsymbol{s} : \langle \nabla \hat{f}_a(\boldsymbol{x}), \boldsymbol{s} \rangle \geq 0 \ \forall a\}$, falling back to the MGDA min norm direction (Désidéri, 2012) where the projection vanishes. The step therefore follows $\boldsymbol{w}$ as far as the objectives permit and never trades one objective for another, with no parameters beyond the onset t_{start} and strength κ, where $\eta(t) = \kappa \|\boldsymbol{v}_\theta\| / \|\mathcal{G}_{\boldsymbol{w}}\|$, shared by all operators. Additional details in Appendix C.6.

Theorem 2 (Per objective improvement)**.** *The RFM-Cone step $\boldsymbol{s}$ satisfies $\langle \nabla \hat{f}_a(\boldsymbol{x}), \boldsymbol{s} \rangle \geq 0$ for every a. If the proxy gradients are L-Lipschitz, then for a small enough step every objective with $\langle \nabla \hat{f}_a(\boldsymbol{x}), \boldsymbol{s} \rangle > 0$ increases and the rest change only at second order. The cone contains a strictly ascending direction if and only if $\boldsymbol{x}$ is not Pareto stationary, and where the projection vanishes at a stationary point the min-norm fallback is zero and the step reduces to the unmodified flow.*

Each guided step improves every proxy to first order, which no scalarized guidance can guarantee (proof in § C.6.1). Steering alone adds no gradient evaluation. Each operator adds one proxy backpropagation per guided step, and RFM-Cone an active-set projection whose cost grows with m, so RFM-DAMG runs below ParetoFlow's cost and RFM-Cone trades cost for the best rank (Table 2).

5 EXPERIMENTS

We evaluate on the 47 tasks of Off-MOO-Bench (Xue et al., 2024) against 14 representative baselines, comprising the 11 DNN-based surrogate methods of Off-MOO-Bench and three generative methods. The tasks span 2 to 10,184 design dimensions, 2 to 8 objectives, and continuous, categorical, and mixed-integer design spaces.

Benchmark tasks. Each task supplies a dataset $\mathcal{D}$ and an oracle $\boldsymbol{f}$ used for evaluation. The 47 tasks form six families: synthetic DTLZ1–7 and ZDT1–4, ZDT6,[1] neural architecture search C-10/MOP1–9 and IN-1K/MOP1–9 (Lu et al., 2024), MO-Hopper and MO-Swimmer from multi-objective RL (Xu et al., 2020), and 15 RE engineering design problems (Tanabe & Ishibuchi, 2020).

[1] Yuan et al. (2025); Shrestha et al. (2026) excluded DTLZ2–6 due to a since-corrected evaluation error (https://github.com/lamda-bbo/offline-moo/issues/14).

Table 1: Average rank (↓) per task family (number of tasks in parentheses), based on 100th-percentile hypervolume (Section 5.1); mean ± std over 5 seeds. Best per column in bold, second best underlined. The last two rows count tasks on which RFM-Cone wins, ties, or loses (W/T/L) against PCD and against ParetoFlow⋆, the better of ParetoFlow and ParetoFlow+ on each task, with ties within 5×10^{-4} in hypervolume.

Methods	DTLZ (7)	ZDT (5)	C-10/MOP (9)	IN-1K/MOP (9)	MORL (2)	RE (15)	All Tasks (47)
$\mathcal{D}$(best)	12.43 ± 0.32	13.00 ± 0.45	10.58 ± 0.51	6.38 ± 0.10	4.10 ± 0.89	16.80 ± 0.27	12.02 ± 0.13
E2E	**3.34 ± 0.61**	5.16 ± 0.62	12.41 ± 0.55	11.38 ± 1.06	17.40 ± 1.95	8.37 ± 1.20	9.01 ± 0.59
E2E + GradNorm	9.71 ± 0.76	13.32 ± 0.59	13.80 ± 1.24	17.60 ± 0.41	7.70 ± 3.78	13.04 ± 0.67	13.37 ± 0.46
E2E + PcGrad	4.97 ± 1.19	7.56 ± 1.82	11.94 ± 0.36	12.42 ± 0.75	14.90 ± 3.45	6.05 ± 0.22	8.78 ± 0.33
MH	6.00 ± 0.96	**4.88 ± 1.23**	12.26 ± 1.68	13.58 ± 0.38	11.20 ± 3.56	6.59 ± 0.30	8.94 ± 0.43
MH + GradNorm	9.91 ± 0.45	11.36 ± 0.55	15.56 ± 1.42	16.47 ± 0.56	8.10 ± 6.18	11.37 ± 0.69	12.79 ± 0.56
MH + PcGrad	8.03 ± 0.52	10.40 ± 0.88	11.43 ± 1.00	12.87 ± 0.50	14.10 ± 1.29	7.17 ± 0.40	9.84 ± 0.32
MM	5.86 ± 0.45	5.60 ± 0.76	10.56 ± 0.54	9.16 ± 0.96	12.30 ± 4.04	8.19 ± 0.76	8.38 ± 0.36
MM + RoMA	13.17 ± 0.54	9.64 ± 1.50	11.66 ± 0.75	10.78 ± 0.92	9.40 ± 1.67	10.48 ± 0.73	11.03 ± 0.26
MM + IOM	9.49 ± 0.67	10.24 ± 0.46	8.38 ± 1.02	6.51 ± 0.91	11.90 ± 4.71	7.47 ± 0.47	8.24 ± 0.45
MM + ICT	10.71 ± 0.80	9.00 ± 0.28	9.70 ± 0.99	8.18 ± 0.94	**3.70 ± 1.25**	6.05 ± 0.81	**8.07 ± 0.54**
MM + Tri-Mentor	11.37 ± 0.67	11.76 ± 0.48	9.13 ± 0.82	11.40 ± 0.58	8.20 ± 3.46	**5.65 ± 0.65**	9.03 ± 0.17
PCD	9.60 ± 0.31	11.08 ± 1.14	7.64 ± 1.22	**5.60 ± 0.80**	8.50 ± 4.23	12.95 ± 0.54	9.64 ± 0.42
ParetoFlow	15.31 ± 0.41	14.12 ± 0.41	10.60 ± 0.83	11.09 ± 0.89	12.70 ± 3.07	15.40 ± 0.37	13.39 ± 0.07
ParetoFlow+	16.20 ± 0.13	12.16 ± 0.67	10.82 ± 0.42	10.11 ± 0.72	14.60 ± 2.68	16.37 ± 0.36	13.56 ± 0.07
RFM-AN (ours)	12.40 ± 0.64	11.56 ± 0.33	5.96 ± 0.66	6.74 ± 1.34	8.10 ± 2.90	9.64 ± 0.49	8.93 ± 0.54
RFM-APG (ours)	10.77 ± 0.75	10.36 ± 1.03	6.03 ± 0.52	6.62 ± 1.25	5.40 ± 2.95	9.39 ± 0.26	8.36 ± 0.30
RFM-DAMG (ours)	10.91 ± 0.51	9.32 ± 1.25	6.13 ± 0.16	6.81 ± 0.73	9.50 ± 2.85	9.38 ± 0.58	8.49 ± 0.34
RFM-Cone (ours)	9.80 ± 0.48	9.48 ± 1.01	**5.41 ± 0.73**	6.31 ± 0.91	8.20 ± 2.51	9.63 ± 0.48	8.13 ± 0.22
RFM-Cone vs. PCD (W/T/L)	4/0/3	3/0/2	5/0/4	4/0/5	1/0/1	10/2/3	27/2/18
RFM-Cone vs. ParetoFlow⋆ (W/T/L)	4/0/3	3/0/2	7/0/2	8/0/1	2/0/0	12/0/3	36/0/11

Categorical NAS designs are optimized as continuous logits, so D denotes the logit dimension. Following Yuan et al. (2025); Shrestha et al. (2026), we exclude the combinatorial tasks (see § A.1).

5.1 Experimental Setup

We compare three families of methods, separated by rules in Table 1. *DNN-based* methods fit surrogates on $\mathcal{D}$ and optimize them with NSGA-II (Deb et al., 2002). The surrogate is End-to-End (E2E), Multi-Head (MH), or Multiple Models (MM). E2E and MH are optionally trained with GradNorm or PcGrad, and MM is optionally regularized per objective with RoMA, IOM, ICT, or Tri-Mentoring (Chen et al., 2018; Yu et al., 2020; 2021; Qi et al., 2022; Yuan et al., 2023; Chen et al., 2023). *Generative* methods are our primary comparison, since they share our modeling assumption and differ only in where the optimization signal enters. ParetoFlow and ParetoFlow+ guide the flow velocity with a weighted proxy gradient (Yuan et al., 2025), where ParetoFlow+ is obtained by replacing ParetoFlow's flow network and proxy predictor with those used in our RFM variants for a fair comparison. PCD (Shrestha et al., 2026) conditions a diffusion model on extrapolated targets under classifier-free guidance (CFG). PGD-MOO is excluded (§ A.6). *Ours* comprises **RFM-AN**, our active-subspace noise steering (ASNS) Eq. (10) followed by one unmodified ODE solve, and three variants that add a guidance operator $\mathcal{G}_{\boldsymbol{w}}$ (Algorithm 1, line 11): the data-adaptive **RFM-APG** (active-subspace projected ascent) and **RFM-DAMG** (manifold-constrained ascent), and the Pareto-aware **RFM-Cone**. None of them differentiates through the ODE solve. All four share a rectified-flow backbone trained on the full dataset and one geometry precomputation per task (Table 4).

Everything other than where the signal enters is held fixed across the proxy-based methods. Surrogates follow the MM configuration of Xue et al. (2024), the weighted-sum scalarization is swept over the same Das–Dennis weights (Das & Dennis, 1998), and generative methods select the final 256 candidates by non-dominated sorting under the proxies (Yuan et al., 2025). PCD needs no proxies, so we follow its published configuration. All baselines are rerun from released code on identical hardware, seeds, and evaluation (additional details in Appendix A.6).

Evaluation. Following Xue et al. (2024), every method returns 256 solutions, scored by hypervolume (HV) against each task's reference point (Table 3), which measures optimality and diversity jointly. We report HV of the full set in the main text and of its non-dominated top half in Appendix D.3. As a reference level, $\mathcal{D}$(best) is the HV of the non-dominated subset of $\mathcal{D}$, which a method must exceed to generalize past its data. Since HV is not commensurable across tasks, we aggregate by average rank (lower is better). On each task and seed we rank all methods and $\mathcal{D}$(best) by HV (1 is best), average the ranks over the tasks of a family, and report mean ± std over 5 seeds. Per-task values are in Appendix D.1.

5.2 RESULTS

Table 1 and 2 support three claims: (i) Noise steering is effective, (ii) additional guidance operator further improves MOO performance, and (iii) our RFM+AGOP approaches are among the cheapest generative methods to sample, and noise steering alone is the fastest of all.

Noise steering improves generative backbones. RFM-Cone attains the best overall rank among generative methods and the second best overall (Table 1), and every RFM variant outranks $\mathcal{D}(\text{best})$ overall. ParetoFlow+, which applies ParetoFlow's per-step guidance to our flow and proxies, isolates the effect of steering, since it differs from RFM-AN only in where the optimization signal enters. RFM-AN improves the overall rank from 13.56 to 8.93, so the gain comes from noise steering rather than the backbone. RFM-AN also outranks PCD, the best prior generative method (8.93 vs. 9.64). With guidance, RFM-Cone outperforms ParetoFlow* in every task family (36 of 47 tasks) and matches or beats PCD at about a third of its sampling cost.

Guidance further improves performance. Each of our guidance operators further improves on RFM-AN in overall rank. The two data-adaptive operators perform similarly, and the Pareto-aware RFM-Cone improves most, with the largest gains on DTLZ and ZDT, where noise-space-only steering (RFM-AN) is weakest. These are synthetic, box-supported families whose fronts lie far outside the data, and surrogate-based search leads there by the widest margin. This is consistent with the regime Theorem 3 isolates, where the best solutions lie beyond one steering step and late correction on the proxies recovers only part of the gap. On the NAS and MORL families, where the data lie on low-dimensional structure, ours outranks every generative baseline or matches it in mean HV.

Sampling / inference cost. RFM-AN is the fastest method on every timed task, roughly four times faster than ParetoFlow and an order of magnitude ($5\times$ to $24\times$) faster than PCD (Table 2). On average, RFM-DAMG samples faster than ParetoFlow with a better HV rank, and RFM-Cone trades further cost for the best HV rank among generative methods, so the family spans the accuracy–cost front, and both data-adaptive variants dominate every generative baseline in rank and mean cost.

Ablations. Appendix B isolates the remaining components. Displacing along the leading eigenvector raises the true objective while a trailing eigenvector or a random direction of equal norm does not, and the gain saturates at a task-dependent step (§ B.2.1 and Figure 1b). Under soft weighting the retained rank is insensitive across a wide range, unlike the hard cutoff (§ B.2.2–B.2.4), and the RFM fit is robust (§ B.2.5). One precomputation serves Tchebycheff scalarizations with the same front quality as the weighted sum (§ B.2.6), and § B.3 studies the guidance strength.

Table 2: Sampling wall-clock time in seconds (mean $\pm$ std over 5 seeds) to produce the final 256 candidates, same hardware for all methods (see § A.4). The first row is the one-off per-task RFM precomputation, cached and shared within seeds and variants and excluded from the sampling times. Best per column in bold.

Method	DTLZ2	ZDT4	C-10/MOP1	MO-Hopper	RE21	RE61	Mean
RFM precomputation (one-off, shared by all RFM variants)	7.6 ± 0.1	7.4 ± 0.0	7.5 ± 0.1	22.7 ± 0.1	7.4 ± 0.1	7.5 ± 0.0	10.0
ParetoFlow	7.4 ± 0.1	8.9 ± 2.0	7.9 ± 0.1	10.8 ± 0.2	8.6 ± 0.3	11.8 ± 0.2	9.2
ParetoFlow+	12.4 ± 0.2	12.2 ± 0.2	11.1 ± 0.2	16.2 ± 0.5	13.5 ± 0.1	19.3 ± 0.1	14.1
PCD	6.8 ± 1.5	7.6 ± 2.3	22.2 ± 5.3	149.4 ± 30.0	22.6 ± 4.9	25.3 ± 4.2	39.0
RFM-AN (ours)	$\mathbf{1.3 \pm 0.0}$	$\mathbf{1.3 \pm 0.1}$	$\mathbf{2.0 \pm 0.0}$	$\mathbf{6.3 \pm 0.2}$	$\mathbf{1.2 \pm 0.4}$	$\mathbf{1.2 \pm 0.0}$	**2.2**
RFM-APG (ours)	3.3 ± 0.2	4.2 ± 0.4	8.2 ± 0.2	31.0 ± 1.3	2.8 ± 0.1	3.7 ± 0.1	8.9
RFM-DAMG (ours)	3.9 ± 1.1	3.4 ± 0.0	4.7 ± 0.1	27.3 ± 0.3	3.3 ± 0.9	4.6 ± 1.3	7.9
RFM-Cone (ours)	6.8 ± 0.2	4.6 ± 0.1	6.0 ± 0.1	32.2 ± 0.9	4.2 ± 0.2	20.0 ± 0.5	12.3

6 CONCLUSION

We introduced noise space steering for offline MOO, which learns the geometry of the objectives in the noise space of a rectified flow with a Recursive Feature Machine and displaces the initial noise along a few cached directions, leaving the sampler as trained. One precomputation serves every trade off, each candidate costs one ODE solve, the step provably ascends the scalarized proxy, and two guidance families extend it where a single displacement cannot reach, attaining the best average rank among generative methods, and steering alone samples fastest of all generative approaches. Our main limitations are that all guarantees are on the proxies, surrogate based search remains stronger on box supported tasks whose fronts lie far outside the data, and the method needs a deterministic sampler and a continuous representation of the design space (Appendix A.7).

AI USE STATEMENT

In this work, we used generative AI to help develop code related to our experiments and analysis, computing aggregate statistics, survey recent related work and support early research ideation. All theoretical claims and proofs were formulated and written by the authors. AI tools were used occasionally to flag possible errors, clarify or find published references supporting a few claims and every suggestion was verified by the authors. We did not use generative AI for creating our benchmarks tasks, formulate our mathematical claims, collect our results or design our method (all our variants) or the core ideas of this work. The bibliography was created manually by the authors from using Semantic Scholar and Open review to find latest published proceedings citation references. The paper was written by the authors, with AI assistance limited to improving phrasing and correcting grammar. We have reviewed all AI-assisted work thoroughly. Every piece of LLM-generated code was verified and tested for correctness by both the leading authors throughout development, all statistics were checked against our logs and every suggested reference was checked against the original paper. We take responsibility for the final content of this work, including any text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only public benchmark tasks (Off-MOO-Bench, including simulated MuJoCo control, NAS benchmarks, and engineering design problems) and involves no human subjects or personal data. We do not foresee direct negative societal impacts beyond those of optimization methods in general. To ensure a fair comparison, all baselines were rerun from released code under a common evaluation protocol, and the corrections we applied to baseline pipelines are documented in Appendix A. All experiments used about 550 GPU-hours on a single GPU (Appendix A.4). We made thorough effort to account for all experimental details and provide comprehensive reporting of aggregate statistics and comparison with previous related works to account for any deviations in reported results in Appendix A.6 and A.6.1.

REPRODUCIBILITY STATEMENT

Algorithm 1 gives the full method. We use the shared hyperparameter configuration in Table 4 and discussed in Appendix A.3. Appendix A.1 describes the tasks, and Table 3 lists the reference points we use to compute hypervolume. We explain how we evaluate and report results in Appendix A.2, and how we reran each baseline, including the fixes we made to their pipelines, in Appendix A.5 and A.6. Every result is averaged over five seeds, and per-task numbers are in Appendix D.1. The proofs and their assumptions are in Appendix C.3.2–C.5. Our anonymized code is included in the supplementary material.

REFERENCES

Tsuyoshi Andô. Comparison of norms $|||f(A) - f(B)|||$ and $|||f(|A - B|)|||$. *Mathematische Zeitschrift*, 197:403–409, 1988. URL https://api.semanticscholar.org/CorpusID:122492348.

Yashas Annadani, Syrine Belakaria, Stefano Ermon, Stefan Bauer, and Barbara E. Engelhardt. Preference-guided diffusion for multi-objective offline optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Daniel Beaglehole, David Holzmüller, Adityanarayanan Radhakrishnan, and Mikhail Belkin. xRFM: Accurate, scalable, and interpretable feature learning models for tabular data. In *The Fourteenth International Conference on Learning Representations*, 2026a.

Daniel Beaglehole, Adityanarayanan Radhakrishnan, Enric Boix-Adserà, and Mikhail Belkin. Toward universal steering and monitoring of ai models. *Science*, 391(6787):787–792, 2026b.

David Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Can Chen, Christopher Beckham, Zixuan Liu, Xue Liu, and Christopher Pal. Parallel-mentoring for offline model-based optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Paul G. Constantine. *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*. Society for Industrial and Applied Mathematics, USA, 2015. ISBN 1611973856.

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, 2002.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5):313–318, 2012. ISSN 1631-073X. doi: https://doi.org/10.1016/j.crma.2012.03.014. URL https://www.sciencedirect.com/science/article/pii/S1631073X12000738.

Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. ReNO: Enhancing one-step text-to-image models through reward-based noise optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Stephanie Holly, Alexandru-Ciprian Zavoianu, 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.

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.

Siddarth Krishnamoorthy, Satvik Mehul Mashkaria, and Aditya Grover. Diffusion models for black-box optimization. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Aviral Kumar and Sergey Levine. Model inversion networks for model-based optimization. In *Advances in Neural Information Processing Systems*, 2020.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhichao Lu, Ran Cheng, Yaochu Jin, Kay Chen Tan, and Kalyanmoy Deb. Neural architecture search as multiobjective optimization benchmarks: Problem formulation and performance assessment. *IEEE Transactions on Evolutionary Computation*, 28(2):323–337, 2024. doi: 10.1109/TEVC.2022.3233364.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Scaling inference time compute for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2523–2534, June 2025.

Han Qi, Yi Su, Aviral Kumar, and Sergey Levine. Data-driven offline decision-making via invariant representation learning. In *Advances in Neural Information Processing Systems*, 2022.

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383(6690):1461–1467, 2024.

Adityanarayanan Radhakrishnan, Mikhail Belkin, and Dmitriy Drusvyatskiy. Linear recursive feature machines provably recover low-rank matrices. *Proceedings of the National Academy of Sciences*, 122(13):e2411325122, 2025. doi: 10.1073/pnas.2411325122. URL https://www.pnas.org/doi/abs/10.1073/pnas.2411325122.

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.

Jan Pawel Stanczuk, Georgios Batzolis, Teo Deveney, and Carola-Bibiane Schönlieb. Diffusion models encode the intrinsic dimension of data manifolds. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

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 *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-bench: Benchmarks for data-driven offline model-based optimization. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Shubhendu Trivedi, Jialei Wang, Samory Kpotufe, and Gregory Shakhnarovich. A consistent estimator of the expected gradient outerproduct. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pp. 819–828, Arlington, Virginia, USA, 2014. AUAI Press. ISBN 9780974903910.

Jean-Philippe Vial. Strong and weak convexity of sets and functions. *Mathematics of Operations Research*, 8(2):231–259, 1983. ISSN 0364765X, 15265471. URL http://www.jstor.org/stable/3689591.

Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Qingsong Wang, Zhengchao Wan, Mikhail Belkin, and Yusu Wang. Seeds of structure: Patch PCA reveals universal compositional cues in diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Qingsong Wang, Mikhail Belkin, and Yusu Wang. General and efficient steering of unconditional diffusion models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Ke Xue, Rongxi Tan, Xiaobin Huang, and Chao Qian. Offline multi-objective optimization. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Sihyun Yu, Sungsoo Ahn, Le Song, and Jinwoo Shin. RoMA: Robust model adaptation for offline model-based optimization. In *Advances in Neural Information Processing Systems*, 2021.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 2020.

Yi Yu, Tengyao Wang, and Richard J. Samworth. A useful variant of the davis–kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015. ISSN 0006-3444. doi: 10.1093/biomet/asv008. URL https://doi.org/10.1093/biomet/asv008.

Ye Yuan, Can Chen, Zixuan Liu, Willie Neiswanger, and Xue Liu. Importance-aware co-teaching for offline model-based optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

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.

Zikai Zhou, Shitong Shao, Lichen Bai, Shufei Zhang, Zhiqiang Xu, Bo Han, and Zeke Xie. Golden noise for diffusion models: A learning framework. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17688–17697, 2025.

APPENDIX

A Additional Experimental details ***15***

A.1 Benchmark Tasks . . . 15

A.2 Evaluation Methodology . . . 15

A.3 Hyperparameter and Tuning used for reporting results . . . 16

A.4 Hardware and Compute . . . 16

A.5 Training Details . . . 17

A.6 Baseline Algorithm Details . . . 17

A.6.1 Validity of the Benchmarks . . . 18

A.7 Limitations . . . 18

B Ablation Studies ***18***

B.1 Effective rank across tasks . . . 18

B.2 Controlled Parameter studies . . . 19

B.2.1 Steering strength and the reach limit . . . 20

B.2.2 Retained rank and hard versus soft weighting . . . 21

B.2.3 Step length versus shape . . . 22

B.2.4 Soft exponent . . . 23

B.2.5 Estimating the noise-space metric . . . 24

B.2.6 Robustness to the choice of scalarization . . . 24

B.2.7 Steering beyond the offline Support on ZDT1 . . . 28

B.3 Guidance Ablations . . . 29

C Theoretical Results ***31***

C.1 Notation and setup . . . 31

C.2 The steering algorithm . . . 32

C.3 First-order guarantee . . . 32

C.3.1 Assumptions . . . 32

C.3.2 Steering ascends the scalarized objective . . . 33

C.3.3 The active subspace is the optimal steering subspace . . . 35

C.4 Estimation and truncation of the metric . . . 35

C.4.1 Truncation error of the active subspace . . . 38

C.5 Pareto-front recovery . . . 38

C.5.1 Pareto-front recovery as distance to an oracle . . . 39

C.6 Guided Noise-Space RFM Steering . . . 42

C.6.1 Preconditioned ascent: adaptive, projected, manifold . . . 42

D Additional Results ***46***

D.1 100th Percentile Results 46
D.2 75th Percentile Results 49
D.3 50th Percentile Results 52

A Additional Experimental details

A.1 Benchmark Tasks

Each Off-MOO-Bench task supplies a fixed dataset $\mathcal{D}$ and an oracle $\boldsymbol{f}$ reserved for evaluation; Table 3 lists the dimension, number of objectives, and dataset size of every task.

Synthetic. DTLZ1–7 (three objectives) and ZDT1–4, ZDT6 (two objectives) are continuous functions with $D \leq 30$ and 60,000 offline designs each, and their fronts are known in closed form. Yuan et al. (2025); Shrestha et al. (2026) excluded DTLZ2–6 due to an evaluation error in the benchmark that has since been corrected, so we include them.

Neural architecture search. C-10/MOP1–9 and IN-1K/MOP1–9 are the NAS benchmarks of Lu et al. (2024) with 2 to 8 objectives. The categorical design space (up to 34 variables) is optimized as continuous logits and decoded by argmax before evaluation, so D is the logit dimension rather than the variable count (Trabucco et al., 2022; Xue et al., 2024).

MORL MO-Hopper and MO-Swimmer are two-objective continuous control problems in which a design is a full policy parameter vector ($D = 10{,}184$ and $9{,}734$). The data are policies collected during PG-MORL training (Xu et al., 2020), and the oracle is a MuJoCo rollout.

Real-world engineering. RE21–25, RE31–37, RE41–42, and RE61 are engineering design problems (Tanabe & Ishibuchi, 2020) with 2 to 7 variables and 2, 3, 4, or 6 objectives. Several have integer variables, and none has a closed-form front.

Following Yuan et al. (2025); Shrestha et al. (2026), we exclude the combinatorial tasks (MO-TSP, MO-CVRP, MO-KP), which would require discrete generative models.

A.2 Evaluation Methodology

Hypervolume. Let $\mathcal{S} \subset \mathcal{X}$ be a solution set and $F = \boldsymbol{f}(\mathcal{S}) \subset \mathbb{R}^m$ its image under the oracle. For a reference point $\boldsymbol{r} \in \mathbb{R}^m$ the hypervolume indicator is the Lebesgue measure of the region dominated by F and bounded by $\boldsymbol{r}$,

$$\mathrm{HV}_{\boldsymbol{r}}(F) = \mathrm{vol}_m\Big(\bigcup_{\boldsymbol{y} \in F} \{\, \boldsymbol{p} \in \mathbb{R}^m \mid \boldsymbol{r} \preceq \boldsymbol{p} \preceq \boldsymbol{y} \,\}\Big), \tag{12}$$

where $\preceq$ is weak dominance under the maximization convention of Section 2, so $\boldsymbol{r}$ lies below every attainable objective vector. Hypervolume is the only indicator that is strictly monotone with respect to Pareto dominance, and it rewards both proximity to $\mathcal{F}^\star$ and spread across it, which is why we report it alone rather than a separate diversity measure. Objectives are min–max normalized using the benchmark's stored per-objective bounds and sign-reversed to match our maximization convention before Eq. (12) is evaluated. We compute hypervolume using the exact `pymoo` implementation for all objective counts.

Percentile protocol. Every method returns exactly $|\mathcal{S}| = 256$ solutions. To report the P-th percentile hypervolume we apply non-dominated sorting to $\boldsymbol{f}(\mathcal{S})$, which partitions $\mathcal{S}$ into fronts $F_1 \succ F_2 \succ \cdots$, order solutions by front index with ties broken by crowding distance, discard the last $(100 - P)\%$ of the ordering, and evaluate Eq. (12) on the remaining $\lfloor 256P/100 \rfloor$ solutions. $P = 100$ therefore scores the full returned set and $P = 50$ scores its better half, which is less sensitive to a small number of strong outliers. We report $P = 100$ in the main text and $P = 50$ in Appendix D.3.

Dataset baseline. $\mathcal{D}(\text{best})$ is the hypervolume of the non-dominated subset of the offline dataset, computed with the same $\boldsymbol{r}$ after selecting its top 256 solutions by non-dominated sorting, and re-

ported at the 100th percentile only. It has no seed variance and is the level any offline method must exceed to have generalized beyond its data.

Reference and Ideal points. Table 3 lists, for every task, the design variables d,the number of objectives m, the size of the offline dataset $|\mathcal{D}|$, the type of design space, and the reference point $\boldsymbol{r} \in \mathbb{R}^m$ against which hypervolume is computed. Reference points follow Off-MOO-Bench (Xue et al., 2024), using the task-specific nadirs from the benchmark snapshot bundled with our code. Under our maximization convention, the reference point is $\boldsymbol{r} = -2.2\,\mathcal{N}(\mathbf{n})$, where $\mathbf{n}$ is the benchmark nadir and $\mathcal{N}$ is the same min–max normalization applied to the objectives. Changes to task-specific nadirs across benchmark versions can account for differences in reported HV and $\mathcal{D}(\text{best})$ from PCD and ParetoFlow papers (Shrestha et al., 2026; Yuan et al., 2025).

Oracle access and decoding. The oracle $\boldsymbol{f}$ is called once per method, seed, and task, on the final 256 solutions only, and never during training, hyperparameter selection, or sampling. NAS logits are decoded to a discrete architecture by taking the argmax over categories for each variable. For some of the RE tasks, the benchmark oracle applies task-specific rounding or mapping to allowed discrete values during evaluation. So hypervolume is computed on feasible solutions rather than on continuous outputs.

Statistical reporting. Each method is run with 5 seeds per task, covering proxy and flow initialization, the noise draws, and any sampler stochasticity. The per-task tables (Appendix D.1) report mean $\pm$ sample standard deviation of hypervolume over the five seeds; methods within one standard deviation of the best, or within 5×10^{-4} of it, are bolded. For Table 1 we rank all methods and $\mathcal{D}(\text{best})$ on each task separately for each seed, average the ranks over the tasks of a family, and report mean $\pm$ sample standard deviation of that family average over the five seeds, so the $\pm$ there is seed variability of the average rank, not dispersion across tasks. The W/T/L rows compare seed-averaged hypervolume per task, counting a tie when the gap is below 5×10^{-4}; no significance test is applied. Wall-clock times (Table 2) are mean $\pm$ standard deviation over the same five seeds.

A.3 Hyperparameter and Tuning used for reporting results

All reported RFM results use the configuration of Table 4. These defaults were fixed before the sensitivity studies of Appendix B.2 were run and are not tuned per task. Those studies test robustness around the defaults rather than select them. The MORL tasks differ only where their dimension requires it. At $D \approx 10^4$ the RFM precomputation is compute-bound, so we fit it with fewer samples and AGOP iterations and retain a basis of 100 directions, and the flow is trained for more epochs.

Table 4: Hyperparameters for flow and proxy training (left) and noise-space steering and sampling (right), shared by all tasks. MORL values are given in parentheses where they differ.

Flow and Proxy Training			Noise-Space Steering and Sampling	
Parameter	**Flow v_θ**	**Proxy $\hat{f}_a$**	**Parameter**	**Value(s)**
Architecture	Residual MLP	MLP	Kernel	Laplace (Mahalanobis)
Depth	4 residual blocks	2 hidden layers	Bandwidth ℓ	20
Width	512	2048	Ridge λ	10^{-3}
Activation	SiLU	LeakyReLU	RFM samples n	10,000 (5,000)
Normalization	LayerNorm	—	AGOP iterations T	7 (4)
Optimizer	AdamW	Adam	Retained directions	D (100)
Learning Rate	2×10^{-4}	10^{-3}	Steering Strength γ	10
LR Schedule	Cosine Annealing	$\times 0.995$/epoch	Soft exponent α	0.3
Batch Size	128	128	Weights $\boldsymbol{w} \times$ draws	256×2 (Das–Dennis)
Epochs	1,000 (3,000)	$\leq$5,000 (early stopping)	Returned set $\lvert\mathcal{S}\rvert$	256
Gradient Clipping	1.0	—	Euler steps	1,000
Checkpoint	EMA 0.999	best val. MSE	Seeds	0–4
RFM Variants (all use soft steering, $\beta_j = (\widehat{\lambda}_j^{(\boldsymbol{w})}/\widehat{\lambda}_1^{(\boldsymbol{w})})^\alpha$)				
RFM-AN	unguided decoding		RFM-DAMG	$t_{\text{start}} = 0.7$, $\kappa = 0.5$, $\eta \leq 11$, kNN $k = 12$, $\tau_{\text{pca}} = 0.2$
RFM-APG	$t_{\text{start}} = 0.8$, $\eta(t) = \eta_{\max}\frac{t-0.8}{0.2}$, for $t \geq 0.8$, $\eta_{\max} = 11$		RFM-Cone	$t_{\text{start}} = 0.7$, $\kappa = 0.5$, uncapped norm matching; min-norm fallback

A.4 Hardware and Compute

Each experiment runs on a single NVIDIA RTX A6000 GPU (48 GB VRAM), on a server with 64 physical CPU cores and approximately 1 TiB of RAM, using PyTorch 2.0.1 and CUDA 11.8. No

method is distributed across devices. Baselines use their released implementations with the sampling configurations described above.

We compare wall-clock sampling times on the same hardware with identical CPU thread limits and an output budget of 256 solutions. For ParetoFlow and PCD, we follow the sampling configurations and hyperparameters reported in their respective papers (Yuan et al., 2025; Shrestha et al., 2026). We report the mean and standard deviation over five seeds, synchronizing GPU computation at the timing boundaries. These measurements exclude model training and oracle evaluation.

RFM additionally requires an initial geometry precomputation, performed once per task and trained-model seed and reused across preference weights. This takes 7.4–22.7 seconds on the six timed tasks and is reported separately from sampling time in Table 2.

The full set of experiments, including all seeds, baselines, and ablations, took approximately 550 GPU-hours, of which 300 were spent on baselines and 250 on our method.

A.5 Training Details

For each task and seed, we train a rectified-flow model and objective-specific proxy predictors on the offline dataset. All RFM variants share these trained models and differ only in their steering and guidance operators. ParetoFlow+ uses the same flow and proxies while retaining ParetoFlow's sampling scheme, allowing us to separate the contribution of our method from that of the backbone and predictors. For PCD, we follow the training and sampling procedures described in its paper and released implementation, with the boundary handling described below.

We ensure that input normalization is consistent between training and inference and that generated designs are mapped back to the original design space using the corresponding inverse transformation. We correct preprocessing inconsistencies in the baseline pipelines and apply task-specific bound handling where needed. In particular, PCD can generate continuous designs outside the prescribed box constraints, so we clip these designs to the task-specific bounds before evaluation [2]. Discrete designs are decoded using the benchmark's decoding rules.

A.6 Baseline Algorithm Details

All RFM variants and ParetoFlow+ share the same trained rectified-flow backbone and objective proxies. Both our flow network and PCD's denoiser use residual MLPs, but PCD includes an additional input/condition (Shrestha et al., 2026). Table 5 summarizes the architectures. Each of our objective proxy is an independent MLP with two hidden layers of width 2048 and LeakyReLU activations, similar to ParetoFlow (Yuan et al., 2025).

Table 5: Architecture comparison with the released PCD implementation. Main-path layer counts exclude the separate time/noise embedding MLP.

Component	RFM / ParetoFlow+	PCD
Generative model	Rectified flow	EDM diffusion
Backbone	Residual MLP	Residual MLP
Hidden width	512	512
Residual blocks	4	4
Normalization	LayerNorm	LayerNorm
Input projection layers	1	2
Output projection layers	1	1
Backbone activation	SiLU	ReLU
Time/noise embedding MLP	$129 \to 256 \to 512$	$17 \to 128 \to 128$

Our backbone follows the residual MLP design used in PCD, adapted to predict a rectified-flow velocity field. We use four residual blocks of width 512, with differences in the input projection, activation functions, and time embedding summarized in Table 5. We notice that ParetoFlow, PGD-MOO, and PCD (Shrestha et al., 2026; Yuan et al., 2025; Annadani et al., 2025) use different gen-

[2] Shrestha et al. (2026) do not explicitly handle this in their code: https://github.com/jatan12/PCD/

erative model architectures. We follow PCD's residual MLP design as closely as practical, with adaptations for our rectified-flow backbone.

PGD-MOO is **excluded** in our experiments as its released code covers only RE, ZDT, and DTLZ (Annadani et al., 2025), and could not be run under our setup, which has also been pointed out in Shrestha et al. (2026).

A.6.1 Validity of the Benchmarks

Hypervolume depends on the reference point and objective normalization, so results from different benchmark versions are not always directly comparable. We found that several task-specific nadir values had been revised across Off-MOO-Bench versions (Xue et al., 2024). Our experiments use the nadir configuration retained in our implementation from the May 2025–June 2026 development period. Under this configuration, our $\mathcal{D}(\text{best})$ values are generally close to those reported by PGD-MOO (Annadani et al., 2025) (within +/-2% of other baseline papers). We evaluate our methods and baseline reruns using the same task-specific normalization and reference points. These settings help explain differences from previously published values. Our NSGA-II-based baseline reruns also obtain positive hypervolume on some tasks reported as zero in the PCD paper (Shrestha et al., 2026). We report the results of these reruns under our common evaluation protocol.

Additionally, following Xue et al. (2024); Shrestha et al. (2026), we omit GP-based baselines, which are costly at this data scale and underperform DNN surrogates.

A.7 Limitations

Our work has several limitations. Our guarantees hold for the learned proxies, and the recovery bound is relative to the best front the flow can generate, not the true Pareto front. For our proposed approach, the best steering strength is task dependent (Appendix B.2.1) and it lags behind surrogate-based methods on tasks whose fronts lie far outside the data. The method also needs a deterministic sampler and a continuous design representation, so we exclude combinatorial tasks. In future work, we plan to choose the steering strength adaptively per task and to combine noise-space steering with surrogate-based search for fronts far from the data. We also plan to extend the approach to discrete generative models.

B Ablation Studies

B.1 Effective rank across tasks

Figure 3 plots the effective rank of the noise-space metric against the noise dimension D for all 47 tasks. For each task we form $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ by Eq. (8) at the uniform weight $\boldsymbol{w} = \mathbf{1}/m$, with the RFM settings of Table 4, and report the participation ratio of the soft weights of Eq. (10),

$$r_{\text{eff}} = \frac{\left(\sum_j \beta_j\right)^2}{\sum_j \beta_j^2}, \qquad \beta_j = \left(\widehat{\lambda}_j^{(\boldsymbol{w})} / \widehat{\lambda}_1^{(\boldsymbol{w})}\right)^{\alpha}, \quad \alpha = 0.3. \tag{13}$$

We use the participation ratio rather than the cutoff rank k_{eff} of Eq. (9). k_{eff} depends on the choice of $\varepsilon_{\text{rank}}$ and jumps by whole directions, so two tasks with similar spectra can receive different ranks, whereas r_{eff} has no threshold and varies continuously with the spectrum, so it can be compared across tasks. It also equals the number of directions the soft steering step effectively uses. Here D is the dimension of the noise, which is the dimension of the continuous representation the flow is trained on, and not the number of design variables d listed in Table 3.

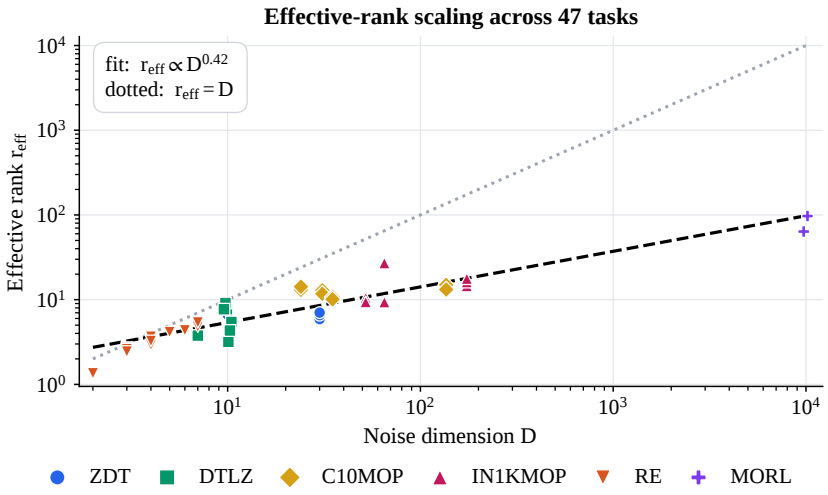

Figure 3: Effective rank Eq. (13) of $\widehat{M}_{\boldsymbol{w}}$ at $\boldsymbol{w} = \mathbf{1}/m$ against noise dimension D for all 47 tasks, log-log axes. The dashed line is a least squares fit, $r_{\text{eff}} \propto D^{0.42}$, and the dotted line is $r_{\text{eff}} = D$.

Across tasks. Pooling the 47 tasks, which span nearly four orders of magnitude in D, a least squares fit of $\log r_{\text{eff}}$ on $\log D$ gives $r_{\text{eff}} \propto D^{0.42}$. The effective rank grows sublinearly, and the gap to $r_{\text{eff}} = D$ widens with dimension. The RE tasks use 1.4 to 5.5 directions at $D \leq 7$, the NAS tasks 9 to 27 at $D \leq 174$, and MORL 65 and 100 at $D \approx 10^4$. The two MORL tasks lie at or just below the fitted line, so the highest-dimensional tasks follow the same trend as the rest, although their RFM retains 100 directions, which caps r_{eff} at 100. The fraction of noise space the objective responds to therefore falls from nearly all of it on the smallest tasks to about one percent on MORL.

Within a family. The rank does not grow with D inside a family. C-10/MOP stays between 10 and 15 as D grows from 24 to 140, IN-1K/MOP at $D = 174$ lies within the range of IN-1K/MOP at $D \approx 50$ to 65, and ZDT lies at 6 to 7 at both $D = 10$ and $D = 30$. This is the pattern predicted in Section 3.1. The rank tracks the intrinsic dimension of the data manifold the flow has learned, which each family fixes independently of D, and the global fit summarizes how the families are arranged along D.

B.2 Controlled Parameter Studies

Noise-space steering has seven hyperparameters. Four belong to the precomputation, the RFM fit that recovers the noise-space geometry. These are the sample budget n, the number of AGOP iterations T, the bandwidth ℓ, and the ridge λ. Three belong to the steering step Eq. (10), namely the retained rank r, the weighting (hard, or soft with exponent α), and the steering strength γ. The guided variants add the guidance onset t_{start} and the guidance strength κ, which we study separately in Appendix B.3. The main results use one configuration for all tasks (Table 4), and the sensitivity studies below instead anchor the steering strength and the rank per task (Table 6).

We vary each of the seven in turn and ask which of them the method is sensitive to, and whether these sensitivities match the theory. The precomputation parameters should matter little. They enter only through the surrogate error, and $\widehat{M}_{\boldsymbol{w}}$ converges to the metric of the surrogates as n grows (Proposition 2). The steering parameters are where the theory makes predictions. The choice of directions matters (Proposition 1), and a displacement helps only up to a task-dependent steering strength (Theorems 1 and 3). Five studies follow below, on the steering strength, the rank and weighting, the soft exponent, the RFM fit, step length versus shape and scalarization.

Common setup. We use one task from each of five families, RE36, ZDT4, C-10/MOP1, IN-1K/MOP8, and MO-Hopper, fixed before any ablation was run, with noise dimensions in Table 6. For every swept value we run RFM-AN (steering only) and RFM-Cone (steering followed by Pareto-aware guidance, Section 4), so each conclusion covers both noise-space-only steering and the full

method. Within a task and seed, all runs share checkpoints, proxies, evaluation noise, and the Das–Dennis weights, and each run generates 512 candidates and keeps 256 by non-dominated sorting on the proxy predictions. We use five seeds per setting and report HV normalized by $\mathrm{HV}(\mathcal{D}(\mathrm{best}))$, so 1.0 is the offline data.

Every parameter not being swept is held at the anchor of Table 4, namely soft weighting with $\alpha = 0.3$, $n = 10{,}000$, $T = 7$, $\ell = 20$, and $\lambda = n\lambda_{\mathrm{stat}} = 10^{-3}$, where the sweeps vary λ_{stat}. The steering strength γ^* and the rank r^* are instead set per task from preliminary sweeps (Table 6), since no single γ suits all tasks. γ^* is the largest γ within one seed standard deviation of the task maximum, and r^* is the rank with the highest mean HV under soft weighting, with ties resolved toward the smaller rank. Each study is also repeated at $r = 4$ and at the full basis $r = r_{\max}$, or at $r = 100$ and 300 on MO-Hopper, so that no conclusion depends on the rank.

Table 6: Per-task anchor settings for the sensitivity studies. D is the noise dimension and $r_{\max}$ the number of available directions. γ^* is the largest steering strength within one seed standard deviation of the task's best HV in preliminary sweeps, and r^* the rank with the highest mean HV under soft weighting, with ties resolved toward the smaller rank. On MO-Hopper both are the main-result defaults. The last column is the seed standard deviation of normalized HV at these settings.

Task	D	$r_{\max}$	r^*	γ^*	seed s.d. at anchor
RE36	4	4	3	20	0.01
ZDT4	10	10	1	2.5	< 0.01
C-10/MOP1	31	31	1	10	0.02–0.03
IN-1K/MOP8	174	174	4	20	0.01
MO-Hopper	10,184	300	100	10	0.03–0.04

Robustness criterion. Overlapping error bars do not show that a parameter is unimportant, so we use a fixed criterion. A parameter is *robust* over its tested range if, for every swept value, the seed-paired difference from the anchor has a 95% confidence interval inside ± 0.02 normalized HV. Otherwise it is *sensitive*, and we report where its good range lies.

B.2.1 Steering strength and the reach limit

A single noise displacement helps where the front lies beyond the data and hurts where it does not, and the steering strength at which it stops helping is task dependent in the way the theory predicts. Theorem 1 guarantees ascent on the proxy for a small enough γ, with a gain linear in γ, and the term $\rho(\boldsymbol{w})$ of Theorem 3, the value the steered noise has not yet collected, shrinks as the displacement grows. HV should therefore rise with γ at first. Beyond some point, two effects take over. The curvature penalty of Theorem 1 grows quadratically, and once the steered noise leaves the region the flow was trained on, the decode becomes less reliable (Remark 4). Tasks should therefore fall into three regimes according to where their front lies relative to that region. A front already inside it gains nothing and degrades once the displacement leaves it, a front reached after a moderate displacement rises and saturates, and a front beyond it keeps rising while the flow extrapolates.

Setup. We sweep $\gamma \in \{0, 1, 2.5, 5, 7.5, 10, 20\}$ with soft weighting at $r = 4$, all other settings at the anchor. On MO-Hopper we sweep $\gamma \in \{0, 10, 20\}$ under two RFM fits, $r = 100$ with $n = 10{,}000$ and $r = 300$ with $n = 40{,}000$. At $\gamma = 0$, RFM-AN reduces to unguided decoding while RFM-Cone keeps its late guidance. Since the same γ moves the noise by different amounts across tasks, Figure 4 also plots HV against the per-coordinate displacement $\|\Delta \boldsymbol{z}\|_2/\sqrt{D}$, the quantity in Remark 4.

Results. All three regimes appear. ZDT4 is in the first. HV is flat up to $\gamma = 2.5$ and falls by about 15% at $\gamma = 20$. The NAS tasks are in the second. IN-1K/MOP8 saturates from $\gamma = 2.5$, and C-10/MOP1 peaks near $\gamma = 5$ and declines only at $\gamma = 20$. RE36 is in the third, still rising at $\gamma = 20$ to $1.22\times$ the offline data. The displacement axis shows that where gains stop is set by the task. IN-1K/MOP8 saturates near 0.5 units of per-coordinate displacement and C-10/MOP1 near 1.4, while RE36 still gains at about 15. MO-Hopper is insensitive to γ within its seed noise under both RFM fits. RFM-Cone matches or exceeds RFM-AN at nearly every steering strength, so guidance never shifts the useful range and γ can be chosen for steering alone.

Recommendation. A steering strength of $\gamma \leq 2.5$ is safe on every task, never losing HV. Larger strengths help where the front lies beyond the data. The default $\gamma = 10$ used in the main results

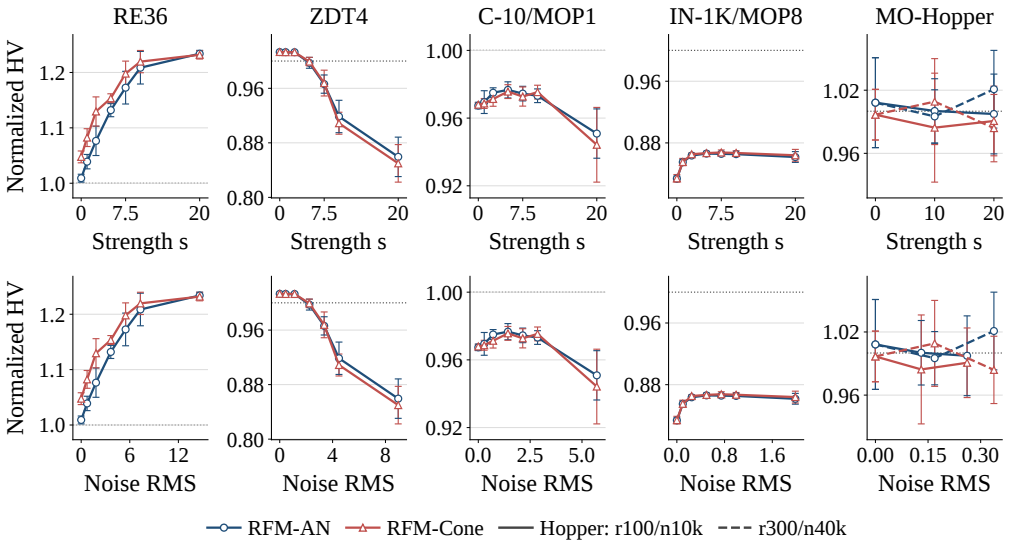

Figure 4: Steering strength sweep. Normalized HV against the steering strength γ (top) and against the per-coordinate displacement $\|\Delta \boldsymbol{z}\|_2/\sqrt{D}$ (bottom), mean $\pm$ s.d. over five seeds. The dotted line is the offline data. On MO-Hopper, solid and dashed lines are the RFM fits with $r = 100$, $n = 10{,}000$ and $r = 300$, $n = 40{,}000$.

lies within 0.02 of each task's best except on ZDT4, and we recommend $\gamma = 2.5$ when no such headroom is expected. The sensitivity studies use a fixed per-task γ^* (Table 6), within 0.016 of the best steering strength of this sweep on every task, and $\gamma^* = 10$ on MO-Hopper, where HV does not depend on γ.

B.2.2 Retained rank and hard versus soft weighting

The retained rank r sets how many directions the step moves along, and the weighting sets how far it moves along each. Indicator weights move the full steering strength along every retained direction. Adding a direction the objective barely responds to therefore lengthens the step without raising the objective, so the step leaves the region where the flow and proxies are reliable and HV falls. Soft weights scale each direction by its eigenvalue. A weak direction receives a small weight, so adding it barely changes the step, and HV should not depend on how many directions are kept. Remark 3 makes this precise. The ascent bound for a weighted step is $\gamma\langle \boldsymbol{\beta}, |\tilde{\boldsymbol{d}}_{\boldsymbol{w}}|\rangle - \frac{1}{2} L_{\boldsymbol{w}} \gamma^2 \|\boldsymbol{\beta}\|_2^2$. With indicator weights $\|\boldsymbol{\beta}\|_2^2 = r$, so each added direction adds a full unit to the curvature penalty while trailing directions add little to the gain. With soft weights $\beta_j = (\lambda_j/\lambda_1)^\alpha$ the penalty is set by the spectrum, so directions with small λ_j change neither term. We therefore expect HV to fall with r past r^* under indicator weights and to stay flat up to the full basis under soft weights.

Setup. We sweep r up to each task's $r_{\max}$, and up to 300 on MO-Hopper with the $r = 300$ fit, at $\gamma = \gamma^*$ and all other settings at the anchor. Three weightings are compared on the same ordered basis. These are soft weights with $\alpha = 0.3$, indicator weights (the hard mode of Eq. (10)), and geometric weights $1, \frac{1}{2}, \frac{1}{4}, \ldots$ within the top r, a control that decays without using the spectrum. At $r = 1$ all three coincide.

Results. One direction is not enough. On RE36 every weighting scores about half the offline data at $r = 1$, and soft weighting reaches $1.23\times$ it at $r = 3$. Past r^* the indicator weights lose HV on every task, by about 23% on IN-1K/MOP8 and 14% on ZDT4. On MO-Hopper they lose 5% to 11% between $r = 4$ and $r = 300$, a drop that stands out from its seed noise for RFM-Cone. Soft weights stay within 0.02 of their value at r^* up to the full basis on every task. The geometric control is also flat, so any decaying weighting removes the rank sensitivity. The spectrum decides how much is gained beyond that. On RE36 soft weighting beats the geometric control by about 0.16, while on the NAS tasks the two lie within 0.03. RFM-Cone reproduces every pattern.

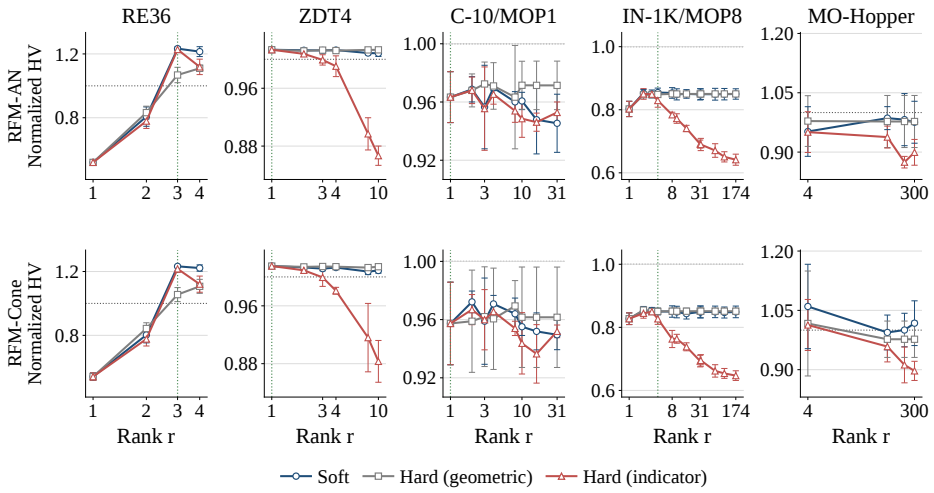

Figure 5: Retained rank sweep for RFM-AN (top) and RFM-Cone (bottom), mean ± s.d. over five seeds. The vertical line marks r^*, the anchor rank of Table 6, and the horizontal dotted line the offline data. Indicator weights lose HV as r grows on every task with room to add directions, while soft and geometric weights stay flat.

Recommendation. Soft weighting makes the retained rank a free choice, which supports the full-basis default of the main results. Indicator weights should be used only with a small, well-chosen rank.

B.2.3 Step length versus shape

Indicator weights, the hard mode of Eq. (10), set $\beta_j = 1$ for the top r directions and $\beta_j = 0$ for the rest, so the step moves equally along every retained direction. They can lose HV at high rank for two reasons. Their step is longer than the soft step at the same γ, and it is spread evenly over all retained directions instead of concentrated on the strong ones. A longer step pays a larger curvature penalty in Theorem 1, and a step spread over weak directions collects less gain. Rescaling the hard step to the length of the soft step removes the first effect, so any gap that remains is due to the weighting itself.

Setup. We rerun the rank sweep of Appendix B.2.2 with both hard modes rescaled at every rank to the soft $\|\Delta \boldsymbol{z}\|_2$, all other settings unchanged. We report the seed-paired difference in normalized HV between soft weighting and each hard mode, with and without this rescaling, as percentages.

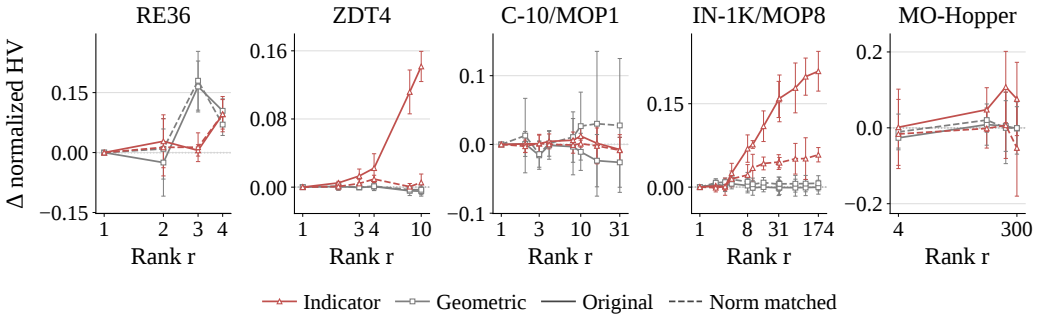

Figure 6: Step length versus shape. Paired difference in normalized HV, soft weighting minus each hard mode, for RFM-AN, mean ± s.d. over five seeds. Positive values favor soft weighting, so a higher curve means that hard mode loses more. Solid lines use the original step, dashed lines rescale the hard step to the length of the soft step.

Results. Figure 6 plots soft minus hard, so a curve above zero means soft weighting wins. Both reasons appear, on different tasks. On ZDT4 the indicator gap of 14% closes to zero once the step

is rescaled, so there the indicator mode only steps too far, and the same holds on MO-Hopper. On IN-1K/MOP8 the gap falls from 21% but stays at 2% to 6% for every $r \geq 8$, and on RE36 the gap of about 10% at $r = 4$ does not change under rescaling, so on both tasks spreading the step evenly costs HV even at equal length. The spectrum also matters beyond decay. On RE36 soft weighting leads the geometric control by about 17% at $r = 3$ with and without rescaling. On C-10/MOP1 the geometric control leads by up to 2.6% with the original step, but soft weighting leads by about 3% once the steps are matched, so the geometric edge comes from step length rather than from a better weighting. RFM-Cone shows the same pattern.

Recommendation. Soft weighting wins for both reasons. It keeps the step short when weak directions are added, and at equal length it still places the weight where the gradient is, which supports its use as the default throughout.

B.2.4 Soft exponent

Following the comparison of hard and soft weighting, we now study the one hyperparameter of soft steering, the soft exponent α in $\beta_j = (\lambda_j/\lambda_1)^\alpha$. The exponent sets how many directions the step uses. As $\alpha \to 0$ every weight tends to one and soft weighting becomes the indicator mode, and as α grows the weight concentrates on the leading direction, which is steering at $r = 1$. Both limits lost HV in Appendix B.2.2, the first on tasks with many weak directions and the second on tasks where several directions carry the gradient, so HV should peak in between. We count the directions a weighting uses by the participation ratio r_{eff} of Eq. (13).

Setup. We sweep $\alpha \in \{0.1, 0.3, 1, 1.5, 3\}$ at $r = r^*$ and at $r = r_{\max}$, with $\gamma = \gamma^*$ and all other settings at the anchor. On MO-Hopper we use the fit with $r = 300$. We report gaps in normalized HV as percentages.

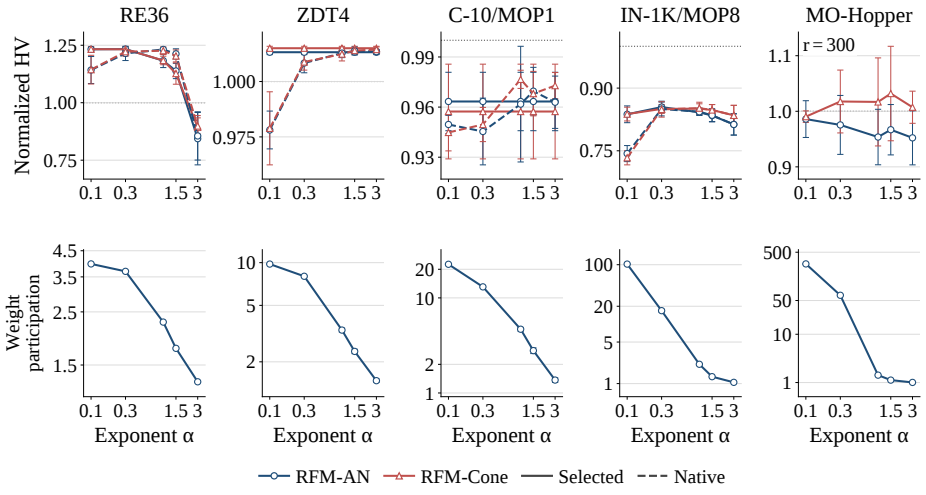

Figure 7: Soft exponent. Top, normalized HV against α at $r = r^*$ (solid) and $r = r_{\max}$ (dashed), mean $\pm$ s.d. over five seeds. Bottom, the number of directions the weighting uses, r_{eff} of Eq. (13), at $r = r_{\max}$. It falls from nearly the full basis at $\alpha = 0.1$ to a single direction at $\alpha = 3$.

Results. Both limits fail where predicted. At $\alpha = 0.1$ the step spreads over nearly the full basis, and HV at $r = r_{\max}$ drops by 12% on IN-1K/MOP8, 4% on ZDT4, and 2% to 3% on C-10/MOP1, the loss of the indicator mode in Appendix B.2.2. At $\alpha = 3$ the step collapses onto one direction, and RE36 loses 34% to 39%, the same collapse $r = 1$ produced. In between, $\alpha = 0.3$ is the best exponent or within 0.1% of it on every task at r^*, and within 3% of the best at $r_{\max}$. At r^*, ZDT4 and C-10/MOP1 are flat because $r^* = 1$ leaves nothing to reweight, and MO-Hopper is flat within its seed noise. RFM-Cone follows RFM-AN throughout. At $\alpha = 0.3$ the step uses only a handful of directions, about 17 of 174 on IN-1K/MOP8 and 64 of 300 on MO-Hopper, consistent with the effective ranks of Appendix B.1.

Recommendation. Any $\alpha \in [0.3, 1]$ is a good choice, with HV within 3% of the best on every task except RE36 at r^*, where $\alpha = 1$ loses 5%. We use $\alpha = 0.3$, the elbow at the flat end of this range. It

spreads the step over the most directions that still keep HV near the best, which protects tasks such as RE36 that need several directions. Below 0.3 the full-basis curves fall toward the loss at $\alpha = 0.1$ as the step spreads over nearly every direction, and toward the sharp end the losses grow to 39%. The elbow therefore gives the best worst case, never more than 3% below the best on any task.

B.2.5 Estimating the noise-space metric

Steering uses the noise-space metric $\boldsymbol{M}_{\boldsymbol{w}}$, the AGOP of $H_{\boldsymbol{w}}$ (Section 3.2), which we cannot compute directly and replace by its estimate $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$. The RFM builds this estimate from n decoded noise samples. For each objective it fits a kernel ridge surrogate whose Laplace kernel uses a Mahalanobis metric $\boldsymbol{M}_a$, and it alternates the fit with the AGOP update Eq. (7) for T iterations. The converged $\boldsymbol{M}_a$ are the diagonal blocks $\boldsymbol{C}_{aa}$ of $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ in Eq. (8).

The sample budget n, the iterations T, the bandwidth ℓ, and the ridge λ therefore set how well $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ is estimated, not how the step is taken. They enter only through the surrogate error, and $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ converges to the metric of the surrogates as n grows (Proposition 2), so HV should not depend on them over a reasonable range.

Setup. We sweep $n \in \{5{,}000, 10{,}000, 20{,}000\}$ and $T \in \{4, 7, 11\}$ over their full grid, and $\ell \in \{10, 20, 40\}$ and $\lambda_{\text{stat}} \in \{10^{-8}, 10^{-7}, 10^{-6}\}$ one at a time at $n = 5{,}000$ and $T = 7$, where the ridge is $\lambda = n\lambda_{\text{stat}}$ and the anchor $\lambda_{\text{stat}} = 10^{-7}$ gives $\lambda = 10^{-3}$ at $n = 10{,}000$. On MO-Hopper, n runs from 10,000 to 40,000, the bandwidth and ridge sweeps are at $n = 10{,}000$, and every sweep is repeated under both RFM fits, $r = 100$ and $r = 300$.

Results. Figure 8 shows that HV does not depend on any of the four parameters over the tested range. Across a $4\times$ range of n, a $3\times$ range of T, a $4\times$ range of ℓ, and a $100\times$ range of λ_{stat}, HV changes by at most 1.6% on RE36, ZDT4, C-10/MOP1, and IN-1K/MOP8, and the largest shift is on RE36 at $\lambda_{\text{stat}} = 10^{-6}$. MO-Hopper is flat within its seed noise under both fits. RFM-Cone follows RFM-AN throughout.

Note, the flat curves mean the directions are easy to estimate, not that they do not matter, since how the step is spread over them changes HV by up to 23% (Appendix B.2.2). Because the objective depends on only a few directions (Appendix B.1), the RFM recovers them reliably under any of these settings. The smallest setting tested, $n = 5{,}000$ with $T = 4$, stays within 1% of the anchor on every task and fits in 0.4 s against 0.95 s at the anchor, which supports the cheaper precomputation used on the MORL tasks.

Recommendation. Noise-space metric estimation is robust. The defaults $n = 10{,}000$, $T = 7$, $\ell = 20$, and $\lambda = 10^{-3}$ lie inside this robust range, and a smaller budget can be used where the precomputation is expensive.

B.2.6 Robustness to the choice of scalarization

The main results use the weighted sum Eq. (2), and the middle expression of Eq. (8) extends the cache to any differentiable scalarization. This section tests that extension. We ask three questions. (i) Does steering ascend the scalarized objective under scalarizations other than the weighted sum, including non-smooth ones? (ii) Does the one-off precomputation serve them without refitting the RFM or re-running the flow, and at what cost in quality? And (iii) is the steering magnitude at which the gains saturate a property of the sampler, as Remark 4 suggests, or of the scalarization?

Scalarizations and setup. All maximize, with $\boldsymbol{w} \in \Delta^{m-1}$ and the ideal point $\boldsymbol{z}^\star$ set per task to the componentwise maximum of the offline objective values. Besides the weighted sum (WS), we test Tchebycheff (Tch), augmented Tchebycheff (aug-Tch, $\zeta = 0.05$), and a smoothed Tchebycheff (softmin-Tch), where ν is fixed per task to 10% of the offline objective range and not tuned, and $h^{\text{soft}}_{\boldsymbol{w}}$ is smooth with $h^{\text{soft}}_{\boldsymbol{w}} \to h^{\text{Tch}}_{\boldsymbol{w}}$ as $\nu \to 0$,

$$
\begin{aligned}
h^{\text{Tch}}_{\boldsymbol{w}}(\boldsymbol{x}) &= \min_a \, w_a\big(\hat{f}_a(\boldsymbol{x}) - z^\star_a\big), \qquad h^{\text{aug}}_{\boldsymbol{w}}(\boldsymbol{x}) = h^{\text{Tch}}_{\boldsymbol{w}}(\boldsymbol{x}) + \zeta\, \boldsymbol{w}^\top \hat{\boldsymbol{f}}(\boldsymbol{x}), \\
h^{\text{soft}}_{\boldsymbol{w}}(\boldsymbol{x}) &= -\nu \log \textstyle\sum_a \exp\big(-w_a(\hat{f}_a(\boldsymbol{x}) - z^\star_a)/\nu\big).
\end{aligned}
\tag{14}
$$

Each scalarization assembles $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ from the cached per-objective gradients by the middle expression of Eq. (8). The precomputation runs once per task and seed and is never repeated for a scalarization,

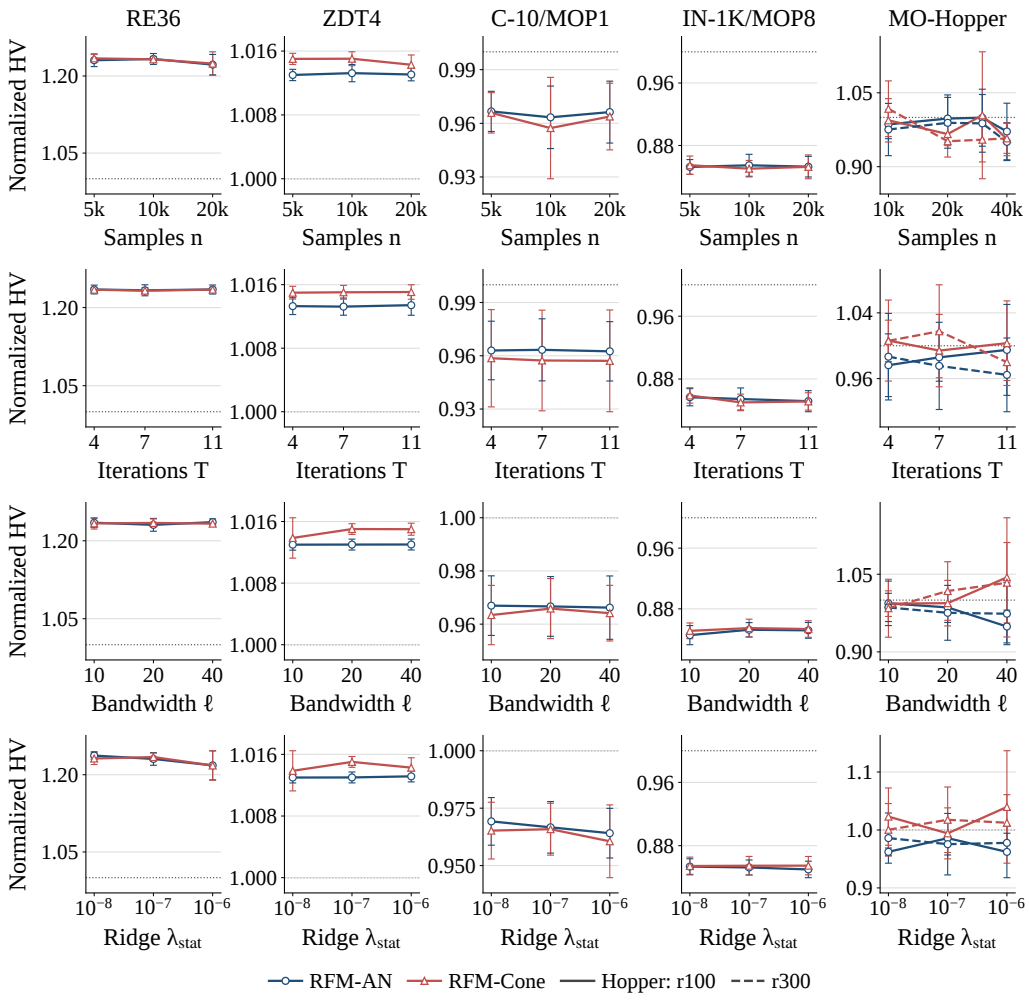

Figure 8: Estimating the noise-space metric. Normalized HV against the sample budget n, the AGOP iterations T, the bandwidth ℓ, and the ridge λ_{stat} (rows), mean $\pm$ s.d. over five seeds. The dotted line is the offline data. On MO-Hopper, solid and dashed lines are the fits with $r = 100$ and $r = 300$.

and no scalarization issues a new flow evaluation at any weight. All three studies use ZDT2, seeds 0–4, the main-experiment RFM configuration in Table 4 for everything not varied, and RFM-AN only, so that the mechanism is isolated from the guidance operators of Section 4, whose gradient combination is itself scalarization-specific. Equivalence is judged by the criterion of Appendix B.2, paired differences against WS at matched weight and seed with a 95% confidence interval inside ± 0.02 normalized HV.

Study 1. Steering ascends under every scalarization. Theorem 1 guarantees ascent for any smooth scalarized objective and never uses linearity in $\boldsymbol{w}$. Ascent should therefore hold under all four scalarizations, with the possible exception of Tch and aug-Tch, which are not smooth. The Tchebycheff value follows whichever objective is currently the worst, so its gradient jumps wherever two objectives tie for that role, and at such points the curvature bound of Assumption 1 that the proof of Theorem 1 relies on does not exist. Whether these kinks matter in practice is exactly what the comparison against the smooth softmin-Tch decides.

Setup. The matched-norm protocol of Figure 1b, run once per scalarization at the same $\boldsymbol{w}$ and the same base noise. The leading arms displace along the top or the softly weighted directions of that scalarization's $\widehat{M}_{\boldsymbol{w}}$, and three controls of equal norm displace along a trailing eigenvector, a random direction, and the sign-flipped leading direction. We read the true scalarized objective of

the scalarization that steered, over $\|\Delta \boldsymbol{z}\|/\sqrt{D} \in [0, 1]$. All four scalarizations take their directions from the same cached precomputation.

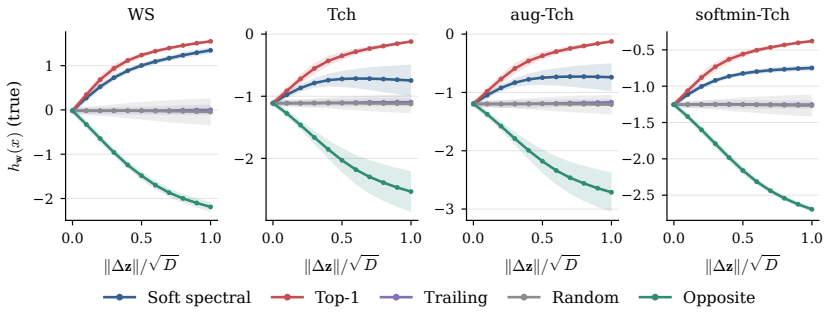

Figure 9: Steering ascends under every scalarization. True scalarized objective against step size on ZDT2 (mean $\pm$ s.d. over five seeds, same $\boldsymbol{w}$ and base noise). In every panel the leading arms climb, the controls stay flat, and the reversed arm descends, and all four reach the same HV. Only the soft arm under the non-smooth Tch and aug-Tch becomes unreliable at large steps.

Results. All runs use the balanced weight $\boldsymbol{w} = (\frac{1}{2}, \frac{1}{2})$, and every scalarization steers with directions taken from the same cached precomputation, with no refit. Steering improves the objective under all four (Figure 9). The top-1 arm climbs steadily, while the trailing and random directions produce no systematic gain and the reversed direction makes the objective worse. The four panels have different vertical scales, because each plots its own scalarized value, so we compare the scalarizations through hypervolume, which is the same metric for all four. Irrespective of the scalarization, steering raises the hypervolume by 3.6% to 4.6% and all four attain the same level, within one percentage point of one another. The precomputation therefore serves every scalarization equally well, and the ascent of Theorem 1 is not specific to the weighted sum.

The one difference concerns smoothness. Theorem 1 guarantees ascent only for a smooth objective (Assumption 1). Tch and aug-Tch violate this assumption, because they take a minimum over the objectives and so have a kink wherever the minimizing objective switches. softmin-Tch replaces the minimum with a smooth approximation and satisfies it. The soft arm shows the consequence. Under Tch and aug-Tch it stops improving once the step size $\|\Delta \boldsymbol{z}\|/\sqrt{D}$ reaches about 0.5, and its results vary much more across runs, with a standard deviation of about 0.24 at the largest step. Under softmin-Tch it keeps climbing with a standard deviation of 0.05. Smoothing the scalarization is therefore enough to restore reliable ascent, as the theorem predicts. The top-1 arm stays consistent under all four scalarizations, including the non-smooth ones.

Study 2: One precomputation serves every scalarization. Study 1 showed that steering works under every scalarization when its directions come from the cached precomputation. This study asks what the cache costs. The cache is fit once, and for any new weight it assembles $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ without refitting and without running the flow. For the weighted sum the assembly is the bilinear combination of Eq. (8). For the Tchebycheff family, whose gradient is not linear in $\boldsymbol{w}$, the same cached per-sample gradients are recombined directly. The most accurate alternative is to refit the RFM from scratch at every weight. We compare the two in quality and in cost.

Setup. ZDT2, five runs, seven weights spread across the simplex. At each weight and for each scalarization we build $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ from the cache and from a fresh refit, steer the same starting noise with both, and compare the hypervolume of the results, normalized by that of the best offline designs. For the weighted sum we also refit with the features held fixed to those of the cache, which checks that Eq. (8) is implemented exactly.

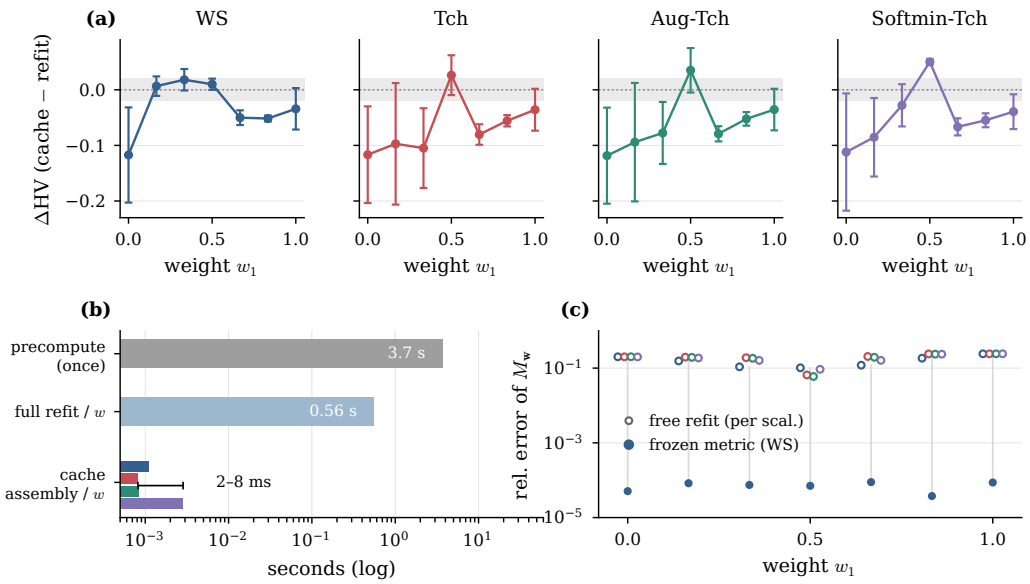

Figure 10: One precomputation serves every scalarization at a fraction of the cost of refitting. **(a)** HV of cache-based steering minus that of refit-based steering on ZDT2 (mean and 95% CI over five runs, grey band ± 0.02). The cache matches or beats the refit at the balanced weight and trails it by a few percent elsewhere, similarly for all four scalarizations. **(b)** Cost per weight on a log scale. The precomputation runs once, a refit takes 0.56 s per weight, and cache assembly takes about a millisecond. Neither path runs the flow to build $\widehat{M}_{\boldsymbol{w}}$. **(c)** Difference between the cache and refit matrices. With features held fixed, the refit reproduces the weighted-sum assembly to about 10^{-4}, so Eq. (8) is exact. A free refit differs by 6% to 24% for every scalarization, because it learns new features.

Results. Nonlinear scalarizations run from the cache just as the weighted sum does. They need no refit and no flow evaluations, and assembly costs less than 0.2% of a refit for every scalarization, about a millisecond against 0.56 s per weight (Figure 10b). The weighted sum assembles two to three times faster than the Tchebycheff family. The asymptotic costs of Section C.5, $O(m^2D^2)$ for the weighted sum and $O(nD^2)$ for the Tchebycheff family, suggest a much larger gap, but at this dimension ($D = 30$ on ZDT2) fixed overheads dominate both assemblies, so the measured gap is smaller. The weighted-sum assembly is also exact, matching a fixed-feature refit to within 0.02% at every weight (Figure 10c). The cache is not free in quality. A full refit learns features tailored to one weight, while the cache shares one set of features across all objectives, and this makes the refit somewhat better on average.

Measured as a percentage of the best offline HV and averaged over weights, the refit leads by 3.1% for the weighted sum and by 4.8% to 6.6% for the Tchebycheff family (Figure 10a). How large the gap is depends mainly on the weight. At the balanced weight the cache matches the refit for the weighted sum and beats it by 3% to 5% for the Tchebycheff family. Toward the ends of the simplex the refit leads, by 5% to 8% at $w_1 = \frac{2}{3}$ and $\frac{5}{6}$ and by about 12% at the corner $w_1 = 0$. This pattern is what shared features predict, since they fit mixtures of the objectives better than any single objective.

Takeaway. One precomputation serves linear and nonlinear scalarizations alike, at a per-weight cost more than 500 times below refitting. The price is a few percent of hypervolume away from the balanced weight, which a per-weight refit recovers when one specific trade-off matters.

Study 3. Full-sweep equivalence across scalarizations. Studies 1 and 2 look at one weight at a time. The main results are produced differently, by sweeping many weights and keeping the non-dominated candidates, so we repeat the comparison at that level. If the whole pipeline is run once per scalarization, do the four scalarizations end up with the same front quality?

Setup. Full Das–Dennis sweeps per scalarization at two steering budgets, $\|\Delta \boldsymbol{z}\|/\sqrt{D} \in \{0.3, 1\}$, on ZDT1–3 and DTLZ2, over five seeds. The Tchebycheff variants use the absolute-value convention. We report HV normalized by $\mathcal{D}(\text{best})$ and IGD against the analytic front, and compare each

Tchebycheff variant with the weighted sum at matched budget and seed, counting a difference within $\pm 2\%$ HV as equivalent.

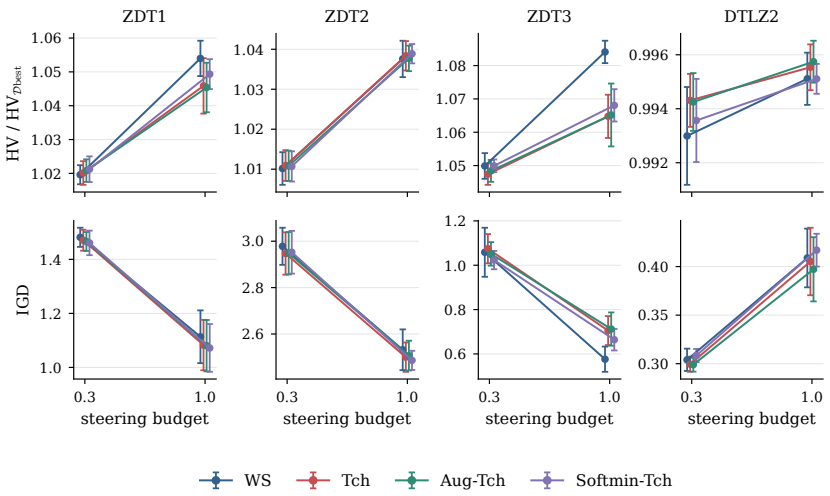

Figure 11: Every scalarization reaches the same front quality. Hypervolume relative to the best offline designs (top) and IGD to the analytic front (bottom) at two steering budgets, mean and 95% CI over five seeds. The four scalarizations agree within the 2% tolerance on every task. More steering helps on the ZDT tasks, whose optima lie on the design boundary, and hurts IGD on DTLZ2, whose optima lie in its interior.

Results. Irrespective of the scalarization, the full pipeline reaches the same front quality (Figure 11). All 24 comparisons (four tasks, two budgets, three Tchebycheff variants) fall within the 2% tolerance of the weighted sum in hypervolume, and in 21 of them the entire confidence interval does. The remaining three are ZDT3 at the larger budget, where the weighted sum leads by 1.6% to 1.9%, inside the tolerance on average but with intervals that extend slightly beyond it. On the ZDT tasks every scalarization ends 1% to 8% above the hypervolume of the best offline designs. On DTLZ2 steering improves on the unsteered samples but stays just below the offline best, at 99.3% to 99.6% of it.

A larger steering budget changes the results the same way for every scalarization, but not the same way on every task. It pushes more candidates onto the boundary of the design space, from about 40% to over 90% on the ZDT tasks and from 8% to 31% on DTLZ2. On ZDT the Pareto-optimal designs lie on that boundary, so a larger budget helps, raising hypervolume by 2% to 3% and lowering IGD by 15% to 37%. On DTLZ2 they lie in the interior, so a larger budget raises hypervolume by only 0.2% and worsens IGD by about 34%, because the generated points crowd toward the edges and cover less of the front. Since this happens identically under all four scalarizations, the complete approach of Section 5 behaves the same whichever scalarization drives it, which is the sweep-level version of Studies 1 and 2.

B.2.7 Steering beyond the offline Support on ZDT1

Figure 12 compares the solutions generated by RFM-AN with the best offline points and the true Pareto front on ZDT1. We use the weighted-sum sweep from Study 3 at two steering budgets, $\|\Delta \boldsymbol{z}\|_2/\sqrt{D} \in \{0.3, 1\}$. The offline points are the non-dominated subset used to compute $\mathcal{D}(\text{best})$. Both objectives are plotted in their original minimization convention.

With the smaller step, the generated frontier already extends below the best offline points. Increasing the step produces a clearer improvement, reaching lower f_2 values at comparable f_1 across much of the displayed range and moving the frontier closer to the true Pareto front. This gives a concrete example of steering beyond the trade-offs observed in the offline data: changing the initial noise alone is enough to reach better objective values, without adding guidance along the sampling trajectory.

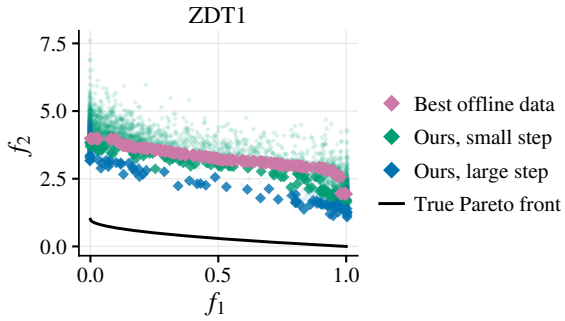

Figure 12: Steering beyond the offline front on ZDT1. Best offline points, RFM-AN samples under weighted-sum scalarization at steering budgets $\|\Delta \boldsymbol{z}\|_2/\sqrt{D} = 0.3$ and 1, and the true Pareto front. Both objectives are minimized. The larger step moves the generated frontier further beyond the offline points and toward the true front.

B.3 Guidance Ablations

Guidance is applied on top of noise-space steering, late in the trajectory (Section 4). Its main parameter is the guidance strength κ, which scales the guided velocity relative to the flow, $\eta(t) = \kappa\|\boldsymbol{v}_\theta\|/\|\mathcal{G}_{\boldsymbol{w}}\|$, or the ramp maximum $\eta_{\max}$ for RFM-APG. Theorem 2 predicts how HV should respond. For a small enough step the RFM-Cone direction improves every objective to first order, so HV should rise with the strength while the first-order term dominates. At large strengths the second-order term takes over and HV can fall, and where noise-space steering already reaches the front there is little left for guidance to add.

Setup. We sweep the guidance strength over $\{0, \frac{1}{4}, \frac{1}{2}, 1, 2, 4\}$ times its default for RFM-Cone, RFM-DAMG, and RFM-APG, on ZDT2, ZDT4, DTLZ2, DTLZ6, C-10/MOP8, IN-1K/MOP8, RE41, and MO-Swimmer, with five seeds. Steering is fixed at $\gamma = 10$ with soft weighting and $\alpha = 0.3$, and guidance starts at $t_{\text{start}} = 0.7$. RFM-APG starts at $t_{\text{start}} = 0.8$, and in this sweep its steering stage uses indicator weights. Its HV without guidance lies within 3% of that of the other operators on seven of eight tasks, and every result below is measured against each operator's own run without guidance. At strength 0 each operator reduces to steering alone, and we report the seed-paired change in normalized HV from that point, in percentage points (pp).

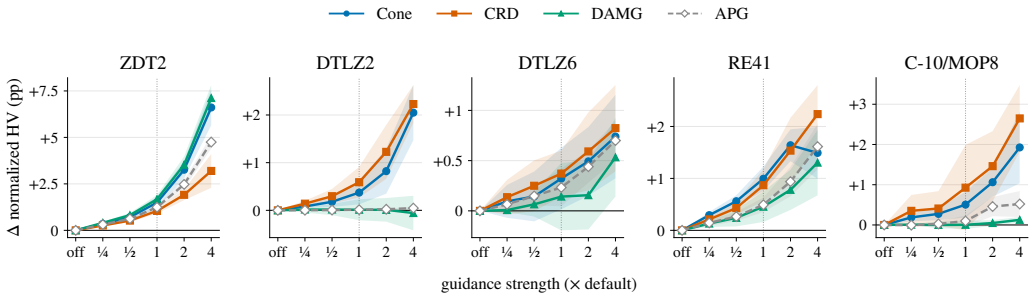

Figure 13: Guidance adds to steering and grows with its strength. Change in normalized HV over steering alone against the guidance strength, relative to its default (dotted line), on the five tasks where steering leaves headroom, mean and 95% CI over five seeds. The remaining three tasks are in Figure 14.

Results. Guidance improves on steering alone wherever steering leaves headroom (Figure 13). At the default strength RFM-Cone raises HV significantly on five of eight tasks, by 0.3 to 1.5 pp, and the gain keeps growing with the strength, reaching 6.6 pp on ZDT2 and about 2 pp on DTLZ2 and C-10/MOP8 at 4×. On the other three tasks RFM-Cone stays within 0.6 pp of steering alone up to 2× (Figure 14). Across all eight tasks RFM-Cone never loses HV significantly up to 2× its default, and a loss appears only at 4×, on MO-Swimmer (3.3 pp) and IN-1K/MOP8, which is the second-order turnover Theorem 2 predicts. RFM-DAMG is the more conservative operator. It gains 1.7 pp

on ZDT2 and 0.5 pp on RE41 at the default and stays within 0.3 pp of steering alone on C-10/MOP8, DTLZ2, and IN-1K/MOP8, so it acts where the geometry supports a step and otherwise leaves the steered sample unchanged. RFM-APG gains on the same tasks as RFM-Cone and loses only on ZDT4, by 3.0 pp at twice its default.

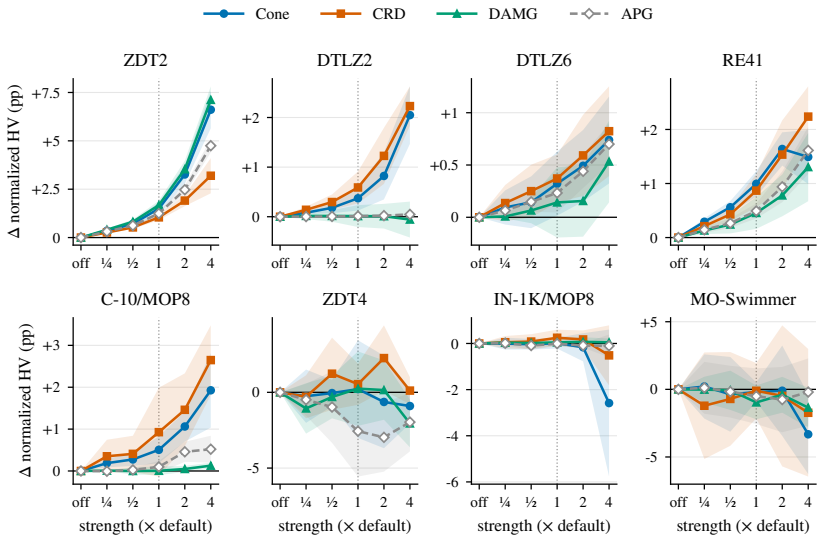

Figure 14: Guidance strength on all eight tasks. As Figure 13, including ZDT4, IN-1K/MOP8, and MO-Swimmer.

The gains are larger on the better half of the returned set (Figure 15). At the default strength RFM-Cone raises the 50th-percentile HV significantly on five tasks, by 1.1 to 2.1 pp, and on DTLZ2 by 21 pp at 4×. Guidance is also consistent across runs (Figure 16). At the default strength it improves on steering alone in 78% of task and seed pairs for RFM-Cone, 72% for RFM-APG, and 65% for RFM-DAMG, and every operator stays above 60% at every strength.

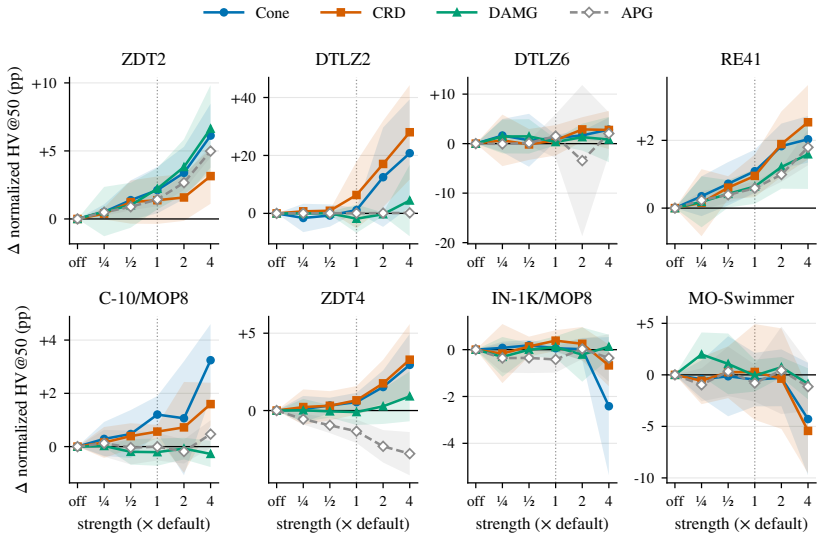

Figure 15: Guidance strength at the 50th percentile. As Figure 14, for the HV of the better half of the returned set.

Recommendation. Guidance is safe over an eightfold range of its strength, and the default is conservative. Doubling it raises HV further on five of eight tasks without a significant loss on any, so

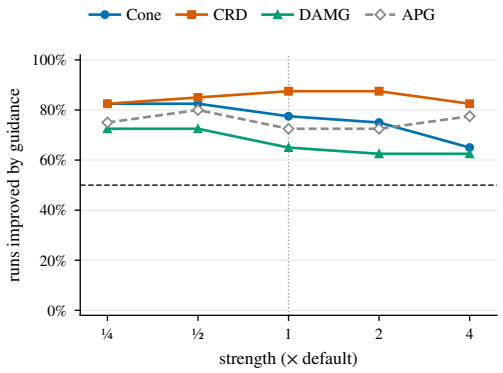

Figure 16: Guidance improves most runs. Fraction of task and seed pairs in which guidance raises HV over steering alone, against the guidance strength. The dashed line is chance.

the default leaves room for tasks where steering stops short of the front. The main results keep the default so that a single setting serves every task.

C Theoretical Results

C.1 Notation and setup

We maximize m objectives $\boldsymbol{f} = (f_1, \dots, f_m) : \mathcal{X} \to \mathbb{R}^m$ over a design space $\mathcal{X} \subseteq \mathbb{R}^D$, with access only to an offline dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$, $\boldsymbol{y}_i = \boldsymbol{f}(\boldsymbol{x}_i)$. No queries to $\boldsymbol{f}$ are permitted. We train proxies $\hat{\boldsymbol{f}} = (\hat{f}_1, \dots, \hat{f}_m)$ on $\mathcal{D}$. For a scalarization weight $\boldsymbol{w}$ in the simplex $\Delta^{m-1} = \{\boldsymbol{w} \in \mathbb{R}^m : w_a \geq 0,\ \sum_a w_a = 1\}$, write the scalarized proxy $h_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^\top \hat{\boldsymbol{f}}(\boldsymbol{x})$.

A rectified flow with velocity field $\boldsymbol{v}_\theta$ induces a *deterministic* sampler $\Phi_\theta : \mathbb{R}^D \to \mathbb{R}^D$, $\boldsymbol{z} \mapsto \boldsymbol{x}$, integrating the unmodified ODE from a seed $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D)$. We abbreviate $\mathcal{N} := \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D)$. Because Φ_θ is deterministic, every objective is equally a function of the seed. We therefore work throughout with the *noise-space* maps,

$$g_a(\boldsymbol{z}) := \hat{f}_a\big(\Phi_\theta(\boldsymbol{z})\big), \qquad H_{\boldsymbol{w}}(\boldsymbol{z}) := h_{\boldsymbol{w}}\big(\Phi_\theta(\boldsymbol{z})\big) = \sum_{a=1}^m w_a\, g_a(\boldsymbol{z}), \tag{5}$$

so that $\nabla H_{\boldsymbol{w}} = \sum_a w_a \nabla g_a$.

Noise-space geometry. Define the per-objective gradient-covariance blocks and the direction-specific metric

$$\boldsymbol{C}^\star_{ab} := \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\big[\nabla g_a(\boldsymbol{z})\, \nabla g_b(\boldsymbol{z})^\top\big] \in \mathbb{R}^{D\times D}, \qquad \boldsymbol{M}_{\boldsymbol{w}} := \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\big[\nabla H_{\boldsymbol{w}}\, \nabla H_{\boldsymbol{w}}^\top\big] = \sum_{a,b=1}^m w_a w_b\, \boldsymbol{C}^\star_{ab}. \tag{15}$$

The cached blocks $\boldsymbol{C}_{ab}$ of Eq. (8) estimate $\boldsymbol{C}^\star_{ab}$ through the RFM surrogates (Section C.4). As in Section 3.2, $\boldsymbol{M}_{\boldsymbol{w}}$ has eigenpairs $(\lambda_j^{(\boldsymbol{w})}, \boldsymbol{u}_{\boldsymbol{w}}^{(j)})$ with $\lambda_1^{(\boldsymbol{w})} \geq \cdots \geq \lambda_D^{(\boldsymbol{w})} \geq 0$, its top-$r$ eigenvectors form the *active frame* $\boldsymbol{U}_{\boldsymbol{w}} \in \mathbb{R}^{D\times r}$, and $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ is its estimate from the RFM cache.

Note, results in this appendix are stated for the population quantities $\boldsymbol{M}_{\boldsymbol{w}}, \lambda_j^{(\boldsymbol{w})}, \boldsymbol{u}_{\boldsymbol{w}}^{(j)}$. Algorithm 1 uses their estimates (hatted), and Section C.4 bounds the difference.

The rank r is not a free input but is read off the spectrum: we keep the fewest directions carrying all but an $\varepsilon_{\text{rank}}$-fraction of the objective's sensitivity.

Definition 1 (Effective rank)**.** *For a tolerance $\varepsilon_{\text{rank}} \in (0,1)$, the effective rank of $\boldsymbol{M}_{\boldsymbol{w}}$ is*

$$k_{\text{eff}}(\boldsymbol{w}) = \min\Big\{r \in \{1,\dots,D\} :\ \textstyle\sum_{j>r} \lambda_j^{(\boldsymbol{w})} < \varepsilon_{\text{rank}} \sum_{j=1}^D \lambda_j^{(\boldsymbol{w})}\Big\}. \tag{9}$$

For the hard cutoff we take $r = k_{\text{eff}}(\boldsymbol{w})$; when $k_{\text{eff}}(\boldsymbol{w}) \ll D$ the objective depends on only a few noise directions. The discarded tail $\sum_{j>r} \lambda_j^{(\boldsymbol{w})}$ is exactly the truncation error bounded in Proposition 3, so the rank rule and the error bound share one quantity. (A single fixed r across all $\boldsymbol{w}$, or the spectral-gap choice $r = \arg\min_j \lambda_{j+1}^{(\boldsymbol{w})}/\lambda_j^{(\boldsymbol{w})}$, are drop-in alternatives.)

Steering. Given the eigenbasis, a steering step is

$$\boldsymbol{z}_{\boldsymbol{w}} = \boldsymbol{z} + \gamma \sum_j \beta_j \, \sigma_j \, \boldsymbol{u}_{\boldsymbol{w}}^{(j)}, \qquad \hat{\boldsymbol{x}}_{\boldsymbol{w}} = \Phi_\theta(\boldsymbol{z}_{\boldsymbol{w}}), \tag{10}$$

integrated through the *unmodified* ODE, so $\hat{\boldsymbol{x}}_{\boldsymbol{w}}$ is an ordinary sample of the trained model. Here $\gamma > 0$ is the steering strength, $\boldsymbol{\beta} \geq \mathbf{0}$ are the direction weights, and $\boldsymbol{\sigma} \in \{\pm 1\}^D$ is a sign pattern fixed across $\boldsymbol{z}$ (chosen by the Pearson rule of Algorithm 1). The displacement has norm $R := \gamma \|\boldsymbol{\beta}\|_2$. We write $\boldsymbol{d}_{\boldsymbol{w}}(\boldsymbol{z}) := \boldsymbol{U}_{\boldsymbol{w}}^\top \nabla H_{\boldsymbol{w}}(\boldsymbol{z})$ for the active-subspace projection of the gradient.

C.2 The steering algorithm

Algorithm 1 runs in two phases. The one-time *precomputation* learns the noise-space geometry of the objectives: it draws seeds from the prior, decodes each through the flow, scores the resulting design with the proxies, and fits an RFM surrogate ϕ_a to every objective map g_a. The surrogate supplies cheap analytic gradients, so we obtain the gradient-covariance blocks $\boldsymbol{C}_{ab}$ without ever differentiating through Φ_θ. These blocks are the only quantities carried forward.

The *steering* phase produces one candidate per weight $\boldsymbol{w}$. We assemble $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ from the cached blocks and compute its eigenvectors $\widehat{\boldsymbol{u}}_{\boldsymbol{w}}^{(j)}$, which are the noise directions the scalarized objective is most sensitive to. Since the metric does not determine which way along each direction is uphill, we orient each eigenvector by the sign of its Pearson correlation with the scalarized proxy values on the cache. The noise is then displaced by a weighted steering step along these directions and decoded through the unmodified flow. The precomputation does not depend on $\boldsymbol{w}$, so a full sweep over weights reuses it, and each new weight costs one metric assembly, one eigendecomposition, and one ODE solve.

The base method is a single sign-selected steering step whose r directions are scaled by a weight vector $\boldsymbol{\beta} \geq \mathbf{0}$ (line 10). The two instances are an adaptive hard cutoff $\beta_j = \mathbf{1}[j \leq k_{\text{eff}}(\boldsymbol{w})]$ and a cutoff-free soft weighting $\beta_j = (\lambda_j^{(\boldsymbol{w})}/\lambda_1^{(\boldsymbol{w})})^\alpha$. The results of Section C.3 hold for either (Remark 3). Optionally, a *guidance operator* $\mathcal{G}_{\boldsymbol{w}}$ adds a correction $\eta(t)\,\mathcal{G}_{\boldsymbol{w}}$ to the velocity $\boldsymbol{v}_\theta$ late in the trajectory, for $t \geq t_{\text{start}}$ (line 11). The operators use the precomputed geometry or the per-objective proxy gradients and never differentiate through the ODE solve and they are analyzed in Section C.6. When no operator is supplied, the method reduces to one unmodified ODE solve.

C.3 First-order guarantee

We now show that a steering step provably increases the scalarized objective in expectation over the noise, and that the AGOP frame is the best rank-r frame in which to take that steering step. Both results rest on a *single* regularity condition, smoothness of $H_{\boldsymbol{w}}$. Sign selection, by contrast, is part of the algorithm and not an assumption. The theorem holds for any fixed sign pattern, and the Pearson rule of the Algorithm 1 targets the pattern that maximizes the resulting bound.

C.3.1 Assumptions

Assumption 1 (Smoothness)**.** *For each $\boldsymbol{w} \in \Delta^{m-1}$, the noise-space objective $H_{\boldsymbol{w}}$ is twice continuously differentiable (C^2), its gradient is square-integrable under $\mathcal{N}$, and its curvature is bounded, with $\|\nabla^2 H_{\boldsymbol{w}}(\boldsymbol{z})\|_2 \leq L_{\boldsymbol{w}}$ for every $\boldsymbol{z} \in \mathbb{R}^D$ and some constant $L_{\boldsymbol{w}} > 0$.*

Note, that this assumption is mild because it follows from the architecture rather than from the data. Since $H_{\boldsymbol{w}} = h_{\boldsymbol{w}} \circ \Phi_\theta$, it is C^2 with a bounded Hessian whenever the velocity field $\boldsymbol{v}_\theta$ and the proxies $\hat{f}_a$ have bounded first and second derivatives. Integrating such a velocity field over the finite interval $[0, 1]$ yields a flow map with bounded derivatives as well. Our flow model uses SiLU activations, whose first and second derivatives are bounded, so the finite weights the bound holds everywhere without further conditions.

Our proxies use LeakyReLU, which is not twice differentiable at zero. Replacing it by a smooth approximation, such as a softplus-based version, gives a proxy that satisfies Assumption 1 and stays uniformly close to the original. Theorem 1 then carries over to the original proxy up to a small error that vanishes as the approximation becomes exact. Furthermore, square-integrability of $\nabla H_{\boldsymbol{w}}$ makes the metric $\boldsymbol{M}_{\boldsymbol{w}} = \mathbb{E}[\nabla H_{\boldsymbol{w}} \nabla H_{\boldsymbol{w}}^\top]$ finite, and by the Gaussian Poincaré inequality it also makes $H_{\boldsymbol{w}}$ itself square-integrable under $\mathcal{N}$, so all expectations below are finite. The constant $L_{\boldsymbol{w}}$ is an *upper* bound on curvature, equivalently the Lipschitz constant of $\nabla H_{\boldsymbol{w}}$. A small $L_{\boldsymbol{w}}$ means the landscape bends gently, so the first-order approximation stays accurate over a longer steering step, while a large $L_{\boldsymbol{w}}$ forces a smaller steering strength. It is exactly what fixes the admissible steering strength in Theorem 1.

Sign selection, a device, not an assumption. The one thing the metric cannot supply is orientation. $\boldsymbol{M}_{\boldsymbol{w}}$ is built from outer products $\nabla H_{\boldsymbol{w}} \nabla H_{\boldsymbol{w}}^\top$ and is invariant to flipping the sign of any eigenvector, so it identifies *which* directions the objective is sensitive to but not *which way* along them is uphill. The algorithm resolves this once per direction, independently of the noise vector being steered, by setting σ_j to the sign of the Pearson correlation between $\langle \boldsymbol{z}_i, \boldsymbol{u}_{\boldsymbol{w}}^{(j)} \rangle$ and $\boldsymbol{w}^\top \hat{\boldsymbol{y}}_i$ over the cache, which requires no ODE solve. Theorem 1 does not depend on how the signs are chosen. Its inequality holds for every fixed sign pattern, and the signs only determine how large the guaranteed gain is. The Pearson rule is therefore a choice rather than a requirement, and a natural one (Beaglehole et al., 2026b), since its population value is exactly the pattern that makes the guaranteed gain largest. An empirical sign that disagrees with the population sign costs only the share of the gain carried by that direction (Remark 1).

What we do not assume. Two conditions are deliberately absent. First, we do *not* assume proxy fidelity ($\hat{f} \approx f$): Theorem 1 guarantees ascent on the *proxy* scalarization $H_{\boldsymbol{w}}$, and whether that tracks the true objective is a separate question, isolated to the recovery result (Theorem 3). Second, we do *not* assume the active subspace carries gradient signal at every seed. The theorem instead *reports* this boundary, since the expected gain is positive for a small enough steering strength unless the mean gradient $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}[\nabla H_{\boldsymbol{w}}]$ is orthogonal to $\mathrm{span}(\boldsymbol{U}_{\boldsymbol{w}})$. That degenerate case is the practically important failure mode. Because $\boldsymbol{U}_{\boldsymbol{w}}$ comes from the *averaged* metric, a similar failure can occur at an individual seed, where $\nabla H_{\boldsymbol{w}}(\boldsymbol{z})$ may be nearly orthogonal to $\mathrm{span}(\boldsymbol{U}_{\boldsymbol{w}})$ even when the subspace is correct on average (Remark 1).

C.3.2 Steering ascends the scalarized objective

Theorem 1 (Steering ascends the scalarized objective). *A single steering step moves uphill on $H_{\boldsymbol{w}}$ on average, by an amount set by how much of the mean gradient the weighted active directions capture. Precisely, let $\boldsymbol{u}_{\boldsymbol{w}}^{(1)}, \dots, \boldsymbol{u}_{\boldsymbol{w}}^{(D)}$ be the full orthonormal eigenbasis of $\boldsymbol{M}_{\boldsymbol{w}}$, let $\boldsymbol{\beta} \geq \boldsymbol{0}$ be the direction weights of Algorithm 1 (hard cutoff $\beta_j = \mathbf{1}[j \leq r]$ or soft $\beta_j = (\lambda_j^{(\boldsymbol{w})}/\lambda_1^{(\boldsymbol{w})})^\alpha$), and let $d_j(\boldsymbol{z}) := \boldsymbol{u}_{\boldsymbol{w}}^{(j)\top} \nabla H_{\boldsymbol{w}}(\boldsymbol{z})$. Under Assumption 1, the steering step $\boldsymbol{z}_{\boldsymbol{w}} = \boldsymbol{z} + \gamma \sum_j \beta_j \sigma_j \boldsymbol{u}_{\boldsymbol{w}}^{(j)}$ with any fixed sign pattern $\boldsymbol{\sigma} \in \{\pm 1\}^D$ satisfies, for every $\boldsymbol{z}$,*

$$H_{\boldsymbol{w}}(\boldsymbol{z}_{\boldsymbol{w}}) - H_{\boldsymbol{w}}(\boldsymbol{z}) \;\geq\; \gamma \sum_j \beta_j \sigma_j d_j(\boldsymbol{z}) \;-\; \tfrac{1}{2} L_{\boldsymbol{w}} \gamma^2 \|\boldsymbol{\beta}\|_2^2. \tag{16}$$

Let $\bar{\boldsymbol{d}}_{\boldsymbol{w}} \in \mathbb{R}^D$, with components $\bar{d}_j := \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}[d_j(\boldsymbol{z})]$, be the mean directional gradient. The fixed sign that maximizes the expected gain is $\sigma_j = \mathrm{sgn}(\bar{d}_j)$, which is the population value of the Pearson sign of Algorithm 1, and for it,

$$\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\big[H_{\boldsymbol{w}}(\boldsymbol{z}_{\boldsymbol{w}}) - H_{\boldsymbol{w}}(\boldsymbol{z})\big] \;\geq\; \gamma \,\langle \boldsymbol{\beta}, |\bar{\boldsymbol{d}}_{\boldsymbol{w}}| \rangle \;-\; \tfrac{1}{2} L_{\boldsymbol{w}} \gamma^2 \|\boldsymbol{\beta}\|_2^2. \tag{17}$$

Hence the steering step increases $H_{\boldsymbol{w}}$ in expectation for every $0 < \gamma < 2\gamma^\star$, where the optimal steering strength $\gamma^\star = \langle \boldsymbol{\beta}, |\bar{\boldsymbol{d}}_{\boldsymbol{w}}| \rangle / (L_{\boldsymbol{w}} \|\boldsymbol{\beta}\|_2^2)$ attains expected gain of at least $\langle \boldsymbol{\beta}, |\bar{\boldsymbol{d}}_{\boldsymbol{w}}| \rangle^2 / (2 L_{\boldsymbol{w}} \|\boldsymbol{\beta}\|_2^2)$. Such a γ exists whenever $\beta_j \bar{d}_j \neq 0$ for some j. For the hard cutoff Eq. (17) reduces to $\gamma \|\bar{\boldsymbol{d}}_{\boldsymbol{w},1:r}\|_1 - \frac{1}{2} L_{\boldsymbol{w}} r \gamma^2$, and no steering strength guarantees ascent exactly when $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}[\nabla H_{\boldsymbol{w}}] \perp \mathrm{span}(\boldsymbol{u}_{\boldsymbol{w}}^{(1)}, \dots, \boldsymbol{u}_{\boldsymbol{w}}^{(r)})$.

Proof. We first take the hard cutoff $\beta_j = \mathbf{1}[j \leq r]$, so the step is $\boldsymbol{z}_{\boldsymbol{w}} = \boldsymbol{z} + \gamma \boldsymbol{U}_{\boldsymbol{w}} \boldsymbol{\sigma}$ with $\boldsymbol{U}_{\boldsymbol{w}} = [\boldsymbol{u}_{\boldsymbol{w}}^{(1)} \cdots \boldsymbol{u}_{\boldsymbol{w}}^{(r)}]$ and $\boldsymbol{\sigma} \in \{\pm 1\}^r$. The general $\boldsymbol{\beta}$ is handled at the end.

Let $\boldsymbol{g} := \nabla H_{\boldsymbol{w}}(\boldsymbol{z})$, let $\boldsymbol{d} := \boldsymbol{d}_{\boldsymbol{w}}(\boldsymbol{z}) = \boldsymbol{U}_{\boldsymbol{w}}^{\top}\boldsymbol{g}$ be the gradient projected onto the active subspace (the directional gradient), and let $\boldsymbol{\delta} := \boldsymbol{z}_{\boldsymbol{w}} - \boldsymbol{z} = \gamma\, \boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{\sigma}$ be the step displacement. By Taylor's theorem with Lagrange remainder, Assumption 1 gives a point $\xi \in [\boldsymbol{z}, \boldsymbol{z}_{\boldsymbol{w}}]$ with,

$$H_{\boldsymbol{w}}(\boldsymbol{z}_{\boldsymbol{w}}) - H_{\boldsymbol{w}}(\boldsymbol{z}) = \boldsymbol{g}^{\top}\boldsymbol{\delta} + \tfrac{1}{2}\,\boldsymbol{\delta}^{\top}\nabla^2 H_{\boldsymbol{w}}(\xi)\,\boldsymbol{\delta}. \tag{18}$$

Expanding the linear term as,

$$\boldsymbol{g}^{\top}\boldsymbol{\delta} = \boldsymbol{g}^{\top}(\gamma\, \boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{\sigma}) = \gamma\, \boldsymbol{g}^{\top}\boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{\sigma} = \gamma\, (\boldsymbol{U}_{\boldsymbol{w}}^{\top}\boldsymbol{g})^{\top}\boldsymbol{\sigma} = \gamma\, \boldsymbol{d}^{\top}\boldsymbol{\sigma},$$

where we substituted $\boldsymbol{\delta} = \gamma\, \boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{\sigma}$ and substituted $\boldsymbol{d}$. Now, for the quadratic term of Eq. (18), norm of the step length or displacement,

$$\|\boldsymbol{\delta}\|_2^2 = \|\gamma\, \boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{\sigma}\|_2^2 = \gamma^2\, \|\boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{\sigma}\|_2^2 = \gamma^2\, \|\boldsymbol{\sigma}\|_2^2 = \gamma^2 r,$$

where the third equality uses orthonormality of the columns of $\boldsymbol{U}_{\boldsymbol{w}}$. Then, the bound $\boldsymbol{v}^{\top}A\boldsymbol{v} \geq -\|A\|_2\|\boldsymbol{v}\|_2^2$ for symmetric A applied to $A = \nabla^2 H_{\boldsymbol{w}}(\xi)$,

$$\tfrac{1}{2}\,\boldsymbol{\delta}^{\top}\nabla^2 H_{\boldsymbol{w}}(\xi)\,\boldsymbol{\delta} \geq -\tfrac{1}{2}\,\|\nabla^2 H_{\boldsymbol{w}}(\xi)\|_2\,\|\boldsymbol{\delta}\|_2^2 \geq -\tfrac{1}{2}\,L_{\boldsymbol{w}}\,\|\boldsymbol{\delta}\|_2^2 = -\tfrac{1}{2}\,L_{\boldsymbol{w}}\,r\,\gamma^2,$$

Combining the two terms, we get,

$$H_{\boldsymbol{w}}(\boldsymbol{z}_{\boldsymbol{w}}) - H_{\boldsymbol{w}}(\boldsymbol{z}) \;\geq\; \gamma\, \boldsymbol{d}^{\top}\boldsymbol{\sigma} - \tfrac{1}{2}\,L_{\boldsymbol{w}}r\gamma^2$$

for every sign pattern $\boldsymbol{\sigma}$. Taking expectations over $\boldsymbol{z} \sim \mathcal{N}$ on both sides for the above equation, the curvature term $-\frac{1}{2}L_{\boldsymbol{w}}r\gamma^2$ is the same constant for every $\boldsymbol{z}$ (Assumption 1), and $\boldsymbol{d}_{\boldsymbol{w}}$ is integrable since $\nabla H_{\boldsymbol{w}} \in L^2(\mathcal{N})$ so,

$$\mathbb{E}_{\boldsymbol{z}}\big[H_{\boldsymbol{w}}(\boldsymbol{z}_{\boldsymbol{w}}) - H_{\boldsymbol{w}}(\boldsymbol{z})\big] \;\geq\; \gamma\, \boldsymbol{\sigma}^{\top}\bar{\boldsymbol{d}}_{\boldsymbol{w}} - \tfrac{1}{2}L_{\boldsymbol{w}}r\gamma^2 \qquad \text{for every fixed } \boldsymbol{\sigma}$$

Over fixed sign patterns the right side is largest when $\boldsymbol{\sigma}^{\top}\bar{\boldsymbol{d}}_{\boldsymbol{w}} = \sum_j \sigma_j \bar{d}_j$ is, and $\sum_j \sigma_j \bar{d}_j \leq \sum_j |\bar{d}_j| = \|\bar{\boldsymbol{d}}_{\boldsymbol{w}}\|_1$ with equality at $\sigma_j = \operatorname{sgn}(\bar{d}_j)$, which gives Eq. (17).

Note, this is the sign Algorithm 1 computes. By Stein's lemma, for $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D)$ and $H_{\boldsymbol{w}}$ differentiable with $\mathbb{E}\|\nabla H_{\boldsymbol{w}}\|_2 < \infty$, $\mathbb{E}_{\boldsymbol{z}}[\boldsymbol{z}\, H_{\boldsymbol{w}}(\boldsymbol{z})] = \mathbb{E}_{\boldsymbol{z}}[\nabla H_{\boldsymbol{w}}(\boldsymbol{z})]$, therefore,

$$\mathbb{E}_{\boldsymbol{z}}\big[\langle \boldsymbol{z}, \boldsymbol{u}_{\boldsymbol{w}}^{(j)}\rangle\, H_{\boldsymbol{w}}(\boldsymbol{z})\big] = \boldsymbol{u}_{\boldsymbol{w}}^{(j)\top}\mathbb{E}_{\boldsymbol{z}}[\nabla H_{\boldsymbol{w}}] = \bar{d}_j.$$

As $\langle \boldsymbol{z}, \boldsymbol{u}_{\boldsymbol{w}}^{(j)}\rangle$ has mean zero, the left side is $\operatorname{Cov}\big(\langle \boldsymbol{z}, \boldsymbol{u}_{\boldsymbol{w}}^{(j)}\rangle, H_{\boldsymbol{w}}(\boldsymbol{z})\big)$, whose sign is the sign of the Pearson correlation between the coordinate along $\boldsymbol{u}_{\boldsymbol{w}}^{(j)}$ and the objective value. The algorithm uses its empirical version on the cached pairs $\big(\langle \boldsymbol{z}_i, \boldsymbol{u}_{\boldsymbol{w}}^{(j)}\rangle,\ \boldsymbol{w}^{\top}\hat{\boldsymbol{y}}_i\big)$, where $\boldsymbol{w}^{\top}\hat{\boldsymbol{y}}_i = H_{\boldsymbol{w}}(\boldsymbol{z}_i)$ exactly. Thus the Pearson sign equals $\operatorname{sgn}(\bar{d}_j)$ in the population, and its empirical version converges to it as n grows whenever $\bar{d}_j \neq 0$. At finite n the two may disagree, which Remark 1 addresses.

Note, that the bound Eq. (17) is concave in γ, maximized at $\gamma^{\star} = \|\bar{\boldsymbol{d}}_{\boldsymbol{w}}\|_1/(L_{\boldsymbol{w}}r)$ with value $\|\bar{\boldsymbol{d}}_{\boldsymbol{w}}\|_1^2/(2L_{\boldsymbol{w}}r)$, which is positive on $0 < \gamma < 2\|\bar{\boldsymbol{d}}_{\boldsymbol{w}}\|_1/(L_{\boldsymbol{w}}r)$, and $\|\bar{\boldsymbol{d}}_{\boldsymbol{w}}\|_1 = 0$ exactly when $\boldsymbol{U}_{\boldsymbol{w}}^{\top}\mathbb{E}_{\boldsymbol{z}}[\nabla H_{\boldsymbol{w}}] = \boldsymbol{0}$, i.e. $\mathbb{E}_{\boldsymbol{z}}[\nabla H_{\boldsymbol{w}}] \perp \operatorname{span}(\boldsymbol{U}_{\boldsymbol{w}})$. Also, since $L_{\boldsymbol{w}}$ is a worst-case constant, the guarantee explains why steering ascends rather than prescribing γ, which we set empirically (§ B.2.1).

General weights. For arbitrary $\boldsymbol{\beta} \geq \boldsymbol{0}$ the displacement is $\boldsymbol{\delta} = \gamma \sum_j \beta_j\sigma_j \boldsymbol{u}_{\boldsymbol{w}}^{(j)}$, and the argument above goes through with two replacements: the linear term is $\boldsymbol{g}^{\top}\boldsymbol{\delta} = \gamma \sum_j \beta_j\sigma_j d_j(\boldsymbol{z})$, and by orthonormality $\|\boldsymbol{\delta}\|_2^2 = \gamma^2 \sum_j \beta_j^2\sigma_j^2 = \gamma^2\|\boldsymbol{\beta}\|_2^2$ in place of $\gamma^2 r$. Taking expectations and maximizing over fixed signs gives $\sum_j \beta_j\sigma_j\bar{d}_j \leq \langle \boldsymbol{\beta}, |\bar{\boldsymbol{d}}_{\boldsymbol{w}}|\rangle$, with equality at $\sigma_j = \operatorname{sgn}(\bar{d}_j)$, which is Eq. (17). The Stein step remains unchanged, and $\gamma^{\star}$ follows by the same concavity argument with r replaced by $\|\boldsymbol{\beta}\|_2^2$. □

The guarantee is first-order and local, and it holds without further conditions. The ascent gain grows linearly in γ while the curvature penalty grows quadratically, so shrinking γ always tips the balance toward ascent, and the theorem records the admissible window and the optimal steering strength $\gamma^{\star}$. What sets the gain is $\langle \boldsymbol{\beta}, |\bar{\boldsymbol{d}}_{\boldsymbol{w}}|\rangle$, the weighted mean gradient the active directions capture. The sign $\boldsymbol{\sigma}$ only fixes which way the steering step moves along each direction, not how far. The bound is vacuous in exactly one case, when the mean gradient $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}[\nabla H_{\boldsymbol{w}}]$ is orthogonal to the weighted active directions. This is a property of the subspace rather than the step, and it is what the next result addresses: no rank-r subspace captures more of the gradient in expectation than $\operatorname{span}(\boldsymbol{U}_{\boldsymbol{w}})$.

C.3.3 THE ACTIVE SUBSPACE IS THE OPTIMAL STEERING SUBSPACE

Here we ask which r-dimensional subspace captures the most gradient, averaged over the noise $\boldsymbol{z} \sim \mathcal{N}$ the flow is driven by.

Proposition 1 (The active subspace captures the maximum gradient). *Let $\boldsymbol{V} \in \mathbb{R}^{D\times r}$ have orthonormal columns. Averaged over the noise $\boldsymbol{z} \sim \mathcal{N}$, the squared gradient the subspace* $\mathrm{span}(\boldsymbol{V})$ *captures is*

$$\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\big[\|\boldsymbol{V}^\top \nabla H_{\boldsymbol{w}}(\boldsymbol{z})\|_2^2\big] = \mathrm{tr}\big(\boldsymbol{V}^\top \boldsymbol{M}_{\boldsymbol{w}} \boldsymbol{V}\big) \;\le\; \sum_{j=1}^{r} \lambda_j^{(\boldsymbol{w})}, \tag{19}$$

with equality iff $\mathrm{span}(\boldsymbol{V})$ *is the span of the top-r eigenvectors of $\boldsymbol{M}_{\boldsymbol{w}}$, that is the active subspace* $\mathrm{span}(\boldsymbol{U}_{\boldsymbol{w}})$*, provided $\lambda_r^{(\boldsymbol{w})} > \lambda_{r+1}^{(\boldsymbol{w})}$ so that this span is unique. Hence no rank-r subspace captures more expected squared gradient than the active subspace, and the amount it captures is the top-r eigenvalue sum.*

The expected squared gradient it leaves out is exactly $\sum_{j>r} \lambda_j^{(\boldsymbol{w})}$ (Proposition 3).

Proof. Using $\|\boldsymbol{V}^\top \boldsymbol{g}\|_2^2 = \mathrm{tr}(\boldsymbol{V}^\top \boldsymbol{g}\boldsymbol{g}^\top \boldsymbol{V})$ for any vector $\boldsymbol{g}$, and linearity of expectation,

$$\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\big[\|\boldsymbol{V}^\top \nabla H_{\boldsymbol{w}}\|_2^2\big] = \mathrm{tr}\Big(\boldsymbol{V}^\top\, \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}[\nabla H_{\boldsymbol{w}} \nabla H_{\boldsymbol{w}}^\top]\, \boldsymbol{V}\Big) = \mathrm{tr}\big(\boldsymbol{V}^\top \boldsymbol{M}_{\boldsymbol{w}} \boldsymbol{V}\big),$$

where the last equality is the definition of $\boldsymbol{M}_{\boldsymbol{w}}$ in Eq. (15). By Ky Fan's maximum principle, over all $\boldsymbol{V}$ with orthonormal columns the maximum of $\mathrm{tr}(\boldsymbol{V}^\top \boldsymbol{M}_{\boldsymbol{w}} \boldsymbol{V})$ is the sum of the r largest eigenvalues of $\boldsymbol{M}_{\boldsymbol{w}}$, attained exactly when $\mathrm{span}(\boldsymbol{V})$ is the span of the corresponding top-r eigenvectors. □

Together, Theorem 1 and Proposition 1 say that steering along the top-r eigenvectors of $\boldsymbol{M}_{\boldsymbol{w}}$ ascends the scalarized objective, and that these directions capture more expected squared gradient than any other r directions. The remaining caveats of this first-order picture are collected below.

Remark 1 (Caveats of the first-order results)**.** *Three points qualify the two results above.*

- ***Global directions and sign.*** *$\boldsymbol{U}_{\boldsymbol{w}}$ comes from the* averaged *metric $\boldsymbol{M}_{\boldsymbol{w}}$, so it is the globally dominant subspace, not the ascent direction at a particular $\boldsymbol{z}$. The sign $\boldsymbol{\sigma}$ is also computed once from the cache and shared by every noise vector.*
- ***Sign errors degrade gracefully.*** *If S indexes the mis-signed directions, the first-order expected gain is $\gamma\big(\langle \boldsymbol{\beta}, |\bar{\boldsymbol{d}}_{\boldsymbol{w}}|\rangle - 2\sum_{j\in S} \beta_j |\bar{d}_j|\big)$, so the steering step still ascends whenever the mis-signed directions carry less than half the weighted mean gradient, $\sum_{j\in S} \beta_j |\bar{d}_j| < \frac{1}{2}\langle \boldsymbol{\beta}, |\bar{\boldsymbol{d}}_{\boldsymbol{w}}|\rangle$. This is benign because the high-signal directions, where an error costs most, are exactly where the Pearson sign is most reliable.*
- ***Degenerate case and reach.*** *The gain vanishes when $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}[\nabla H_{\boldsymbol{w}}] \perp \mathrm{span}(\boldsymbol{U}_{\boldsymbol{w}})$. This is what a single steering step cannot serve and motivates the complementary operators of Section C.6. The admissible steering step is also bounded, since $\|\boldsymbol{z}_{\boldsymbol{w}} - \boldsymbol{z}\|_2 = \gamma\|\boldsymbol{\beta}\|_2$ must stay in the range where Φ_θ decodes reliably (Remark 4).*

C.4 ESTIMATION AND TRUNCATION OF THE METRIC

Theorem 1 and Proposition 1 are stated for the population metric $\boldsymbol{M}_{\boldsymbol{w}}$. The algorithm, however, can only form the empirical AGOP $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ from n noise samples and the surrogate gradients, and it keeps only the top r of its eigenvectors or down-weights the rest. The next two results control both approximations: the empirical metric converges to its population target at the Monte Carlo rate, and the top-r truncation discards an amount of gradient equal to the tail of the spectrum.

Consistency of the empirical metric. The algorithm assembles $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$ from the surrogate gradients $\nabla\phi_{\boldsymbol{w}}$, $\phi_{\boldsymbol{w}} := \sum_a w_a \phi_a$, averaged over n noise samples. It is therefore an estimator of the *population surrogate metric* $\boldsymbol{M}_{\boldsymbol{w}}^{\phi} := \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}[\nabla\phi_{\boldsymbol{w}}(\boldsymbol{z}) \nabla\phi_{\boldsymbol{w}}(\boldsymbol{z})^\top]$. We prove convergence to $\boldsymbol{M}_{\boldsymbol{w}}^{\phi}$ at the standard rate, and afterwards separate this estimation error from the two approximation gaps the estimator does not control.

Proposition 2 (**The empirical AGOP is consistent**). *Let $\boldsymbol{z}_1, \dots, \boldsymbol{z}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_D)$ be i.i.d. and independent of the surrogate $\phi_{\boldsymbol{w}}$, which we treat as fixed, and let $\widehat{\boldsymbol{M}}_{\boldsymbol{w}} := \frac{1}{n}\sum_{i=1}^n \nabla\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)\nabla\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)^\top$ be the empirical AGOP, an estimator of $\boldsymbol{M}_{\boldsymbol{w}}^\phi = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}[\nabla\phi_{\boldsymbol{w}}\nabla\phi_{\boldsymbol{w}}^\top]$. If the fourth moment $\kappa_{\boldsymbol{w}} := \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\|\nabla\phi_{\boldsymbol{w}}\|_2^4$ is finite, then,*

(i) (consistency) $\widehat{\boldsymbol{M}}_{\boldsymbol{w}} \to \boldsymbol{M}_{\boldsymbol{w}}^\phi$ *almost surely as* $n \to \infty$

(ii) (rate) $\mathbb{E}\big\|\widehat{\boldsymbol{M}}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^\phi\big\|_F \le \sqrt{\kappa_{\boldsymbol{w}}/n}$.

(iii) (direction recovery) *The steering weights are consistent, and for soft weighting no eigengap is needed: the weighted projector $\mathbf{R}_{\boldsymbol{w}} := \boldsymbol{U}_{\boldsymbol{w}}\operatorname{diag}(\boldsymbol{\beta})\boldsymbol{U}_{\boldsymbol{w}}^\top$ with $\beta_j = (\lambda_j/\lambda_1)^\alpha$ satisfies $\|\widehat{\mathbf{R}}_{\boldsymbol{w}} - \mathbf{R}_{\boldsymbol{w}}\|_F = O_p\big(n^{-\min(\alpha,1)/2}\big)$. For a hard top-$r$ cutoff the retained subspace satisfies $\|\sin\Theta(\widehat{\boldsymbol{U}}_{\boldsymbol{w}}, \boldsymbol{U}_{\boldsymbol{w}}^\phi)\|_2 = O_p\big(n^{-1/2}/\Delta_r\big)$, provided the eigengap $\Delta_r := \lambda_r(\boldsymbol{M}_{\boldsymbol{w}}^\phi) - \lambda_{r+1}(\boldsymbol{M}_{\boldsymbol{w}}^\phi) > 0$.*

Proof. Let's set $\mathbf{A}_i := \nabla\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)\nabla\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)^\top$. Since the $\boldsymbol{z}_i$ are i.i.d., so are the $\mathbf{A}_i$, and by the definition of $\boldsymbol{M}_{\boldsymbol{w}}^\phi$ each has mean $\mathbb{E}\mathbf{A}_i = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}[\nabla\phi_{\boldsymbol{w}}\nabla\phi_{\boldsymbol{w}}^\top] = \boldsymbol{M}_{\boldsymbol{w}}^\phi$.

(i) Suppose the second moment $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\|\nabla\phi_{\boldsymbol{w}}\|_2^2 < \infty$ (implied by the assumed finite fourth moment). If we assume a fixed entry (k,l), the scalars $(\mathbf{A}_i)_{kl} = \partial_k\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)\,\partial_l\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)$ are i.i.d. with mean $(\boldsymbol{M}_{\boldsymbol{w}}^\phi)_{kl}$. Each is bounded in absolute value by the squared gradient norm, since

$$|(\mathbf{A}_i)_{kl}| = |\partial_k\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)|\,|\partial_l\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)| \le \|\nabla\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)\|_2\,\|\nabla\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)\|_2 = \|\nabla\phi_{\boldsymbol{w}}(\boldsymbol{z}_i)\|_2^2,$$

using $|\partial_k\phi_{\boldsymbol{w}}| \le \|\nabla\phi_{\boldsymbol{w}}\|_2$ for every coordinate. Taking expectations, $\mathbb{E}|(\mathbf{A}_i)_{kl}| \le \mathbb{E}\|\nabla\phi_{\boldsymbol{w}}\|_2^2 < \infty$, so each $(\mathbf{A}_i)_{kl}$ is integrable. The strong law of large numbers then gives $(\widehat{\boldsymbol{M}}_{\boldsymbol{w}})_{kl} = \frac{1}{n}\sum_i(\mathbf{A}_i)_{kl} \to (\boldsymbol{M}_{\boldsymbol{w}}^\phi)_{kl}$ almost surely, and since there are finitely many entries, $\widehat{\boldsymbol{M}}_{\boldsymbol{w}} \to \boldsymbol{M}_{\boldsymbol{w}}^\phi$ almost surely, hence in Frobenius norm.

(ii) Let $\mathbf{Y}_i := \mathbf{A}_i - \boldsymbol{M}_{\boldsymbol{w}}^\phi$, which are i.i.d. and mean zero. Writing the squared error through the Frobenius inner product,

$$\mathbb{E}\big\|\widehat{\boldsymbol{M}}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^\phi\big\|_F^2 = \mathbb{E}\Big\|\tfrac{1}{n}\textstyle\sum_i \mathbf{Y}_i\Big\|_F^2 = \tfrac{1}{n^2}\textstyle\sum_{i,j}\mathbb{E}\langle\mathbf{Y}_i, \mathbf{Y}_j\rangle.$$

For $i \ne j$, cancel out due to independence and $\mathbb{E}\mathbf{Y}_i = 0$ gives $\mathbb{E}\langle\mathbf{Y}_i, \mathbf{Y}_j\rangle = \langle\mathbb{E}\mathbf{Y}_i, \mathbb{E}\mathbf{Y}_j\rangle = 0$, so only the n diagonal terms survive:

$$\mathbb{E}\big\|\widehat{\boldsymbol{M}}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^\phi\big\|_F^2 = \frac{1}{n}\,\mathbb{E}\|\mathbf{Y}_1\|_F^2 \le \frac{1}{n}\,\mathbb{E}\|\mathbf{A}_1\|_F^2 = \frac{1}{n}\,\mathbb{E}\|\nabla\phi_{\boldsymbol{w}}\|_2^4 = \frac{\kappa_{\boldsymbol{w}}}{n},$$

using $\mathbb{E}\|\mathbf{Y}_1\|_F^2 = \mathbb{E}\|\mathbf{A}_1\|_F^2 - \|\boldsymbol{M}_{\boldsymbol{w}}^\phi\|_F^2 \le \mathbb{E}\|\mathbf{A}_1\|_F^2$ and $\|\boldsymbol{v}\boldsymbol{v}^\top\|_F = \|\boldsymbol{v}\|_2^2$ for the rank-one $\mathbf{A}_1$. Finally, Jensen's inequality gives $\mathbb{E}\|\widehat{\boldsymbol{M}}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^\phi\|_F \le \sqrt{\kappa_{\boldsymbol{w}}/n}$.

(iii) The algorithm steers with the span of the top-r eigenvectors of $\widehat{\boldsymbol{M}}_{\boldsymbol{w}}$, which we compare to that of $\boldsymbol{M}_{\boldsymbol{w}}^\phi$. Parts (i)–(ii) give convergence of the matrix, but that does not by itself transfer to the object we steer with, because that object is built from the eigenvectors, which need not depend continuously on the matrix. The two weighting schemes differ exactly here.

Soft weighting forms the steering operator $\mathbf{R}_{\boldsymbol{w}} = \sum_j \beta_j\,\boldsymbol{u}_{\boldsymbol{w}}^{(j)}\boldsymbol{u}_{\boldsymbol{w}}^{(j)\top} = \psi(\boldsymbol{M}_{\boldsymbol{w}}^\phi)$, the matrix function applying $\beta(\lambda) = (\lambda/\lambda_1)^\alpha$ to the eigenvalues. A matrix function is continuous in the matrix even across eigenvalue crossings i.e. individual eigenprojectors jump at a crossing, but their weighted sum does not, so no gap is needed. For Lipschitz β ($\alpha \ge 1$ on the bounded spectrum) ψ is Lipschitz in Frobenius norm, hence $\|\widehat{\mathbf{R}}_{\boldsymbol{w}} - \mathbf{R}_{\boldsymbol{w}}\|_F \le L\,\|\widehat{\boldsymbol{M}}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^\phi\|_F = O_p(n^{-1/2})$ by (ii) and Markov. But for $0 < \alpha < 1$, β is only α-Hölder at 0, and the operator-Hölder inequality for matrix powers (Andô, 1988) gives $\|\widehat{\mathbf{R}}_{\boldsymbol{w}} - \mathbf{R}_{\boldsymbol{w}}\|_F \le C\,\|\widehat{\boldsymbol{M}}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^\phi\|_F^\alpha = O_p(n^{-\alpha/2})$. In either case the rate is $O_p(n^{-\min(\alpha,1)/2})$ with no eigengap.

The hard cutoff instead keeps a sharp top-r subspace, whose basis is a discontinuous eigenprojector. The instability is a near-tie: when λ_r and λ_{r+1} are close, the "top-r versus rest" split is almost a tie, and an arbitrarily small perturbation can flip which directions are retained, rotating the subspace by a large angle. A clear gap $\lambda_r - \lambda_{r+1} > 0$ rules the tie out and makes the subspace move only in proportion to the perturbation.

Let $\Delta_r = \lambda_r(\boldsymbol{M}_{\boldsymbol{w}}^{\phi}) - \lambda_{r+1}(\boldsymbol{M}_{\boldsymbol{w}}^{\phi})$ be that gap, and measure the distance between the empirical and population top-r subspaces by the principal angles Θ between them, so that $\|\sin\Theta\|_2 = \sin\theta_{\max}$ is 0 iff the subspaces coincide. Given $\Delta_r > 0$, the Davis–Kahan $\sin\Theta$ theorem, in the form of Yu et al. (2015), bounds the rotation by the perturbation over the gap as,

$$\|\sin\Theta(\widehat{\boldsymbol{U}}_{\boldsymbol{w}}, \boldsymbol{U}_{\boldsymbol{w}}^{\phi})\|_2 \le \|\sin\Theta(\widehat{\boldsymbol{U}}_{\boldsymbol{w}}, \boldsymbol{U}_{\boldsymbol{w}}^{\phi})\|_F \le \frac{2\,\|\widehat{\boldsymbol{M}}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^{\phi}\|_F}{\Delta_r}$$

where $\widehat{\boldsymbol{U}}_{\boldsymbol{w}}$ and $\boldsymbol{U}_{\boldsymbol{w}}^{\phi}$ are orthonormal top-r eigenvector bases of the empirical and population metrics. Since $\|\widehat{\boldsymbol{M}}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^{\phi}\|_F = O_p(n^{-1/2})$ by (ii) and using Markov's inequality, the right-hand side is $O_p(n^{-1/2}/\Delta_r)$. □

Remark 2 (Two gaps we do not close)**.** *The estimator converges to $\boldsymbol{M}_{\boldsymbol{w}}^{\phi}$, the metric of the* surrogate *gradients. Two further gaps separate it from the true-objective metric $\boldsymbol{M}_{\boldsymbol{w}}^{f} := \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}[\nabla H_{\boldsymbol{w}}^{f} \nabla H_{\boldsymbol{w}}^{f\top}]$, with $H_{\boldsymbol{w}}^{f} := \sum_a w_a\, f_a \circ \Phi_\theta$, and we name rather than prove them:*

- ***RFM approximation gap*** *$\|\boldsymbol{M}_{\boldsymbol{w}}^{\phi} - \boldsymbol{M}_{\boldsymbol{w}}\|$: the surrogate gradient $\nabla\phi_{\boldsymbol{w}}$ differs from the proxy's composed gradient $\nabla H_{\boldsymbol{w}}$. It is controlled by the RFM gradient error $\|\nabla\phi_{\boldsymbol{w}} - \nabla H_{\boldsymbol{w}}\|_{L^2(\mathcal{N})}$, which we take as an assumption. AGOP estimators built from kernel regression fits are known to be consistent (Trivedi et al., 2014), and RFM provably recovers the relevant directions in the linear setting (Radhakrishnan et al., 2025).*
- ***Proxy gap*** *$\|\boldsymbol{M}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^{f}\|$: the proxy $\hat{f}$ differs from the true objective f. This is the offline problem itself and cannot be closed without querying f. Its effect on the recovered front is isolated to Theorem 3.*

The total error splits accordingly as

$$\widehat{\boldsymbol{M}}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^{f} = \underbrace{(\widehat{\boldsymbol{M}}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^{\phi})}_{\text{(a) Monte-Carlo error}} + \underbrace{(\boldsymbol{M}_{\boldsymbol{w}}^{\phi} - \boldsymbol{M}_{\boldsymbol{w}})}_{\text{(b) RFM approximation}} + \underbrace{(\boldsymbol{M}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^{f})}_{\text{(c) proxy error}} \tag{20}$$

where (a) is proved $O(n^{-1/2})$, (b) is assumed supported by Trivedi et al. (2014); Radhakrishnan et al. (2025), and (c) is irreducible for offline MOO without additional assumptions.

The decomposition becomes a bound once the two named gaps are written explicitly.

Corollary 1 (**Total metric error**)**.** *Write $\|\boldsymbol{h}\|_{L^2(\mathcal{N})} := (\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\|\boldsymbol{h}(\boldsymbol{z})\|_2^2)^{1/2}$ for the root-mean-square size of a function under the noise, and let $\varepsilon_{\mathrm{rfm}} := \|\nabla\phi_{\boldsymbol{w}} - \nabla H_{\boldsymbol{w}}\|_{L^2(\mathcal{N})}$ and $\varepsilon_{\nabla} := \|\nabla H_{\boldsymbol{w}} - \nabla H_{\boldsymbol{w}}^{f}\|_{L^2(\mathcal{N})}$ be the RFM and proxy gradient errors, with gradient scales $\Gamma_{\boldsymbol{w}} := \|\nabla\phi_{\boldsymbol{w}}\|_{L^2} + \|\nabla H_{\boldsymbol{w}}\|_{L^2}$ and $\Gamma'_{\boldsymbol{w}} := \|\nabla H_{\boldsymbol{w}}\|_{L^2} + \|\nabla H_{\boldsymbol{w}}^{f}\|_{L^2}$. Then the empirical AGOP satisfies*

$$\mathbb{E}\big\|\widehat{\boldsymbol{M}}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^{f}\big\|_F \le \underbrace{\sqrt{\kappa_{\boldsymbol{w}}/n}}_{\text{Monte Carlo}} + \underbrace{\varepsilon_{\mathrm{rfm}}\,\Gamma_{\boldsymbol{w}}}_{\text{RFM approximation}} + \underbrace{\varepsilon_{\nabla}\,\Gamma'_{\boldsymbol{w}}}_{\text{proxy}}. \tag{21}$$

The first term vanishes as $n \to \infty$ (Proposition 2), the second as the surrogate improves, and the third is the irreducible offline gap.

Proof. By the triangle inequality on the three-way split of Remark 2, it suffices to bound each piece. The first, $\mathbb{E}\|\widehat{\boldsymbol{M}}_{\boldsymbol{w}} - \boldsymbol{M}_{\boldsymbol{w}}^{\phi}\|_F \le \sqrt{\kappa_{\boldsymbol{w}}/n}$, is Proposition 2(ii). For the second, the rank-one identity $\boldsymbol{u}\boldsymbol{u}^\top - \boldsymbol{v}\boldsymbol{v}^\top = \boldsymbol{u}(\boldsymbol{u}-\boldsymbol{v})^\top + (\boldsymbol{u}-\boldsymbol{v})\boldsymbol{v}^\top$ with $\boldsymbol{u} = \nabla\phi_{\boldsymbol{w}}$, $\boldsymbol{v} = \nabla H_{\boldsymbol{w}}$ gives, under expectation and the triangle inequality,

$$\|\boldsymbol{M}_{\boldsymbol{w}}^{\phi} - \boldsymbol{M}_{\boldsymbol{w}}\|_F \le \mathbb{E}\big[\|\nabla\phi_{\boldsymbol{w}} - \nabla H_{\boldsymbol{w}}\|_2\,(\|\nabla\phi_{\boldsymbol{w}}\|_2 + \|\nabla H_{\boldsymbol{w}}\|_2)\big] \le \varepsilon_{\mathrm{rfm}}\,\Gamma_{\boldsymbol{w}},$$

The last step by Cauchy–Schwarz in $L^2(\mathcal{N})$. The third piece is identical with $\nabla H_{\boldsymbol{w}}$ in place of $\nabla\phi_{\boldsymbol{w}}$ and $\nabla H_{\boldsymbol{w}}^{f}$ in place of $\nabla H_{\boldsymbol{w}}$, giving $\varepsilon_{\nabla}\,\Gamma'_{\boldsymbol{w}}$. □

C.4.1 TRUNCATION ERROR OF THE ACTIVE SUBSPACE

Under the hard cutoff, steering keeps only the top-r eigenvectors of $\boldsymbol{M}_{\boldsymbol{w}}$ and discards the rest. Soft weights are treated in Remark 3. The next result quantifies exactly what is lost, the expected squared gradient the retained subspace cannot see equals the discarded tail of the spectrum, the quantity the effective-rank rule (Definition 1) thresholds.

Proposition 3 (**Rank truncation discards the spectral tail**). *Let* $\mathbf{P}_{\boldsymbol{w}}^{\perp} := \boldsymbol{I}_D - \boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{U}_{\boldsymbol{w}}^{\top}$ *project onto the orthogonal complement of the active subspace. Then*

$$\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\big[\|\mathbf{P}_{\boldsymbol{w}}^{\perp}\nabla H_{\boldsymbol{w}}(\boldsymbol{z})\|_2^2\big] = \sum_{j>r}\lambda_j^{(\boldsymbol{w})}. \tag{22}$$

Together with the captured energy $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\|\boldsymbol{U}_{\boldsymbol{w}}^{\top}\nabla H_{\boldsymbol{w}}\|_2^2 = \sum_{j\le r}\lambda_j^{(\boldsymbol{w})}$ *(Proposition 1), the captured and discarded parts sum to the total* $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\|\nabla H_{\boldsymbol{w}}\|_2^2 = \sum_j \lambda_j^{(\boldsymbol{w})}$*, and the effective-rank rule keeps the discarded* fraction *below* $\varepsilon_{\mathrm{rank}}$.

Proof. Let's define $\mathbf{P}_{\boldsymbol{w}}^{\perp} = \sum_{j>r}\boldsymbol{u}_{\boldsymbol{w}}^{(j)}\boldsymbol{u}_{\boldsymbol{w}}^{(j)\top}$, which is symmetric and idempotent.
Using $\|\mathbf{P}_{\boldsymbol{w}}^{\perp}\boldsymbol{g}\|_2^2 = \mathrm{tr}(\mathbf{P}_{\boldsymbol{w}}^{\perp}\boldsymbol{g}\boldsymbol{g}^{\top}\mathbf{P}_{\boldsymbol{w}}^{\perp})$ and linearity of expectation,

$$\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\big[\|\mathbf{P}_{\boldsymbol{w}}^{\perp}\nabla H_{\boldsymbol{w}}\|_2^2\big] = \mathrm{tr}\big(\mathbf{P}_{\boldsymbol{w}}^{\perp}\boldsymbol{M}_{\boldsymbol{w}}\mathbf{P}_{\boldsymbol{w}}^{\perp}\big).$$

The trace simplifies in three steps. By cyclicity of the trace, rotate the last factor to the front. Then group the two projectors and then apply idempotence $(\mathbf{P}_{\boldsymbol{w}}^{\perp})^2 = \mathbf{P}_{\boldsymbol{w}}^{\perp}$:

$$\mathrm{tr}\big(\mathbf{P}_{\boldsymbol{w}}^{\perp}\boldsymbol{M}_{\boldsymbol{w}}\mathbf{P}_{\boldsymbol{w}}^{\perp}\big) = \mathrm{tr}\big(\mathbf{P}_{\boldsymbol{w}}^{\perp}\mathbf{P}_{\boldsymbol{w}}^{\perp}\boldsymbol{M}_{\boldsymbol{w}}\big) = \mathrm{tr}\big((\mathbf{P}_{\boldsymbol{w}}^{\perp})^2\boldsymbol{M}_{\boldsymbol{w}}\big) = \mathrm{tr}\big(\mathbf{P}_{\boldsymbol{w}}^{\perp}\boldsymbol{M}_{\boldsymbol{w}}\big).$$

Finally, expanding $\mathbf{P}_{\boldsymbol{w}}^{\perp}$ and using $\boldsymbol{M}_{\boldsymbol{w}}\boldsymbol{u}_{\boldsymbol{w}}^{(j)} = \lambda_j^{(\boldsymbol{w})}\boldsymbol{u}_{\boldsymbol{w}}^{(j)}$,

$$\mathrm{tr}\big(\mathbf{P}_{\boldsymbol{w}}^{\perp}\boldsymbol{M}_{\boldsymbol{w}}\big) = \sum_{j>r}\boldsymbol{u}_{\boldsymbol{w}}^{(j)\top}\boldsymbol{M}_{\boldsymbol{w}}\boldsymbol{u}_{\boldsymbol{w}}^{(j)} = \sum_{j>r}\lambda_j^{(\boldsymbol{w})}.$$

Because the discarded mean squared gradient is exactly the tail $\sum_{j>r}\lambda_j^{(\boldsymbol{w})}$, choosing $r = k_{\mathrm{eff}}(\boldsymbol{w})$ in Definition 1 is precisely the statement that steering leaves out less than an $\varepsilon_{\mathrm{rank}}$-fraction of the objective's gradient. □

Remark 3 (Both weightings are covered). *All results of this section hold for the weighted step* $\boldsymbol{z}_{\boldsymbol{w}} = \boldsymbol{z} + \gamma\sum_j \beta_j\sigma_j\boldsymbol{u}_{\boldsymbol{w}}^{(j)}$ *under either the hard cutoff or the soft weighting of Algorithm 1. Theorem 1 is stated for general* $\boldsymbol{\beta}$*, and the discarded gradient of Proposition 3 becomes* $\sum_j(1-\beta_j)^2\lambda_j^{(\boldsymbol{w})}$*, recovering the hard-cutoff form* $\sum_{j>r}\lambda_j^{(\boldsymbol{w})}$*. The decomposition is unchanged, and soft weighting is moreover gap-free in Proposition 2(iii). The weights also enter the curvature penalty of Theorem 1 through* $\|\boldsymbol{\beta}\|_2^2$*. Under the hard cutoff this equals* r*, so every added direction adds a full unit to the penalty, while under soft weights a direction adds only* $(\lambda_j^{(\boldsymbol{w})}/\lambda_1^{(\boldsymbol{w})})^{2\alpha}$*, which is negligible for weak directions.*

C.5 PARETO-FRONT RECOVERY

The results so far concern a single step at a single weight. It remains to show that sweeping the weight $\boldsymbol{w}$ traces out the Pareto front, which chains the ascent, truncation, and decomposition results into a recovery guarantee. But below remark needs to be stated before moving ahead, since it enters that guarantee as an error term rather than an assumption.

Remark 4 (Step size and flow reliability). *The steering step has a definite size,*

$$\|\boldsymbol{z}_{\boldsymbol{w}} - \boldsymbol{z}\|_2 = \|\gamma\sum_j \beta_j\sigma_j\boldsymbol{u}_{\boldsymbol{w}}^{(j)}\|_2 = \gamma\|\boldsymbol{\beta}\|_2$$

(orthonormal $\boldsymbol{u}_{\boldsymbol{w}}^{(j)}$*,* $\sigma_j^2 = 1$*), which for the hard cutoff is* $\gamma\sqrt{r}$*. The flow* Φ_θ *is a deterministic map defined on all of* $\mathbb{R}^D$*, so any step is well-defined, but it was trained on samples from the prior: as*

$\gamma\|\boldsymbol{\beta}\|_2$ grows and the steered noise moves away from that region, the flow reconstructs less reliably and the proxy-through-the-flow value becomes less trustworthy. We do not treat this as a hard limit; its effect enters the recovery bound through the reach term $\rho(\boldsymbol{w})$ and the proxy error, both of which grow smoothly with the steering strength rather than as a cliff.

Since each coordinate of $\boldsymbol{z} \sim \mathcal{N}$ has unit variance, we report the per-coordinate displacement $\gamma\|\boldsymbol{\beta}\|_2/\sqrt{D}$, which makes steering steps comparable across tasks of different dimension.

C.5.1 Pareto-front recovery as distance to an oracle

Steering ascends the scalarized proxy at each weight (Theorem 1) along the directions that capture the most gradient (Proposition 1), and sweeping the weight is cheap by reuse (Eq. (8)). We now state the result that the algorithm approximates the Pareto front, phrased as a distance to the best achievable output.

Fix a noise vector $\boldsymbol{z}_0 \sim \mathcal{N}$ with $\|\boldsymbol{z}_0\|_2 \le 2\sqrt{D}$, a maximal reach $R_{\max} \ge R = \gamma\|\boldsymbol{\beta}\|_2$ (Remark 4), and the flow's reachable set $\mathcal{X}_\Phi := \Phi_\theta(B)$, $B := \{\boldsymbol{z} : \|\boldsymbol{z}\|_2 \le 2\sqrt{D} + R_{\max}\}$, whcih does not depend on steering strength. We assume $\boldsymbol{f}$ is continuous, so the maximizer below exists. Let's define the *oracle* as the true per-weight maximizer $\boldsymbol{x}^\star_{\boldsymbol{w}} := \arg\max_{\boldsymbol{x}\in\mathcal{X}_\Phi} \boldsymbol{w}^\top f(\boldsymbol{x})$, with any noise vector $\boldsymbol{z}^o_{\boldsymbol{w}} \in B$ such that $\Phi_\theta(\boldsymbol{z}^o_{\boldsymbol{w}}) = \boldsymbol{x}^\star_{\boldsymbol{w}}$, and the oracle front $\mathcal{F}^\star_\Phi := \{f(\boldsymbol{x}^\star_{\boldsymbol{w}}) : \boldsymbol{w} \in \Delta^{m-1}\}$. Steering moves within the slice $S_{\boldsymbol{w}} := \{\boldsymbol{z}_0 + \boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{s} : \|\boldsymbol{s}\|_2 \le R\} \subset B$, and our produced output at weight $\boldsymbol{w}$ is $\hat{\boldsymbol{x}}_{\boldsymbol{w}} := \Phi_\theta(\hat{\boldsymbol{z}}_{\boldsymbol{w}})$ with $\hat{\boldsymbol{z}}_{\boldsymbol{w}} := \boldsymbol{z}_0 + \gamma\sum_j \beta_j\sigma_j\boldsymbol{u}^{(j)}_{\boldsymbol{w}} \in S_{\boldsymbol{w}}$ the output of Algorithm 1. We bound how far this is from the oracle, first per weight, then as a distance between the two fronts.

The remaining quantities enter as *error term* that vanish in the ideal case. Let $\boldsymbol{z}^o_\parallel := \boldsymbol{z}_0 + \boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{U}_{\boldsymbol{w}}^\top(\boldsymbol{z}^o_{\boldsymbol{w}} - \boldsymbol{z}_0)$ be the projection of the oracle noise vector onto the affine hull of $S_{\boldsymbol{w}}$, and let $G_\perp(\boldsymbol{w}) := \sup_{\boldsymbol{z}\in B'} \|\mathbf{P}^\perp_{\boldsymbol{w}}\nabla H_{\boldsymbol{w}}(\boldsymbol{z})\|_2$ with $B' := \{\boldsymbol{z} : \|\boldsymbol{z}\|_2 \le 4\sqrt{D} + R_{\max}\} \supset B$ be the off-subspace gradient scale, the pointwise counterpart of the spectral tail $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}}\|\mathbf{P}^\perp_{\boldsymbol{w}}\nabla H_{\boldsymbol{w}}(\boldsymbol{z})\|_2^2 = \sum_{j>r}\lambda_j^{(\boldsymbol{w})}$ (Proposition 3).

The first error term is the truncation $\tau_r(\boldsymbol{w}) := G_\perp(\boldsymbol{w})\,\|\mathbf{P}^\perp_{\boldsymbol{w}}(\boldsymbol{z}^o_{\boldsymbol{w}} - \boldsymbol{z}_0)\|_2$. The second is the reach shortfall $\rho(\boldsymbol{w}) := \big(H_{\boldsymbol{w}}(\boldsymbol{z}^o_\parallel) - H_{\boldsymbol{w}}(\hat{\boldsymbol{z}}_{\boldsymbol{w}})\big)_+$, the proxy value the in-subspace target holds beyond what the step collects, whether because the target lies farther than R from $\boldsymbol{z}_0$ or because the step stops short of the slice maximum. The third is the proxy error $\varepsilon_{\text{prox}} := \sup_{\boldsymbol{x}\in\mathcal{X}_\Phi}\|f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})\|_\infty$. Beyond the global smoothness of Assumption 1, the lemma below needs nothing further, and the theorem adds one curvature condition on the true front.

Lemma 1 (Per-weight sub-optimality)**.** *Under Assumption 1, for every $\boldsymbol{w} \in \Delta^{m-1}$ the scalarized-value gap to the oracle satisfies,*

$$0 \;\le\; \boldsymbol{w}^\top f(\boldsymbol{x}^\star_{\boldsymbol{w}}) - \boldsymbol{w}^\top f(\hat{\boldsymbol{x}}_{\boldsymbol{w}}) \;\le\; \tau_r(\boldsymbol{w}) + \rho(\boldsymbol{w}) + 2\,\varepsilon_{\text{prox}}. \tag{23}$$

Proof. The left inequality holds because $\hat{\boldsymbol{x}}_{\boldsymbol{w}} \in \Phi_\theta(S_{\boldsymbol{w}}) \subset \mathcal{X}_\Phi$ and $\boldsymbol{x}^\star_{\boldsymbol{w}}$ maximizes $\boldsymbol{w}^\top f$ over $\mathcal{X}_\Phi$. For the right, split the gap through the proxy. Note, each proxy term is bounded by $\varepsilon_{\text{prox}}$ because, for any $\boldsymbol{x} \in \mathcal{X}_\Phi$,

$$|\boldsymbol{w}^\top(f - \hat{f})(\boldsymbol{x})| \le \|\boldsymbol{w}\|_1\,\|f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})\|_\infty \le \varepsilon_{\text{prox}}$$

by Hölder's inequality and $\|\boldsymbol{w}\|_1 = 1$. We apply it at the oracle point $\boldsymbol{x}^\star_{\boldsymbol{w}}$ and at our point $\hat{\boldsymbol{x}}_{\boldsymbol{w}}$,

$$\boldsymbol{w}^\top f(\boldsymbol{x}^\star_{\boldsymbol{w}}) - \boldsymbol{w}^\top f(\hat{\boldsymbol{x}}_{\boldsymbol{w}}) = \underbrace{\boldsymbol{w}^\top(f - \hat{f})(\boldsymbol{x}^\star_{\boldsymbol{w}})}_{\le\,\varepsilon_{\text{prox}}\ \text{at}\ \boldsymbol{x}^\star_{\boldsymbol{w}}} + \underbrace{\big(\boldsymbol{w}^\top\hat{f}(\boldsymbol{x}^\star_{\boldsymbol{w}}) - \boldsymbol{w}^\top\hat{f}(\hat{\boldsymbol{x}}_{\boldsymbol{w}})\big)}_{(*)} + \underbrace{\boldsymbol{w}^\top(\hat{f} - f)(\hat{\boldsymbol{x}}_{\boldsymbol{w}})}_{\le\,\varepsilon_{\text{prox}}\ \text{at}\ \hat{\boldsymbol{x}}_{\boldsymbol{w}}}.$$

Denoting $H_{\boldsymbol{w}} = \boldsymbol{w}^\top\hat{f}\circ\Phi_\theta$ and using $\boldsymbol{x}^\star_{\boldsymbol{w}} = \Phi_\theta(\boldsymbol{z}^o_{\boldsymbol{w}})$, $\hat{\boldsymbol{x}}_{\boldsymbol{w}} = \Phi_\theta(\hat{\boldsymbol{z}}_{\boldsymbol{w}})$, the middle term is a difference of noise-space objective values,

$$(*) = H_{\boldsymbol{w}}(\boldsymbol{z}^o_{\boldsymbol{w}}) - H_{\boldsymbol{w}}(\hat{\boldsymbol{z}}_{\boldsymbol{w}}).$$

Inserting the projected oracle noise vector $\boldsymbol{z}^o_\parallel$ defined above,

$$(*) = \underbrace{\big[H_{\boldsymbol{w}}(\boldsymbol{z}^o_{\boldsymbol{w}}) - H_{\boldsymbol{w}}(\boldsymbol{z}^o_\parallel)\big]}_{(a)} + \underbrace{\big[H_{\boldsymbol{w}}(\boldsymbol{z}^o_\parallel) - H_{\boldsymbol{w}}(\hat{\boldsymbol{z}}_{\boldsymbol{w}})\big]}_{(b)}.$$

Term (a) is the value lost by moving off the active subspace, term (b) is the value the step leaves on the table inside the subspace. We bound them separately. For (a), let $\boldsymbol{\delta}_\perp := \boldsymbol{z}^o_{\boldsymbol{w}} - \boldsymbol{z}^o_\parallel$. By the definition of $\boldsymbol{z}^o_\parallel$,

$$\boldsymbol{\delta}_\perp = (\boldsymbol{z}^o_{\boldsymbol{w}} - \boldsymbol{z}_0) - \boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{U}_{\boldsymbol{w}}^\top(\boldsymbol{z}^o_{\boldsymbol{w}} - \boldsymbol{z}_0) = \mathbf{P}^\perp_{\boldsymbol{w}}(\boldsymbol{z}^o_{\boldsymbol{w}} - \boldsymbol{z}_0),$$

so $\boldsymbol{\delta}_\perp$ lies in the orthogonal complement of the active subspace and $\mathbf{P}^\perp_{\boldsymbol{w}}\boldsymbol{\delta}_\perp = \boldsymbol{\delta}_\perp$ by idempotence. The segment $\{\boldsymbol{z}^o_\parallel + t\boldsymbol{\delta}_\perp : t \in [0,1]\}$ joins $\boldsymbol{z}^o_\parallel$ to $\boldsymbol{z}^o_{\boldsymbol{w}}$. Both endpoints lie in B', since $\boldsymbol{z}^o_{\boldsymbol{w}} \in B \subset B'$ and $\boldsymbol{z}^o_\parallel = (\boldsymbol{I}_D - \boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{U}_{\boldsymbol{w}}^\top)\boldsymbol{z}_0 + \boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{U}_{\boldsymbol{w}}^\top\boldsymbol{z}^o_{\boldsymbol{w}}$ gives $\|\boldsymbol{z}^o_\parallel\|_2 \le \|\boldsymbol{z}_0\|_2 + \|\boldsymbol{z}^o_{\boldsymbol{w}}\|_2 \le 2\sqrt{D} + (2\sqrt{D} + R_{\max}) = 4\sqrt{D} + R_{\max}$, and B' is convex, so the whole segment lies in B'.

Since $H_{\boldsymbol{w}}$ is C^1 (Assumption 1), the fundamental theorem of calculus along this segment gives,

$$(a) = \int_0^1 \left\langle \nabla H_{\boldsymbol{w}}(\boldsymbol{z}^o_\parallel + t\boldsymbol{\delta}_\perp),\ \boldsymbol{\delta}_\perp \right\rangle dt.$$

Using $\boldsymbol{\delta}_\perp = \mathbf{P}^\perp_{\boldsymbol{w}}\boldsymbol{\delta}_\perp$ and the symmetry of $\mathbf{P}^\perp_{\boldsymbol{w}}$, the integrand is

$$\left\langle \nabla H_{\boldsymbol{w}}(\cdot), \mathbf{P}^\perp_{\boldsymbol{w}}\boldsymbol{\delta}_\perp \right\rangle = \left\langle \mathbf{P}^\perp_{\boldsymbol{w}}\nabla H_{\boldsymbol{w}}(\cdot), \boldsymbol{\delta}_\perp \right\rangle \le \|\mathbf{P}^\perp_{\boldsymbol{w}}\nabla H_{\boldsymbol{w}}(\cdot)\|_2 \, \|\boldsymbol{\delta}_\perp\|_2 \le G_\perp(\boldsymbol{w}) \, \|\boldsymbol{\delta}_\perp\|_2,$$

by Cauchy–Schwarz and the definition of $G_\perp$, the point $\boldsymbol{z}^o_\parallel + t\boldsymbol{\delta}_\perp$ being in B'.
Integrating over $t \in [0,1]$, we get,

$$(a) \le G_\perp(\boldsymbol{w}) \, \|\boldsymbol{\delta}_\perp\|_2 = G_\perp(\boldsymbol{w}) \, \|\mathbf{P}^\perp_{\boldsymbol{w}}(\boldsymbol{z}^o_{\boldsymbol{w}} - \boldsymbol{z}_0)\|_2 = \tau_r(\boldsymbol{w}).$$

Similarly, for (b), by the definition of ρ,

$$\rho(\boldsymbol{w}) = \big(H_{\boldsymbol{w}}(\boldsymbol{z}^o_\parallel) - H_{\boldsymbol{w}}(\hat{\boldsymbol{z}}_{\boldsymbol{w}})\big)_+ \ \ge \ H_{\boldsymbol{w}}(\boldsymbol{z}^o_\parallel) - H_{\boldsymbol{w}}(\hat{\boldsymbol{z}}_{\boldsymbol{w}}) = (b).$$

Combining, $(*) = (a) + (b) \le \tau_r(\boldsymbol{w}) + \rho(\boldsymbol{w})$, and substituting into the proxy split above gives

$$\boldsymbol{w}^\top f(\boldsymbol{x}^\star_{\boldsymbol{w}}) - \boldsymbol{w}^\top f(\hat{\boldsymbol{x}}_{\boldsymbol{w}}) \le \varepsilon_{\text{prox}} + \tau_r(\boldsymbol{w}) + \rho(\boldsymbol{w}) + \varepsilon_{\text{prox}},$$

which is Eq. (23). □

The lemma is an error decomposition. It does not bound $\rho(\boldsymbol{w})$, which measures what a single steering step leaves inside the subspace, and $\tau_r(\boldsymbol{w})$ is related to the spectral tail only through its pointwise counterpart $G_\perp$.

From value to distance. Lemma 1 bounds the gap in *scalarized value*, but the theorem is about *distance* between fronts, and a value bound alone does not give one. If the front contains a flat face orthogonal to $\boldsymbol{w}$, every point on that face has the same scalarized value as the maximizer, so the value gap is zero while the distance can be as large as the face. Concretely, for two objectives with front the segment from $(1,0)$ to $(0,1)$ and $\boldsymbol{w} = (\frac{1}{2}, \frac{1}{2})$, Lemma 1 is satisfied with $\epsilon = 0$ by every point of the segment. To convert value into distance we therefore need the front to gain value as one moves away from the maximizer, which is a condition on the attainable set $\mathcal{Y} := f(\mathcal{X}_\Phi)$ in objective space, unrelated to anything in noise space. Convexity itself is not assumed, since $\mathcal{F}^\star_\Phi$ is by definition the part of the front reached by linear scalarization, so what remains is curvature. Let's say $\boldsymbol{y}^\star_{\boldsymbol{w}} := f(\boldsymbol{x}^\star_{\boldsymbol{w}})$. Since the scalarization is linear, optimality of $\boldsymbol{y}^\star_{\boldsymbol{w}}$ is exactly the ***variational inequality***,

$$\boldsymbol{w}^\top(\boldsymbol{y}^\star_{\boldsymbol{w}} - \boldsymbol{y}) \ge 0 \quad \text{for all } \boldsymbol{y} \in \mathcal{Y},$$

which is the statement that fails to separate points on a flat face. We strengthen it by a quadratic term,

$$\boldsymbol{w}^\top\big(\boldsymbol{y}^\star_{\boldsymbol{w}} - \boldsymbol{y}\big) \ \ge \ \tfrac{\mu}{2}\,\|\boldsymbol{y}^\star_{\boldsymbol{w}} - \boldsymbol{y}\|_2^2 \qquad \text{for all } \boldsymbol{y} \in \mathcal{Y},\ \boldsymbol{w} \in \Delta^{m-1}, \tag{24}$$

so that a point ϵ-close in value is $\sqrt{2\epsilon/\mu}$-close in position. Because the objective $\boldsymbol{w}^\top\boldsymbol{y}$ is linear, the quadratic term cannot come from the objective, as it would in a strongly monotone variational inequality. It is a property of the set, namely that $\mathcal{Y}$ must bend away from each supporting hyperplane at rate μ, and a sufficient condition is that $\mathcal{Y}$ be μ-strongly convex (Vial, 1983).

Thus, as the example above shows, the condition fails on any flat face of the front. We do not claim these are the only failures, but flat faces are the regime in which linear scalarization is already

uninformative, so we expect the assumption to exclude little that a weight sweep could recover. The condition also makes $\boldsymbol{y}^\star_{\boldsymbol{w}}$ unique, because a second maximizer $\boldsymbol{y}$ would have zero on the left of Eq. (24) and $\frac{\mu}{2}\|\boldsymbol{y}^\star_{\boldsymbol{w}} - \boldsymbol{y}\|_2^2 > 0$ on the right, so the Lipschitz dependence of $\boldsymbol{y}^\star_{\boldsymbol{w}}$ on $\boldsymbol{w}$ assumed below is meaningful. Without Eq. (24) the theorem below still holds in its first form, as an ϵ-approximate front. With it, the per-weight bound lifts to a bound between the two fronts.

Theorem 3 (Front recovery up to the error terms). *Suppose Eq. (24) holds and $\boldsymbol{w} \mapsto f(\boldsymbol{x}^\star_{\boldsymbol{w}})$ is $L_{\mathcal{P}}$-Lipschitz on Δ^{m-1}. Let $\epsilon := \max_{\boldsymbol{w}} \big[\tau_r(\boldsymbol{w}) + \rho(\boldsymbol{w})\big] + 2\varepsilon_{\text{prox}}$. Sweeping $\boldsymbol{w}$ over a grid $W \subset \Delta^{m-1}$ of mesh η_W, the produced front $\hat{\mathcal{F}} = \{f(\hat{\boldsymbol{x}}_{\boldsymbol{w}}) : \boldsymbol{w} \in W\}$ is an ϵ-approximate Pareto front, every produced point being within ϵ in scalarized value of $\mathcal{F}^\star_\Phi$, and the Hausdorff distance satisfies*

$$d_H(\hat{\mathcal{F}}, \mathcal{F}^\star_\Phi) \le \sqrt{2\epsilon/\mu} + L_{\mathcal{P}}\,\eta_W. \tag{25}$$

In particular, when the proxy is exact ($\varepsilon_{\text{prox}} = 0$), steering is full-rank ($\tau_r \equiv 0$), the step collects the in-subspace target ($\rho \equiv 0$), and the grid is dense ($\eta_W \to 0$), $d_H(\hat{\mathcal{F}}, \mathcal{F}^\star_\Phi) = 0$.

Proof. Let's first fix $\boldsymbol{w} \in W$. By Lemma 1 and the definition of ϵ,

$$0 \;\le\; \boldsymbol{w}^\top f(\boldsymbol{x}^\star_{\boldsymbol{w}}) - \boldsymbol{w}^\top f(\hat{\boldsymbol{x}}_{\boldsymbol{w}}) \;\le\; \tau_r(\boldsymbol{w}) + \rho(\boldsymbol{w}) + 2\varepsilon_{\text{prox}} \;\le\; \epsilon,$$

so every produced solution is within ϵ in scalarized value of $\mathcal{F}^\star_\Phi$, which is the first claim. For the distance, note that $\hat{\boldsymbol{x}}_{\boldsymbol{w}} \in \mathcal{X}_\Phi$, so $\boldsymbol{y} = f(\hat{\boldsymbol{x}}_{\boldsymbol{w}}) \in \mathcal{Y}$ and Eq. (24) applies to it,

$$\tfrac{\mu}{2}\,\|f(\boldsymbol{x}^\star_{\boldsymbol{w}}) - f(\hat{\boldsymbol{x}}_{\boldsymbol{w}})\|_2^2 \;\le\; \boldsymbol{w}^\top\big(f(\boldsymbol{x}^\star_{\boldsymbol{w}}) - f(\hat{\boldsymbol{x}}_{\boldsymbol{w}})\big) \;\le\; \epsilon.$$

Rearranging, every produced point is within $\sqrt{2\epsilon/\mu}$ of an oracle point as,

$$\|f(\hat{\boldsymbol{x}}_{\boldsymbol{w}}) - f(\boldsymbol{x}^\star_{\boldsymbol{w}})\|_2 \;\le\; \sqrt{2\epsilon/\mu} \qquad \text{for all } \boldsymbol{w} \in W.$$

This bounds one side of the Hausdorff distance, $\sup_{\hat{\boldsymbol{y}} \in \hat{\mathcal{F}}} \operatorname{dist}(\hat{\boldsymbol{y}}, \mathcal{F}^\star_\Phi) \le \sqrt{2\epsilon/\mu}$.

Now, for the other side, fix any $\boldsymbol{w} \in \Delta^{m-1}$ and let $\boldsymbol{w}' \in W$ be a grid weight with $\|\boldsymbol{w} - \boldsymbol{w}'\| \le \eta_W$. We compare the oracle point at $\boldsymbol{w}$ to the produced point at $\boldsymbol{w}'$ through the oracle point at $\boldsymbol{w}'$. By the triangle inequality,

$$\|f(\boldsymbol{x}^\star_{\boldsymbol{w}}) - f(\hat{\boldsymbol{x}}_{\boldsymbol{w}'})\|_2 \;\le\; \underbrace{\|f(\boldsymbol{x}^\star_{\boldsymbol{w}}) - f(\boldsymbol{x}^\star_{\boldsymbol{w}'})\|_2}_{(a)} + \underbrace{\|f(\boldsymbol{x}^\star_{\boldsymbol{w}'}) - f(\hat{\boldsymbol{x}}_{\boldsymbol{w}'})\|_2}_{(b)}.$$

Term (a) is the movement of the oracle front between the two weights, and the Lipschitz assumption on $\boldsymbol{w} \mapsto f(\boldsymbol{x}^\star_{\boldsymbol{w}})$ gives,

$$(a) \;\le\; L_{\mathcal{P}}\,\|\boldsymbol{w} - \boldsymbol{w}'\| \;\le\; L_{\mathcal{P}}\,\eta_W.$$

Whereas, Term (b) is the per-weight distance at the grid weight $\boldsymbol{w}'$, which we just bounded above,

$$(b) \;\le\; \sqrt{2\epsilon/\mu}.$$

Combining both terms,

$$\|f(\boldsymbol{x}^\star_{\boldsymbol{w}}) - f(\hat{\boldsymbol{x}}_{\boldsymbol{w}'})\|_2 \;\le\; L_{\mathcal{P}}\,\eta_W + \sqrt{2\epsilon/\mu},$$

Therefore, $\sup_{\boldsymbol{y} \in \mathcal{F}^\star_\Phi} \operatorname{dist}(\boldsymbol{y}, \hat{\mathcal{F}}) \le \sqrt{2\epsilon/\mu} + L_{\mathcal{P}}\eta_W$. □

Scalarization agnostic. *The convex front is a property of linear scalarization, not of the method.* Stated for linear weights, the theorem recovers the part of the front reached by supporting hyperplanes, which is classical and holds for any linear-weight sweep. Steering itself works with any scalarization. Theorem 1 never uses linearity in $\boldsymbol{w}$, so it applies to any smooth scalarization. Non-smooth ones such as Tchebycheff are handled by the same smoothing argument used for the proxies. The softmin Tchebycheff of Eq. (14) is smooth and lies within $\nu \log m$ of the Tchebycheff value everywhere, so Theorem 1 carries over up to that error. Empirically, all four scalarizations of Appendix B.2.6 ascend and reach the same front quality. The precomputation also carries over unchanged, because the cached per-objective gradients $\{\nabla\phi_a(\boldsymbol{z}_i)\}$ still suffice to form $\boldsymbol{M}_{\boldsymbol{w}}$ at any weight with no new decodes. Extending Theorem 3 beyond linear scalarization, where it would also cover non-convex parts of the front, is left to future work.

The one thing that changes is the cost of assembling $\boldsymbol{M}_{\boldsymbol{w}}$. Under linear scalarization the gradient $\nabla H_{\boldsymbol{w}} = \sum_a w_a \nabla g_a$ is linear in $\boldsymbol{w}$, so $\boldsymbol{M}_{\boldsymbol{w}}$ recombines from the $m(m+1)/2$ cached blocks by the bilinear rule $\sum_{a,b} w_a w_b \boldsymbol{C}_{ab}$ (Eq. (8)), costing $O(m^2 D^2)$ per weight independent of n. Under Tchebycheff the gradient is not linear in $\boldsymbol{w}$, so the blocks no longer recombine and $\boldsymbol{M}_{\boldsymbol{w}}$ is rebuilt as the sample average of n rank-one outer products, costing $O(nD^2)$ per weight. Since typically $n \gg m^2$, the trade-off between the two scalarizations is efficiency, not correctness.

The proxy error $\varepsilon_{\text{prox}}$ is the offline barrier, since it is the only quantity tying the recovered front to the true one, and it cannot be driven to zero without querying f. The reach shortfall $\rho(\boldsymbol{w})$ is the operators' entry point, since it is positive where the in-subspace target lies beyond the reach R or where the step stops short of it, which is the regime the guidance operators of § 4 are built to reduce.

C.6 Guided Noise-Space RFM Steering

Noise-space-only steering saturates in the two places identified above, where the mean gradient leaves the active subspace and where the per-$\boldsymbol{w}$ optimum lies beyond the reach R. There we add a guidance term $\eta(t)\,\mathcal{G}_{\boldsymbol{w}}$ to the velocity for $t \geq t_{\text{start}}$, using one proxy gradient per guided ODE step and never differentiating through the ODE solve (Algorithms 2 and 3). Gradients are taken with respect to $\boldsymbol{x}_t$, through the denoised estimate $\hat{\boldsymbol{x}}_1(\boldsymbol{x}_t) = \boldsymbol{x}_t + (1-t)\boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$ for RFM-DAMG and RFM-Cone and directly for RFM-APG. Below, $\hat{f}_a$, $h_{\boldsymbol{w}} = \boldsymbol{w}^\top \hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{J}} = [\nabla \hat{f}_1 \cdots \nabla \hat{f}_m]$ denote the proxies and their Jacobian as functions of $\boldsymbol{x}_t$, so the results guarantee first-order improvement of the proxies as $\boldsymbol{x}_t$ moves, not of the final decoded sample.

The operators fall into the two families of Section 4. The data-adaptive operators RFM-APG and RFM-DAMG precondition the scalarized gradient, and the Pareto-aware RFM-Cone combines the per-objective gradients into a common ascent direction.

Algorithm 2 Data-adaptive guidance (RFM-APG and RFM-DAMG)

Require: Steered noise $\boldsymbol{z}_{\boldsymbol{w}}$ (Algorithm 1), flow $\boldsymbol{v}_\theta$, proxies $\hat{\boldsymbol{f}}$, weight $\boldsymbol{w}$, onset t_{start}, strength $\eta_{\max}$ (APG) or κ (DAMG), cached $\boldsymbol{U}_{\boldsymbol{w}}, \boldsymbol{\beta}$ (DAMG), tangent onset t_T, neighbors k, rank floor τ_{pca} (DAMG)
Ensure: Guided sample $\hat{\boldsymbol{x}}_{\boldsymbol{w}}$

$\boldsymbol{x}_0 \leftarrow \boldsymbol{z}_{\boldsymbol{w}}$
if RFM-APG **then** $\tilde{\boldsymbol{U}}, \tilde{\boldsymbol{\beta}} \leftarrow$ Algorithm 1 on $\boldsymbol{x}_{t_{\text{start}}}$ in place of $\boldsymbol{z}$ ▷ once per task
for each ODE step t **do**
  $\boldsymbol{v} \leftarrow \boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$
  if $t < t_{\text{start}}$ **then**
    $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t + \Delta t\, \boldsymbol{v}$
  else if RFM-APG **then**
    $\mathcal{G}_{\boldsymbol{w}} \leftarrow \tilde{\boldsymbol{U}}\,\mathrm{diag}(\tilde{\boldsymbol{\beta}})\tilde{\boldsymbol{U}}^\top \nabla_{\boldsymbol{x}_t} h_{\boldsymbol{w}}(\boldsymbol{x}_t)$ ▷ projected or adaptive
    $\eta(t) \leftarrow \eta_{\max}(t - t_{\text{start}})/(1 - t_{\text{start}})$
    $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t + \Delta t\,(\boldsymbol{v} + \eta(t)\,\mathcal{G}_{\boldsymbol{w}})$
  else ▷ RFM-DAMG
    $\mathcal{G}_{\boldsymbol{w}} \leftarrow \mathbf{R}_{\boldsymbol{w}} \nabla_{\boldsymbol{x}_t} h_{\boldsymbol{w}}(\hat{\boldsymbol{x}}_1(\boldsymbol{x}_t)),\ \mathbf{R}_{\boldsymbol{w}} = \boldsymbol{U}_{\boldsymbol{w}}\,\mathrm{diag}(\boldsymbol{\beta})\boldsymbol{U}_{\boldsymbol{w}}^\top$ ▷ cached spectrum
    if $t \geq t_T$ and the local rank is certified **then**
      $T \leftarrow$ top local PCA directions of the k nearest samples in the batch
      $\mathcal{G}_{\boldsymbol{w}} \leftarrow \mathbf{P}_T\,\mathcal{G}_{\boldsymbol{w}}$ ▷ manifold
    end if
    $\eta(t) \leftarrow \min\big(\kappa\|\boldsymbol{v}\|/(\|\mathcal{G}_{\boldsymbol{w}}\| + \varepsilon),\ \eta_{\max}\big)$
    $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t + \Delta t\,(\boldsymbol{v} + \eta(t)\,\mathcal{G}_{\boldsymbol{w}})$
  end if
end for
return $\hat{\boldsymbol{x}}_{\boldsymbol{w}} \leftarrow \boldsymbol{x}_1$

C.6.1 Preconditioned ascent: adaptive, projected, manifold

Each operator forms a refinement direction by applying a *data-adaptive* preconditioner Q to the proxy gradient, where Q is read from structure the data reveals, namely the AGOP spectrum for the projected and adaptive preconditioners and the local data covariance for the manifold preconditioner. RFM-APG uses the projected or adaptive preconditioner on the active subspace rebuilt at t_{start}, and RFM-DAMG applies the manifold preconditioner after the adaptive one (Algorithm 2). Ascent

holds for any $Q \succeq 0$ and the data shapes the step, giving stronger and better-directed guidance where structure exists and reducing to plain ascent where it does not.

Proposition 4 (**Preconditioned ascent**). *Let $Q \succeq 0$. The direction $Q\,\nabla h_{\boldsymbol{w}}(\boldsymbol{x})$ is an ascent direction for $h_{\boldsymbol{w}}$,*

$$\big\langle \nabla h_{\boldsymbol{w}}(\boldsymbol{x}),\, Q\,\nabla h_{\boldsymbol{w}}(\boldsymbol{x}) \big\rangle = \|\nabla h_{\boldsymbol{w}}(\boldsymbol{x})\|_Q^2 \geq 0, \tag{26}$$

strictly positive unless $\nabla h_{\boldsymbol{w}}(\boldsymbol{x}) \perp \mathrm{range}(Q)$. The three preconditioners are symmetric positive semidefinite. The projected *one is $Q = \boldsymbol{U}\boldsymbol{U}^\top$, the* adaptive *one $Q = \mathbf{R}_{\boldsymbol{w}} := \boldsymbol{U}_{\boldsymbol{w}}\,\mathrm{diag}(\boldsymbol{\beta})\boldsymbol{U}_{\boldsymbol{w}}^\top$ with $\beta_j = (\lambda_j^{(\boldsymbol{w})}/\lambda_1^{(\boldsymbol{w})})^\alpha$ reduces to the projected one on the retained directions at $\alpha = 0$, and the* manifold *one $Q = \mathbf{P}_T$ projects onto a locally estimated tangent space T of the data manifold.*

Proof. Since $Q \succeq 0$, $\langle \boldsymbol{g}, Q\boldsymbol{g}\rangle = \boldsymbol{g}^\top Q\boldsymbol{g} \geq 0$ for every $\boldsymbol{g}$, with equality iff $Q\boldsymbol{g} = \mathbf{0}$, i.e. $\boldsymbol{g} \perp \mathrm{range}(Q)$; apply this with $\boldsymbol{g} = \nabla h_{\boldsymbol{w}}(\boldsymbol{x})$. The directional derivative of $h_{\boldsymbol{w}}$ along $Q\nabla h_{\boldsymbol{w}}$ equals the left side of Eq. (26), so a small step ascends. Each listed Q is a Gram or projection matrix, hence positive semi-definite: $\boldsymbol{U}_{\boldsymbol{w}}\,\mathrm{diag}(\boldsymbol{\beta})\boldsymbol{U}_{\boldsymbol{w}}^\top$ has eigenvalues $\beta_j \geq 0$, and $\boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{U}_{\boldsymbol{w}}^\top$ and $\mathbf{P}_T$ are orthogonal projectors. □

Proposition 5 (**Adaptive weighting concentrates the step by data-learned relevance**). *For the adaptive operator, the ascent gain splits over the data-learned directions,*

$$\big\langle \boldsymbol{g},\, \mathbf{R}_{\boldsymbol{w}}\boldsymbol{g} \big\rangle = \sum_j \beta_j \left(\boldsymbol{u}_{\boldsymbol{w}}^{(j)\top}\boldsymbol{g}\right)^2, \qquad \boldsymbol{g} = \nabla h_{\boldsymbol{w}}(\boldsymbol{x}), \tag{27}$$

so the step places weight β_j on direction $\boldsymbol{u}_{\boldsymbol{w}}^{(j)}$. Because $\lambda_j^{(\boldsymbol{w})}$ is the objective's mean squared sensitivity along $\boldsymbol{u}_{\boldsymbol{w}}^{(j)}$ (Proposition 1), the weights $\beta_j = (\lambda_j^{(\boldsymbol{w})}/\lambda_1^{(\boldsymbol{w})})^\alpha$ concentrate the step on the directions the objective most responds to, and this concentration increases monotonically with α: $\partial_\alpha \beta_j = \beta_j \log(\lambda_j^{(\boldsymbol{w})}/\lambda_1^{(\boldsymbol{w})}) \leq 0$ for every $\lambda_j^{(\boldsymbol{w})} \leq \lambda_1^{(\boldsymbol{w})}$. When the retained spectrum is flat the weights are uniform and the operator reduces to plain projected ascent, and does not reweight directions the data does not distinguish. The weighting is thus strong exactly when the AGOP spectrum is peaked, that is when the data reveals a few dominant objective-sensitive directions.

Proof. Let's denote $\boldsymbol{g} := \nabla h_{\boldsymbol{w}}(\boldsymbol{x})$. Expanding the adaptive operator,

$$\mathbf{R}_{\boldsymbol{w}}\boldsymbol{g} = \sum_j \beta_j\, \boldsymbol{u}_{\boldsymbol{w}}^{(j)}\boldsymbol{u}_{\boldsymbol{w}}^{(j)\top}\boldsymbol{g} = \sum_j \beta_j \left(\boldsymbol{u}_{\boldsymbol{w}}^{(j)\top}\boldsymbol{g}\right)\boldsymbol{u}_{\boldsymbol{w}}^{(j)},$$

so taking the inner product with $\boldsymbol{g}$ and using $\boldsymbol{g}^\top\boldsymbol{u}_{\boldsymbol{w}}^{(j)} = \boldsymbol{u}_{\boldsymbol{w}}^{(j)\top}\boldsymbol{g}$,

$$\big\langle \boldsymbol{g}, \mathbf{R}_{\boldsymbol{w}}\boldsymbol{g} \big\rangle = \sum_j \beta_j \left(\boldsymbol{u}_{\boldsymbol{w}}^{(j)\top}\boldsymbol{g}\right)\left(\boldsymbol{g}^\top\boldsymbol{u}_{\boldsymbol{w}}^{(j)}\right) = \sum_j \beta_j \left(\boldsymbol{u}_{\boldsymbol{w}}^{(j)\top}\boldsymbol{g}\right)^2,$$

which is Eq. (27). Next, to prove that the weights are monotone in α, if we write the weight as an exponential,

$$\beta_j = \left(\frac{\lambda_j^{(\boldsymbol{w})}}{\lambda_1^{(\boldsymbol{w})}}\right)^\alpha = \exp\!\Big(\alpha \log \frac{\lambda_j^{(\boldsymbol{w})}}{\lambda_1^{(\boldsymbol{w})}}\Big),$$

and differentiate in α,

$$\partial_\alpha \beta_j = \exp\!\Big(\alpha \log \frac{\lambda_j^{(\boldsymbol{w})}}{\lambda_1^{(\boldsymbol{w})}}\Big) \cdot \log \frac{\lambda_j^{(\boldsymbol{w})}}{\lambda_1^{(\boldsymbol{w})}} = \beta_j \log \frac{\lambda_j^{(\boldsymbol{w})}}{\lambda_1^{(\boldsymbol{w})}}.$$

Since $\beta_j > 0$ and $\lambda_j^{(\boldsymbol{w})} \leq \lambda_1^{(\boldsymbol{w})}$ gives $\log(\lambda_j^{(\boldsymbol{w})}/\lambda_1^{(\boldsymbol{w})}) \leq 0$, we get $\partial_\alpha \beta_j \leq 0$, with equality only for $j = 1$, therefore, every weight other than the leading one decreases as α grows.

Finally, if the retained spectrum is flat, $\lambda_j^{(\boldsymbol{w})} = \lambda_1^{(\boldsymbol{w})}$ for all retained j, then $\beta_j = 1^\alpha = 1$ for every retained j regardless of α, and

$$\mathbf{R}_{\boldsymbol{w}} = \sum_{j \leq r} \boldsymbol{u}_{\boldsymbol{w}}^{(j)}\boldsymbol{u}_{\boldsymbol{w}}^{(j)\top} = \boldsymbol{U}_{\boldsymbol{w}}\boldsymbol{U}_{\boldsymbol{w}}^\top,$$

which is the projected operator. □

RFM-DAMG applies $\mathbf{P}_T$ after $\mathbf{R}_{\boldsymbol{w}}$, where T is estimated by local PCA over the k nearest samples in the batch. The local rank r is the median over samples of the number of singular values at least τ_{pca} times the largest, and it is *certified* when $1 \le r < \min(k, D)$ and $k \ge r + 2$, so the neighborhoods are genuinely lower-dimensional and hold enough points to estimate T. The product $\mathbf{P}_T\mathbf{R}_{\boldsymbol{w}}$ is not symmetric in general, so Proposition 4 applies to each factor but not to the product, which ascends whenever $\langle \boldsymbol{g}, \mathbf{P}_T\mathbf{R}_{\boldsymbol{w}}\boldsymbol{g}\rangle \ge 0$. Without certification the operator reduces to $\mathbf{R}_{\boldsymbol{w}}\boldsymbol{g}$ and the proposition applies directly. The proposition also does not depend on where Q comes from, so it holds even though $\mathbf{R}_{\boldsymbol{w}}$ is built from noise-space geometry and applied to a design-space gradient.

Algorithm 3 Pareto-aware guidance (RFM-Cone)

Require: Steered noise $\boldsymbol{z}_{\boldsymbol{w}}$ (Algorithm 1), flow $\boldsymbol{v}_\theta$, proxies $\hat{\boldsymbol{f}}$, weight $\boldsymbol{w}$, onset t_{start}, strength κ
Ensure: Guided sample $\hat{\boldsymbol{x}}_{\boldsymbol{w}}$

$\boldsymbol{x}_0 \leftarrow \boldsymbol{z}_{\boldsymbol{w}}$
for each ODE step t **do**
  $\boldsymbol{v} \leftarrow \boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$
  if $t < t_{\mathrm{start}}$ **then**
    $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t + \Delta t\, \boldsymbol{v}$
  else
    $\widehat{\boldsymbol{J}} \leftarrow [\nabla_{\boldsymbol{x}_t} \hat{f}_a(\hat{\boldsymbol{x}}_1(\boldsymbol{x}_t))]_{a=1}^m$, normalize its columns to get $\widehat{\boldsymbol{J}}_u$
    $\boldsymbol{\pi} \leftarrow$ proxy scores at $\hat{\boldsymbol{x}}_1$ shifted by their minimum and normalized to sum to one
    $\boldsymbol{u}_{\mathrm{pref}} \leftarrow \widehat{\boldsymbol{J}}_u(2\boldsymbol{w} - \boldsymbol{\pi})$ ▷ pulls hardest on lagging objectives
    $\boldsymbol{d} \leftarrow \Pi_{\mathcal{K}}(\boldsymbol{u}_{\mathrm{pref}})$ with $\mathcal{K} = \{\boldsymbol{s} \mid \widehat{\boldsymbol{J}}_u^\top \boldsymbol{s} \ge \boldsymbol{0}\}$ ▷ exact active-set projection
    if $\boldsymbol{d} \approx \boldsymbol{0}$ **then**
      $\boldsymbol{d} \leftarrow$ min-norm point of the convex hull of the columns of $\widehat{\boldsymbol{J}}_u$ ▷ MGDA fallback
    end if
    $\eta(t) \leftarrow \kappa\|\boldsymbol{v}\|/(\|\boldsymbol{d}\| + \varepsilon)$
    $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t + \Delta t\,(\boldsymbol{v} + \eta(t)\,\boldsymbol{d})$
  end if
end for
return $\hat{\boldsymbol{x}}_{\boldsymbol{w}} \leftarrow \boldsymbol{x}_1$

Theorem 2 (Cone projection is a per-objective Pareto improvement). *Let $\mathcal{K}(\boldsymbol{x}) := \{\boldsymbol{s} : \langle \nabla \hat{f}_a(\boldsymbol{x}), \boldsymbol{s}\rangle \ge 0\ \forall a\}$ be the common-ascent cone, and let $\boldsymbol{d}_{\mathcal{K}} := \Pi_{\mathcal{K}(\boldsymbol{x})}(\boldsymbol{u}_{\mathrm{pref}})$ be the Euclidean projection of a preference direction $\boldsymbol{u}_{\mathrm{pref}}$ onto $\mathcal{K}(\boldsymbol{x})$. The cone operator takes the step direction $\boldsymbol{d} = \boldsymbol{d}_{\mathcal{K}}$ if $\boldsymbol{d}_{\mathcal{K}} \neq \boldsymbol{0}$, and otherwise the min-norm direction $\boldsymbol{d} = \arg\min\{\|\boldsymbol{v}\|_2 : \boldsymbol{v} \in \mathrm{conv}\{\nabla \hat{f}_a(\boldsymbol{x})\}\}$ of MGDA (Désidéri, 2012). Then:*

(i) (First-order improvement) $\boldsymbol{d} \in \mathcal{K}(\boldsymbol{x})$, *so $\langle \nabla \hat{f}_a(\boldsymbol{x}), \boldsymbol{d}\rangle \ge 0$ for all a. Let each $\hat{f}_a$ be L-smooth and have $c_a := \langle \nabla \hat{f}_a(\boldsymbol{x}), \boldsymbol{d}\rangle$. An objective with $c_a > 0$ improves exactly, $\hat{f}_a(\boldsymbol{x} + \bar{\eta}\boldsymbol{d}) \ge \hat{f}_a(\boldsymbol{x})$, for every $0 < \bar{\eta} \le 2c_a/(L\|\boldsymbol{d}\|_2^2)$. An objective with $c_a = 0$, one whose constraint is tight after projection, is flat to first order and loses at most $\frac{1}{2}L\bar{\eta}^2\|\boldsymbol{d}\|_2^2$. Here $\bar{\eta}$ is the displacement per guided ODE step, $\bar{\eta} = \Delta t\,\eta(t)$ in Algorithm 3.*

(ii) (Existence) $\mathcal{K}(\boldsymbol{x})$ *contains a strictly ascending direction if and only if $\boldsymbol{0} \notin \mathrm{conv}\{\nabla \hat{f}_a(\boldsymbol{x})\}$, i.e. $\boldsymbol{x}$ is not Pareto-stationary.*

(iii) (Graceful stop) *The min-norm direction is $\boldsymbol{0}$ if and only if $\boldsymbol{x}$ is Pareto-stationary. Hence when the projection vanishes at a Pareto-stationary point, $\boldsymbol{d} = \boldsymbol{0}$, the guidance term is switched off, and the decode continues with $\boldsymbol{v}_\theta$ alone.*

Proof. (i) The cone $\mathcal{K}(\boldsymbol{x})$ is an intersection of the closed half-spaces $\{\boldsymbol{s} : \langle \nabla \hat{f}_a(\boldsymbol{x}), \boldsymbol{s}\rangle \ge 0\}$, each containing the origin, so it is a closed convex cone. The Euclidean projection onto a closed convex set lands in that set, hence $\boldsymbol{d}_{\mathcal{K}} \in \mathcal{K}(\boldsymbol{x})$. For the fallback, let $\boldsymbol{d}_{\mathrm{mn}}$ be the min-norm element of the compact convex set $\mathrm{conv}\{\nabla \hat{f}_a(\boldsymbol{x})\}$. Optimality of $\boldsymbol{d}_{\mathrm{mn}}$ over that set gives, for every $\boldsymbol{v}$ in it,

$$\langle \boldsymbol{d}_{\mathrm{mn}},\ \boldsymbol{v} - \boldsymbol{d}_{\mathrm{mn}}\rangle \ \ge\ 0 \qquad \Longrightarrow \qquad \langle \boldsymbol{d}_{\mathrm{mn}}, \boldsymbol{v}\rangle \ \ge\ \|\boldsymbol{d}_{\mathrm{mn}}\|_2^2 \ \ge\ 0,$$

and taking $\boldsymbol{v} = \nabla \hat{f}_a(\boldsymbol{x})$ shows $\boldsymbol{d}_{\mathrm{mn}} \in \mathcal{K}(\boldsymbol{x})$. In either branch $\boldsymbol{d} \in \mathcal{K}(\boldsymbol{x})$, i.e. $c_a \ge 0$ for all a.

Now, if we let each $\hat{f}_a$ be L-smooth. By the descent lemma along the direction $\boldsymbol{d}$,

$$\hat{f}_a(\boldsymbol{x} + \bar{\eta}\boldsymbol{d}) \;\geq\; \hat{f}_a(\boldsymbol{x}) + \bar{\eta}\, c_a - \tfrac{L}{2}\,\bar{\eta}^2 \|\boldsymbol{d}\|_2^2.$$

If $c_a > 0$, the right side is at least $\hat{f}_a(\boldsymbol{x})$ whenever $\bar{\eta}\, c_a \geq \frac{L}{2}\bar{\eta}^2\|\boldsymbol{d}\|_2^2$, i.e. whenever $\bar{\eta} \leq 2c_a/(L\|\boldsymbol{d}\|_2^2)$, which is the exact improvement. If $c_a = 0$, the linear term vanishes and the display reduces to $\hat{f}_a(\boldsymbol{x} + \bar{\eta}\boldsymbol{d}) \geq \hat{f}_a(\boldsymbol{x}) - \frac{L}{2}\bar{\eta}^2\|\boldsymbol{d}\|_2^2$, so the objective is flat to first order and loses at most a second-order amount.

Therefore, Part (i) holds because of how the operator is built, not because of guidance in general. The step is projected onto $\mathcal{K}(\boldsymbol{x})$, and every direction in $\mathcal{K}(\boldsymbol{x})$ has non-negative directional derivative for every objective by definition, so no objective can be traded for another to first order. Also, the same argument holds if the gradients are normalized to unit length, since this rescales each constraint of $\mathcal{K}(\boldsymbol{x})$ by a positive factor and leaves the cone and the Pareto-stationarity condition unchanged.

(ii) We need to show that a direction along which every objective strictly increases exists exactly when the gradients cannot be balanced against each other, that is, when no convex combination of them is zero. The two directions of the equivalence are the two arguments below.

Suppose first that $\mathbf{0} \in \operatorname{conv}\{\nabla\hat{f}_a(\boldsymbol{x})\}$, so $\mathbf{0} = \sum_a \lambda_a \nabla\hat{f}_a(\boldsymbol{x})$ for some $\boldsymbol{\lambda} \in \Delta^{m-1}$. Then for any $\boldsymbol{s}$,

$$\sum_a \lambda_a \,\langle \nabla\hat{f}_a(\boldsymbol{x}), \boldsymbol{s}\rangle = \Big\langle \sum_a \lambda_a \nabla\hat{f}_a(\boldsymbol{x}),\; \boldsymbol{s}\Big\rangle = \langle \mathbf{0}, \boldsymbol{s}\rangle = 0,$$

so the inner products cannot all be strictly positive and no strictly ascending direction exists. Conversely, suppose $\mathbf{0} \notin \operatorname{conv}\{\nabla\hat{f}_a(\boldsymbol{x})\}$. This set is compact and convex, so by the separating hyperplane theorem there is $\boldsymbol{s}$ with $\langle \boldsymbol{v}, \boldsymbol{s}\rangle > 0$ for every $\boldsymbol{v}$ in it, in particular $\langle \nabla\hat{f}_a(\boldsymbol{x}), \boldsymbol{s}\rangle > 0$ for all a, and $\boldsymbol{s} \in \mathcal{K}(\boldsymbol{x})$ is strictly ascending.

(iii) We need to show that the fallback direction is zero exactly at Pareto-stationary points, so that the operator does nothing there. Recall $\boldsymbol{d}_{\mathrm{mn}}$ is the min-norm element of $\operatorname{conv}\{\nabla\hat{f}_a(\boldsymbol{x})\}$, so

$$\|\boldsymbol{d}_{\mathrm{mn}}\|_2 = \min\big\{\|\boldsymbol{v}\|_2 : \boldsymbol{v} \in \operatorname{conv}\{\nabla\hat{f}_a(\boldsymbol{x})\}\big\}.$$

If $\boldsymbol{x}$ is Pareto-stationary, then $\mathbf{0}$ lies in this set, so the minimum is 0 and $\boldsymbol{d}_{\mathrm{mn}} = \mathbf{0}$. If $\boldsymbol{x}$ is not Pareto-stationary, then $\mathbf{0}$ does not lie in the set, so every element has positive norm and $\boldsymbol{d}_{\mathrm{mn}} \neq \mathbf{0}$. Combining the two cases, $\boldsymbol{d}_{\mathrm{mn}} = \mathbf{0}$ iff $\boldsymbol{x}$ is Pareto-stationary.

Now, when the projection vanishes at such a point, the operator takes the fallback, $\boldsymbol{d} = \boldsymbol{d}_{\mathrm{mn}} = \mathbf{0}$. The guidance term $\eta(t)\,\mathcal{G}_{\boldsymbol{w}}$ in Algorithm 1 is then zero, since the norm-matched strength $\eta(t) = \kappa\|\boldsymbol{v}_\theta\|/(\|\mathcal{G}_{\boldsymbol{w}}\|+\varepsilon)$ uses a small constant $\varepsilon > 0$ in the denominator, and the decode is the unmodified flow $\boldsymbol{v}_\theta$. □

D ADDITIONAL RESULTS

D.1 100TH PERCENTILE RESULTS

Tables 7–13 report the 100th-percentile hypervolume of the full returned set, from which the average ranks of Table 1 are computed, for MORL (Table 7), C-10/MOP (Table 8), IN-1K/MOP (Table 9), RE (Tables 10 and 12), DTLZ (Table 11), and ZDT (Table 13).

Table 7: 100th-percentile hypervolume results for MORL. For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	MO-Hopper	MO-Swimmer
$\mathcal{D}$(best)	6.33	3.70
E2E	5.56 ± 0.46	2.99 ± 0.10
E2E + GradNorm	6.17 ± 0.15	**3.55 ± 0.30**
E2E + PcGrad	5.65 ± 0.53	3.31 ± 0.17
MH	6.03 ± 0.22	3.43 ± 0.14
MH + GradNorm	**6.07 ± 0.57**	**3.61 ± 0.15**
MH + PcGrad	5.81 ± 0.15	3.41 ± 0.11
MM	5.78 ± 0.26	3.50 ± 0.20
MM + RoMA	**6.40 ± 0.20**	3.40 ± 0.18
MM + IOM	5.97 ± 0.38	3.46 ± 0.18
MM + ICT	**6.46 ± 0.27**	**3.74 ± 0.07**
MM + Tri-Mentor	5.88 ± 0.33	**3.73 ± 0.12**
PCD	**6.27 ± 0.20**	3.52 ± 0.13
ParetoFlow	5.88 ± 0.02	3.45 ± 0.19
ParetoFlow+	5.79 ± 0.13	3.31 ± 0.28
RFM-AN (ours)	6.18 ± 0.17	3.58 ± 0.07
RFM-APG (ours)	**6.40 ± 0.21**	3.59 ± 0.12
RFM-DAMG (ours)	6.12 ± 0.21	3.54 ± 0.08
RFM-Cone (ours)	6.17 ± 0.23	3.57 ± 0.06

Table 8: 100th-percentile hypervolume results for MO-NAS (Part 1, C-10/MOP). For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	C-10/MOP1	C-10/MOP2	C-10/MOP3	C-10/MOP4	C-10/MOP5	C-10/MOP6	C-10/MOP7	C-10/MOP8	C-10/MOP9
$\mathcal{D}$(best)	1.55	1.41	9.58	19.74	45.09	103.73	446.07	5.68	15.02
E2E	0.00 ± 0.00	0.00 ± 0.00	10.50 ± 0.14	22.27 ± 0.24	40.90 ± 2.56	104.16 ± 1.87	408.71 ± 28.08	5.79 ± 0.04	**15.47 ± 0.19**
E2E + GradNorm	1.22 ± 0.11	1.16 ± 0.08	10.48 ± 0.15	21.93 ± 0.38	39.60 ± 2.35	101.06 ± 3.63	402.82 ± 23.99	5.70 ± 0.09	**14.08 ± 1.66**
E2E + PcGrad	0.69 ± 0.63	0.46 ± 0.64	10.66 ± 0.09	22.29 ± 0.46	43.90 ± 0.15	102.28 ± 2.54	412.49 ± 33.85	5.79 ± 0.03	14.85 ± 0.41
MH	1.15 ± 0.10	0.99 ± 0.20	10.43 ± 0.17	21.71 ± 0.67	38.84 ± 0.12	104.04 ± 2.22	440.62 ± 13.07	5.82 ± 0.02	**15.24 ± 0.41**
MH + GradNorm	0.83 ± 0.56	**0.94 ± 0.53**	10.29 ± 0.21	21.64 ± 0.56	40.19 ± 2.21	97.93 ± 2.09	438.70 ± 16.65	5.36 ± 0.30	14.07 ± 0.75
MH + PcGrad	1.29 ± 0.07	1.10 ± 0.05	10.44 ± 0.16	22.14 ± 0.33	42.71 ± 2.19	101.37 ± 1.83	412.10 ± 21.71	5.78 ± 0.06	**15.45 ± 0.19**
MM	0.74 ± 0.68	0.54 ± 0.52	10.52 ± 0.15	22.31 ± 0.18	43.02 ± 1.63	104.80 ± 1.74	455.64 ± 18.20	**5.84 ± 0.01**	**15.21 ± 0.32**
MM + RoMA	1.28 ± 0.13	**0.95 ± 0.53**	10.31 ± 0.19	21.97 ± 0.20	43.85 ± 0.12	98.81 ± 3.67	467.53 ± 14.54	5.74 ± 0.08	**15.17 ± 0.34**
MM + IOM	1.42 ± 0.04	1.34 ± 0.03	**9.67 ± 1.89**	22.54 ± 0.61	47.48 ± 0.45	**107.24 ± 0.83**	468.83 ± 11.64	5.67 ± 0.12	15.35 ± 0.09
MM + ICT	1.19 ± 0.03	0.85 ± 0.47	10.37 ± 0.24	21.56 ± 0.62	47.43 ± 0.19	106.92 ± 1.03	467.58 ± 12.41	**5.83 ± 0.01**	**15.21 ± 0.38**
MM + Tri-Mentor	1.21 ± 0.05	1.35 ± 0.02	10.57 ± 0.11	21.96 ± 0.65	43.82 ± 0.33	104.85 ± 2.33	482.75 ± 9.78	5.77 ± 0.04	**15.29 ± 0.19**
PCD	**1.50 ± 0.01**	**1.41 ± 0.01**	10.72 ± 0.21	21.86 ± 0.37	43.47 ± 2.23	**107.98 ± 0.34**	473.13 ± 2.69	5.67 ± 0.03	15.05 ± 0.07
ParetoFlow	1.49 ± 0.01	1.38 ± 0.01	10.27 ± 0.40	21.95 ± 0.37	45.85 ± 2.30	**107.16 ± 0.92**	437.95 ± 15.76	5.47 ± 0.11	14.90 ± 0.09
ParetoFlow+	1.48 ± 0.00	1.38 ± 0.00	10.42 ± 0.15	21.50 ± 0.34	44.79 ± 3.23	106.34 ± 1.50	450.40 ± 6.67	5.59 ± 0.03	14.95 ± 0.09
RFM-AN (ours)	1.47 ± 0.01	1.39 ± 0.01	**11.18 ± 0.04**	**23.81 ± 0.17**	**48.63 ± 0.87**	**107.29 ± 1.86**	**489.41 ± 7.90**	5.66 ± 0.03	14.85 ± 0.11
RFM-APG (ours)	1.48 ± 0.02	1.38 ± 0.00	**11.18 ± 0.02**	**23.92 ± 0.06**	**48.63 ± 0.88**	**107.07 ± 2.15**	**489.00 ± 7.13**	5.66 ± 0.03	14.84 ± 0.11
RFM-DAMG (ours)	1.48 ± 0.02	1.39 ± 0.01	**11.18 ± 0.04**	**23.82 ± 0.14**	**48.57 ± 0.85**	**107.10 ± 2.03**	**490.82 ± 4.93**	5.66 ± 0.03	14.85 ± 0.11
RFM-Cone (ours)	**1.46 ± 0.04**	1.38 ± 0.01	11.16 ± 0.02	**23.85 ± 0.37**	**48.54 ± 0.95**	**106.79 ± 2.36**	**494.03 ± 5.57**	5.69 ± 0.04	15.01 ± 0.15

Table 9: 100th-percentile hypervolume results for MO-NAS (Part 2, IN-1K/MOP). For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	IN-1K/MOP1	IN-1K/MOP2	IN-1K/MOP3	IN-1K/MOP4	IN-1K/MOP5	IN-1K/MOP6	IN-1K/MOP7	IN-1K/MOP8	IN-1K/MOP9
$\mathcal{D}$(best)	6.60	7.44	17.59	4.23	4.57	11.36	5.73	14.04	19.66
E2E	5.54 ± 0.22	5.67 ± 0.32	13.96 ± 0.20	**4.51 ± 0.20**	4.85 ± 0.05	**12.49 ± 0.10**	4.86 ± 0.09	9.37 ± 0.46	11.40 ± 0.92
E2E + GradNorm	5.33 ± 0.05	6.18 ± 0.08	13.30 ± 0.43	4.12 ± 0.13	4.36 ± 0.05	9.94 ± 0.20	4.75 ± 0.31	8.57 ± 0.23	10.37 ± 0.37
E2E + PcGrad	5.70 ± 0.13	6.27 ± 0.02	14.10 ± 0.29	4.48 ± 0.05	4.65 ± 0.06	11.86 ± 0.14	4.95 ± 0.10	9.54 ± 0.17	11.63 ± 0.14
MH	5.59 ± 0.04	5.96 ± 0.12	13.67 ± 0.08	4.52 ± 0.04	4.76 ± 0.05	11.53 ± 0.12	4.87 ± 0.04	9.43 ± 0.07	11.96 ± 0.31
MH + GradNorm	5.44 ± 0.22	6.28 ± 0.03	13.72 ± 0.45	3.30 ± 0.38	**4.45 ± 0.49**	8.68 ± 0.34	4.06 ± 0.15	8.71 ± 0.97	13.56 ± 1.91
MH + PcGrad	5.49 ± 0.09	6.33 ± 0.03	13.70 ± 0.29	4.45 ± 0.05	4.76 ± 0.02	11.71 ± 0.17	4.96 ± 0.08	9.45 ± 0.22	11.47 ± 0.41
MM	5.73 ± 0.14	6.01 ± 0.24	14.14 ± 0.27	**4.62 ± 0.07**	**4.91 ± 0.06**	12.38 ± 0.11	4.95 ± 0.02	9.77 ± 0.03	12.00 ± 0.39
MM + RoMA	5.80 ± 0.09	6.03 ± 0.36	13.89 ± 0.28	4.50 ± 0.07	4.75 ± 0.09	12.20 ± 0.16	5.07 ± 0.05	9.67 ± 0.12	10.97 ± 0.85
MM + IOM	**6.02 ± 0.05**	6.39 ± 0.05	**15.78 ± 0.01**	4.51 ± 0.04	**4.72 ± 0.19**	11.73 ± 0.14	5.09 ± 0.12	11.92 ± 0.05	**16.09 ± 0.10**
MM + ICT	**6.06 ± 0.03**	6.29 ± 0.07	14.81 ± 0.07	4.44 ± 0.04	4.82 ± 0.05	12.11 ± 0.07	5.22 ± 0.06	10.86 ± 0.20	13.38 ± 0.43
MM + Tri-Mentor	5.58 ± 0.05	6.29 ± 0.15	14.98 ± 0.06	4.44 ± 0.03	4.81 ± 0.03	12.06 ± 0.09	4.78 ± 0.06	9.37 ± 0.12	13.03 ± 0.89
PCD	**6.03 ± 0.06**	**6.50 ± 0.04**	15.73 ± 0.02	4.44 ± 0.02	4.81 ± 0.02	12.02 ± 0.06	**5.36 ± 0.03**	11.88 ± 0.12	15.71 ± 0.12
ParetoFlow	5.71 ± 0.06	6.34 ± 0.04	15.53 ± 0.05	4.37 ± 0.04	4.68 ± 0.07	11.79 ± 0.11	4.86 ± 0.17	11.81 ± 0.05	15.66 ± 0.05
ParetoFlow+	5.76 ± 0.01	6.43 ± 0.06	15.55 ± 0.06	4.36 ± 0.07	4.65 ± 0.04	11.84 ± 0.10	4.91 ± 0.07	11.96 ± 0.08	15.72 ± 0.08
RFM-AN (ours)	5.85 ± 0.03	6.23 ± 0.26	15.59 ± 0.10	4.44 ± 0.08	4.82 ± 0.04	11.93 ± 0.16	5.21 ± 0.04	**12.15 ± 0.06**	**16.11 ± 0.13**
RFM-APG (ours)	5.85 ± 0.01	6.23 ± 0.26	15.65 ± 0.03	4.45 ± 0.08	4.82 ± 0.04	11.93 ± 0.15	5.21 ± 0.05	**12.15 ± 0.06**	**16.13 ± 0.10**
RFM-DAMG (ours)	5.86 ± 0.03	6.24 ± 0.24	15.62 ± 0.09	4.44 ± 0.08	4.82 ± 0.04	11.97 ± 0.13	5.20 ± 0.03	**12.15 ± 0.06**	**16.07 ± 0.15**
RFM-Cone (ours)	5.85 ± 0.02	6.32 ± 0.08	15.66 ± 0.04	4.45 ± 0.06	4.81 ± 0.04	12.01 ± 0.19	**5.29 ± 0.13**	**12.15 ± 0.08**	**16.17 ± 0.07**

Table 10: 100th-percentile hypervolume results for RE (Part 1). For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33
$\mathcal{D}$(best)	4.18	4.81	4.75	4.60	4.79	10.61	10.61	10.53
E2E	4.26 ± 0.32	**4.84 ± 0.00**	**4.84 ± 0.00**	4.28 ± 0.02	4.35 ± 0.00	**10.65 ± 0.00**	10.63 ± 0.01	10.62 ± 0.00
E2E + GradNorm	4.47 ± 0.07	**4.84 ± 0.00**	**4.81 ± 0.03**	**4.69 ± 0.15**	**4.84 ± 0.00**	**10.61 ± 0.06**	10.62 ± 0.01	10.57 ± 0.04
E2E + PcGrad	**4.60 ± 0.00**	**4.84 ± 0.00**	**4.84 ± 0.00**	**4.74 ± 0.16**	**4.84 ± 0.00**	**10.65 ± 0.00**	10.64 ± 0.00	10.60 ± 0.01
MH	**4.60 ± 0.00**	**4.84 ± 0.00**	**4.84 ± 0.00**	**4.63 ± 0.25**	**4.84 ± 0.00**	**10.65 ± 0.00**	10.63 ± 0.00	10.62 ± 0.00
MH + GradNorm	**4.41 ± 0.25**	**4.84 ± 0.00**	4.80 ± 0.03	**4.49 ± 0.76**	**4.84 ± 0.00**	**10.65 ± 0.01**	7.88 ± 1.23	10.60 ± 0.02
MH + PcGrad	**4.60 ± 0.00**	**4.84 ± 0.00**	**4.39 ± 0.71**	4.79 ± 0.03	**4.84 ± 0.00**	**10.65 ± 0.00**	10.63 ± 0.00	10.62 ± 0.01
MM	**4.59 ± 0.01**	**4.84 ± 0.00**	**4.84 ± 0.00**	4.42 ± 0.10	4.63 ± 0.19	**10.65 ± 0.00**	**10.64 ± 0.01**	10.62 ± 0.00
MM + RoMA	4.46 ± 0.07	**4.84 ± 0.00**	4.82 ± 0.01	4.70 ± 0.09	**4.83 ± 0.01**	**10.65 ± 0.00**	10.55 ± 0.07	10.58 ± 0.02
MM + IOM	4.60 ± 0.00	**4.84 ± 0.00**	4.83 ± 0.00	**4.73 ± 0.21**	**4.83 ± 0.02**	**10.65 ± 0.00**	**10.65 ± 0.00**	10.62 ± 0.01
MM + ICT	4.60 ± 0.00	**4.84 ± 0.00**	**4.83 ± 0.01**	**4.84 ± 0.00**	**4.84 ± 0.00**	**10.65 ± 0.00**	**10.65 ± 0.00**	10.60 ± 0.02
MM + Tri-Mentor	4.60 ± 0.00	**4.84 ± 0.00**	**4.84 ± 0.00**	**4.84 ± 0.00**	**4.84 ± 0.00**	**10.64 ± 0.00**	**10.65 ± 0.00**	10.61 ± 0.01
PCD	4.40 ± 0.01	**4.84 ± 0.00**	4.83 ± 0.00	**4.84 ± 0.00**	**4.84 ± 0.00**	10.63 ± 0.01	**10.65 ± 0.00**	10.60 ± 0.01
ParetoFlow	4.13 ± 0.08	**4.81 ± 0.04**	4.80 ± 0.02	4.83 ± 0.01	4.82 ± 0.01	**10.56 ± 0.10**	10.62 ± 0.01	10.36 ± 0.10
ParetoFlow+	4.12 ± 0.03	4.78 ± 0.04	4.78 ± 0.02	4.71 ± 0.07	4.80 ± 0.00	10.61 ± 0.00	10.61 ± 0.01	10.46 ± 0.03
RFM-AN (ours)	4.56 ± 0.02	**4.81 ± 0.05**	**4.84 ± 0.00**	4.60 ± 0.22	**4.84 ± 0.00**	6.43 ± 3.08	**10.65 ± 0.00**	**10.63 ± 0.00**
RFM-APG (ours)	4.57 ± 0.01	**4.84 ± 0.00**	**4.84 ± 0.00**	4.60 ± 0.22	**4.84 ± 0.00**	7.17 ± 3.26	10.64 ± 0.00	**10.63 ± 0.00**
RFM-DAMG (ours)	4.56 ± 0.01	**4.84 ± 0.00**	**4.84 ± 0.00**	**4.62 ± 0.23**	**4.84 ± 0.00**	5.92 ± 3.27	**10.65 ± 0.00**	**10.63 ± 0.00**
RFM-Cone (ours)	4.56 ± 0.02	**4.69 ± 0.34**	**4.84 ± 0.00**	4.60 ± 0.22	**4.84 ± 0.00**	6.43 ± 3.09	**10.65 ± 0.00**	**10.63 ± 0.00**

Table 11: 100th-percentile hypervolume results for DTLZ. For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
$\mathcal{D}$(best)	10.61	12.43	9.90	17.50	20.98	15.35	8.99
E2E	**10.65 ± 0.00**	**12.45 ± 0.00**	**9.86 ± 0.07**	**17.66 ± 0.03**	21.06 ± 0.00	**15.74 ± 0.01**	**10.80 ± 0.00**
E2E + GradNorm	10.65 ± 0.00	**12.44 ± 0.01**	9.89 ± 0.00	17.53 ± 0.07	21.05 ± 0.01	15.32 ± 0.14	10.69 ± 0.02
E2E + PcGrad	10.65 ± 0.00	**12.44 ± 0.00**	9.89 ± 0.01	17.62 ± 0.04	**21.06 ± 0.00**	15.67 ± 0.01	10.77 ± 0.00
MH	10.64 ± 0.00	**12.44 ± 0.00**	9.89 ± 0.00	**17.61 ± 0.11**	21.06 ± 0.00	15.66 ± 0.01	10.78 ± 0.01
MH + GradNorm	10.64 ± 0.00	12.44 ± 0.00	9.88 ± 0.00	17.56 ± 0.03	20.94 ± 0.05	15.55 ± 0.04	10.77 ± 0.01
MH + PcGrad	10.64 ± 0.00	**12.45 ± 0.00**	9.89 ± 0.00	17.55 ± 0.06	21.06 ± 0.00	15.58 ± 0.01	10.70 ± 0.07
MM	**10.65 ± 0.00**	**12.44 ± 0.00**	9.89 ± 0.00	17.58 ± 0.02	**21.06 ± 0.00**	15.64 ± 0.01	10.78 ± 0.00
MM + RoMA	10.64 ± 0.00	12.43 ± 0.00	9.88 ± 0.00	17.23 ± 0.07	20.99 ± 0.02	15.50 ± 0.02	10.49 ± 0.04
MM + IOM	10.64 ± 0.00	12.44 ± 0.00	**9.90 ± 0.00**	17.10 ± 0.41	21.06 ± 0.00	15.50 ± 0.01	10.52 ± 0.06
MM + ICT	10.65 ± 0.00	12.43 ± 0.00	9.89 ± 0.00	17.49 ± 0.05	20.99 ± 0.01	15.42 ± 0.07	10.72 ± 0.01
MM + Tri-Mentor	10.64 ± 0.00	12.44 ± 0.00	9.17 ± 0.17	17.55 ± 0.03	20.97 ± 0.03	15.58 ± 0.02	10.46 ± 0.16
PCD	10.65 ± 0.00	12.41 ± 0.00	9.89 ± 0.00	**17.66 ± 0.01**	20.92 ± 0.02	15.61 ± 0.05	9.45 ± 0.03
ParetoFlow	10.61 ± 0.01	12.28 ± 0.09	9.77 ± 0.04	17.53 ± 0.05	20.87 ± 0.03	15.40 ± 0.06	8.89 ± 0.06
ParetoFlow+	10.60 ± 0.01	12.30 ± 0.02	9.82 ± 0.01	17.45 ± 0.10	20.86 ± 0.04	15.36 ± 0.04	8.85 ± 0.06
RFM-AN (ours)	**10.65 ± 0.00**	12.12 ± 0.03	9.89 ± 0.00	17.26 ± 0.32	20.38 ± 0.05	15.64 ± 0.05	10.40 ± 0.07
RFM-APG (ours)	**10.65 ± 0.00**	12.13 ± 0.03	**9.90 ± 0.00**	17.21 ± 0.40	20.38 ± 0.07	15.66 ± 0.05	10.51 ± 0.07
RFM-DAMG (ours)	**10.65 ± 0.00**	12.13 ± 0.02	**9.90 ± 0.00**	17.18 ± 0.40	20.39 ± 0.06	15.66 ± 0.03	10.53 ± 0.07
RFM-Cone (ours)	**10.65 ± 0.00**	12.17 ± 0.03	**9.90 ± 0.00**	17.26 ± 0.28	20.48 ± 0.06	15.69 ± 0.04	10.52 ± 0.07

Table 12: 100th-percentile hypervolume results for RE (Part 2). For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	RE34	RE35	RE36	RE37	RE41	RE42	RE61
$\mathcal{D}$(best)	9.65	10.22	7.76	5.92	18.99	15.28	105.13
E2E	**10.12 ± 0.00**	10.52 ± 0.02	10.02 ± 0.12	**6.75 ± 0.00**	**20.85 ± 0.05**	22.38 ± 0.36	**108.89 ± 0.03**
E2E + GradNorm	9.85 ± 0.22	10.38 ± 0.02	9.83 ± 0.24	6.65 ± 0.02	17.52 ± 0.57	21.53 ± 0.26	102.69 ± 2.91
E2E + PcGrad	**10.12 ± 0.00**	**10.58 ± 0.01**	**10.17 ± 0.22**	**6.75 ± 0.00**	**20.89 ± 0.03**	**22.71 ± 0.09**	104.63 ± 0.85
MH	10.11 ± 0.00	**10.56 ± 0.02**	10.07 ± 0.02	**6.75 ± 0.01**	20.76 ± 0.04	22.51 ± 0.14	**108.77 ± 0.16**
MH + GradNorm	9.79 ± 0.07	10.41 ± 0.07	7.39 ± 1.88	6.68 ± 0.03	20.75 ± 0.06	22.24 ± 0.14	105.61 ± 2.98
MH + PcGrad	10.11 ± 0.00	**10.57 ± 0.01**	10.08 ± 0.15	6.74 ± 0.01	20.78 ± 0.02	22.53 ± 0.05	107.66 ± 0.15
MM	10.11 ± 0.01	10.53 ± 0.03	9.91 ± 0.14	6.74 ± 0.01	20.76 ± 0.04	22.45 ± 0.05	**108.92 ± 0.08**
MM + RoMA	10.01 ± 0.03	**10.58 ± 0.00**	**9.43 ± 1.25**	6.70 ± 0.01	20.67 ± 0.10	21.36 ± 0.24	108.40 ± 0.26
MM + IOM	10.10 ± 0.01	**10.58 ± 0.00**	10.22 ± 0.09	6.72 ± 0.01	20.60 ± 0.04	21.89 ± 0.37	108.01 ± 0.22
MM + ICT	10.09 ± 0.02	10.57 ± 0.00	10.27 ± 0.02	6.70 ± 0.01	20.73 ± 0.05	**22.75 ± 0.06**	108.57 ± 0.15
MM + Tri-Mentor	10.09 ± 0.01	10.36 ± 0.20	**10.31 ± 0.01**	6.74 ± 0.01	20.71 ± 0.08	22.69 ± 0.05	108.33 ± 0.42
PCD	9.70 ± 0.01	10.39 ± 0.01	8.28 ± 0.05	5.97 ± 0.01	18.76 ± 0.04	15.71 ± 0.05	101.18 ± 0.06
ParetoFlow	9.71 ± 0.05	10.41 ± 0.10	8.84 ± 0.21	5.70 ± 0.34	18.33 ± 0.44	17.55 ± 1.14	105.54 ± 1.57
ParetoFlow+	9.68 ± 0.02	10.32 ± 0.03	7.98 ± 0.28	5.66 ± 0.18	18.24 ± 0.09	16.29 ± 0.34	105.66 ± 1.38
RFM-AN (ours)	9.93 ± 0.01	**10.54 ± 0.05**	9.27 ± 0.24	6.73 ± 0.01	20.45 ± 0.05	22.16 ± 0.31	106.75 ± 1.47
RFM-APG (ours)	9.93 ± 0.01	10.54 ± 0.04	9.34 ± 0.41	6.73 ± 0.01	20.51 ± 0.04	22.32 ± 0.28	107.44 ± 0.56
RFM-DAMG (ours)	9.93 ± 0.01	10.54 ± 0.03	9.31 ± 0.28	**6.73 ± 0.02**	20.54 ± 0.02	22.25 ± 0.30	106.63 ± 1.24
RFM-Cone (ours)	9.93 ± 0.01	10.53 ± 0.04	9.37 ± 0.28	**6.71 ± 0.06**	20.64 ± 0.06	22.24 ± 0.29	106.85 ± 1.42

Table 13: 100th-percentile hypervolume results for ZDT. For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
$\mathcal{D}$(best)	4.17	4.68	5.15	5.46	4.61
E2E	**4.86 ± 0.00**	**5.71 ± 0.00**	5.71 ± 0.03	4.68 ± 0.24	**4.80 ± 0.00**
E2E + GradNorm	4.16 ± 0.07	5.50 ± 0.14	5.02 ± 0.06	5.09 ± 0.09	4.41 ± 0.08
E2E + PcGrad	4.86 ± 0.00	**5.71 ± 0.00**	5.74 ± 0.09	4.11 ± 0.12	**3.90 ± 1.07**
MH	4.85 ± 0.00	5.63 ± 0.01	**5.80 ± 0.04**	5.14 ± 0.18	4.78 ± 0.00
MH + GradNorm	**4.82 ± 0.04**	**5.63 ± 0.08**	5.42 ± 0.15	3.82 ± 0.08	3.32 ± 1.05
MH + PcGrad	4.83 ± 0.01	5.53 ± 0.06	5.69 ± 0.08	3.78 ± 0.35	**4.31 ± 0.74**
MM	4.84 ± 0.01	5.62 ± 0.01	**5.80 ± 0.08**	5.06 ± 0.15	4.77 ± 0.01
MM + RoMA	4.84 ± 0.00	5.60 ± 0.03	5.68 ± 0.11	**5.03 ± 0.45**	2.69 ± 0.52
MM + IOM	4.37 ± 0.08	4.98 ± 0.05	5.52 ± 0.04	**5.34 ± 0.10**	4.75 ± 0.01
MM + ICT	4.85 ± 0.00	5.64 ± 0.01	5.59 ± 0.02	3.50 ± 0.06	4.67 ± 0.04
MM + Tri-Mentor	4.49 ± 0.02	5.42 ± 0.02	5.39 ± 0.06	**5.39 ± 0.08**	2.56 ± 0.14
PCD	4.39 ± 0.02	4.99 ± 0.03	5.48 ± 0.05	5.23 ± 0.04	4.66 ± 0.11
ParetoFlow	4.14 ± 0.04	4.69 ± 0.07	5.21 ± 0.04	5.04 ± 0.10	4.64 ± 0.02
ParetoFlow+	4.17 ± 0.03	4.69 ± 0.03	5.20 ± 0.05	**5.42 ± 0.05**	4.67 ± 0.01
RFM-AN (ours)	4.64 ± 0.04	5.12 ± 0.03	5.81 ± 0.02	4.88 ± 0.06	3.98 ± 0.49
RFM-APG (ours)	4.66 ± 0.04	5.17 ± 0.03	**5.82 ± 0.02**	4.83 ± 0.12	4.05 ± 0.44
RFM-DAMG (ours)	4.67 ± 0.04	5.20 ± 0.03	**5.84 ± 0.02**	4.89 ± 0.14	4.19 ± 0.45
RFM-Cone (ours)	4.68 ± 0.04	5.19 ± 0.02	**5.84 ± 0.02**	4.89 ± 0.14	4.28 ± 0.47

D.2 75TH PERCENTILE RESULTS

Tables 14–20 report the 75th-percentile hypervolume for DTLZ (Table 14), ZDT (Table 15), C-10/MOP (Table 16), IN-1K/MOP (Table 17), MORL (Table 18), and RE (Tables 19 and 20). $\mathcal{D}$(best) is reported only at the 100th percentile.

Table 14: 75th-percentile hypervolume results for DTLZ. For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
$\mathcal{D}$(best)	–	–	–	–	–	–	–
E2E	**10.65 ± 0.00**	**12.44 ± 0.00**	**9.83 ± 0.09**	**17.65 ± 0.03**	21.06 ± 0.00	**15.72 ± 0.01**	**10.80 ± 0.00**
E2E + GradNorm	10.64 ± 0.00	**12.43 ± 0.02**	9.87 ± 0.00	17.35 ± 0.22	**21.04 ± 0.02**	15.31 ± 0.14	10.67 ± 0.04
E2E + PcGrad	10.64 ± 0.00	**12.44 ± 0.00**	**9.88 ± 0.01**	17.47 ± 0.08	**21.06 ± 0.00**	15.66 ± 0.01	10.77 ± 0.01
MH	10.64 ± 0.00	**12.44 ± 0.00**	9.87 ± 0.01	**17.57 ± 0.13**	**21.05 ± 0.02**	**15.01 ± 0.78**	10.77 ± 0.01
MH + GradNorm	10.64 ± 0.00	12.43 ± 0.00	9.86 ± 0.01	17.47 ± 0.03	20.89 ± 0.14	15.54 ± 0.04	10.76 ± 0.01
MH + PcGrad	10.64 ± 0.00	**12.44 ± 0.00**	9.87 ± 0.01	**17.24 ± 0.44**	21.05 ± 0.01	15.57 ± 0.01	10.69 ± 0.08
MM	10.64 ± 0.00	**12.44 ± 0.00**	9.86 ± 0.02	17.57 ± 0.02	**21.05 ± 0.02**	15.62 ± 0.01	10.77 ± 0.01
MM + RoMA	10.63 ± 0.00	12.43 ± 0.00	9.85 ± 0.01	17.04 ± 0.19	20.95 ± 0.04	15.48 ± 0.02	10.35 ± 0.03
MM + IOM	10.64 ± 0.00	12.44 ± 0.00	**9.89 ± 0.00**	16.82 ± 0.81	21.06 ± 0.00	15.49 ± 0.01	10.46 ± 0.11
MM + ICT	10.64 ± 0.00	12.43 ± 0.00	9.88 ± 0.01	17.28 ± 0.19	20.89 ± 0.04	15.41 ± 0.07	10.70 ± 0.01
MM + Tri-Mentor	10.63 ± 0.00	12.43 ± 0.01	8.86 ± 0.19	17.43 ± 0.05	20.96 ± 0.04	15.56 ± 0.02	10.31 ± 0.28
PCD	10.57 ± 0.02	12.39 ± 0.01	9.86 ± 0.01	17.35 ± 0.22	20.81 ± 0.04	15.45 ± 0.03	9.35 ± 0.06
ParetoFlow	10.60 ± 0.01	12.26 ± 0.08	9.52 ± 0.19	**16.26 ± 1.49**	20.71 ± 0.23	15.35 ± 0.08	8.80 ± 0.08
ParetoFlow+	10.59 ± 0.02	12.28 ± 0.02	9.55 ± 0.05	14.73 ± 0.65	20.69 ± 0.09	15.30 ± 0.06	8.84 ± 0.06
RFM-AN (ours)	10.62 ± 0.01	11.67 ± 0.50	9.85 ± 0.01	16.33 ± 0.67	15.17 ± 0.38	15.43 ± 0.02	10.11 ± 0.20
RFM-APG (ours)	10.62 ± 0.01	11.82 ± 0.12	9.85 ± 0.01	16.55 ± 0.57	15.23 ± 0.33	15.45 ± 0.03	10.29 ± 0.14
RFM-DAMG (ours)	10.62 ± 0.01	11.69 ± 0.39	9.86 ± 0.01	15.78 ± 1.70	16.02 ± 1.68	15.29 ± 0.38	10.28 ± 0.21
RFM-Cone (ours)	10.62 ± 0.01	11.96 ± 0.04	9.85 ± 0.01	16.58 ± 0.59	15.63 ± 0.22	15.50 ± 0.03	10.30 ± 0.12

Table 15: 75th-percentile hypervolume results for ZDT. For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
$\mathcal{D}$(best)	–	–	–	–	–
E2E	**4.86 ± 0.00**	**5.71 ± 0.00**	5.65 ± 0.04	4.52 ± 0.26	**4.79 ± 0.00**
E2E + GradNorm	3.29 ± 0.03	**5.21 ± 0.53**	4.34 ± 0.06	4.80 ± 0.05	3.33 ± 1.08
E2E + PcGrad	4.86 ± 0.00	**5.71 ± 0.00**	**5.55 ± 0.36**	3.92 ± 0.19	1.97 ± 0.29
MH	4.85 ± 0.00	5.62 ± 0.01	**5.77 ± 0.05**	4.88 ± 0.11	4.76 ± 0.00
MH + GradNorm	**4.82 ± 0.05**	**5.62 ± 0.08**	5.29 ± 0.17	3.48 ± 0.05	3.28 ± 1.04
MH + PcGrad	4.83 ± 0.01	5.47 ± 0.05	5.64 ± 0.09	3.57 ± 0.34	**4.26 ± 0.75**
MM	4.84 ± 0.01	5.61 ± 0.02	**5.76 ± 0.10**	4.83 ± 0.18	4.76 ± 0.00
MM + RoMA	4.84 ± 0.00	5.58 ± 0.03	5.65 ± 0.11	4.76 ± 0.44	2.01 ± 0.14
MM + IOM	4.35 ± 0.08	4.93 ± 0.03	5.48 ± 0.04	**5.17 ± 0.13**	4.72 ± 0.04
MM + ICT	4.85 ± 0.00	5.64 ± 0.01	5.55 ± 0.03	3.36 ± 0.08	**3.69 ± 1.24**
MM + Tri-Mentor	4.48 ± 0.02	5.38 ± 0.03	5.28 ± 0.09	5.10 ± 0.05	2.39 ± 0.08
PCD	4.32 ± 0.01	4.85 ± 0.03	5.26 ± 0.01	4.86 ± 0.01	4.37 ± 0.01
ParetoFlow	4.13 ± 0.05	4.68 ± 0.07	5.15 ± 0.11	4.82 ± 0.12	4.53 ± 0.05
ParetoFlow+	4.17 ± 0.03	4.68 ± 0.03	5.19 ± 0.05	**5.21 ± 0.08**	4.58 ± 0.05
RFM-AN (ours)	4.53 ± 0.04	5.04 ± 0.03	5.65 ± 0.03	4.20 ± 0.10	2.72 ± 0.21
RFM-APG (ours)	4.60 ± 0.05	5.09 ± 0.03	5.69 ± 0.05	4.23 ± 0.13	3.24 ± 0.34
RFM-DAMG (ours)	4.58 ± 0.06	5.11 ± 0.03	5.69 ± 0.08	4.19 ± 0.12	3.25 ± 0.41
RFM-Cone (ours)	4.60 ± 0.04	5.10 ± 0.02	**5.74 ± 0.05**	4.23 ± 0.13	3.27 ± 0.39

Table 16: 75th-percentile hypervolume results for MO-NAS (Part 1, C-10/MOP). For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	C-10/MOP1	C-10/MOP2	C-10/MOP3	C-10/MOP4	C-10/MOP5	C-10/MOP6	C-10/MOP7	C-10/MOP8	C-10/MOP9
$\mathcal{D}$(best)	–	–	–	–	–	–	–	–	–
E2E	0.00 ± 0.00	0.00 ± 0.00	10.35 ± 0.21	22.10 ± 0.22	40.22 ± 2.23	103.01 ± 2.57	405.35 ± 29.34	5.75 ± 0.05	**15.38 ± 0.18**
E2E + GradNorm	1.20 ± 0.10	1.16 ± 0.08	10.02 ± 0.12	21.87 ± 0.35	38.07 ± 2.14	98.91 ± 3.63	402.47 ± 23.92	5.51 ± 0.09	**13.78 ± 1.85**
E2E + PcGrad	0.68 ± 0.62	0.26 ± 0.58	10.48 ± 0.24	22.08 ± 0.28	42.50 ± 0.80	101.72 ± 2.49	411.63 ± 33.26	5.77 ± 0.02	14.58 ± 0.32
MH	0.91 ± 0.52	0.98 ± 0.20	10.22 ± 0.32	21.36 ± 1.02	38.41 ± 0.62	101.89 ± 1.48	435.49 ± 6.14	**5.77 ± 0.05**	**15.02 ± 0.42**
MH + GradNorm	0.82 ± 0.55	**0.91 ± 0.51**	9.94 ± 0.43	21.22 ± 0.65	37.69 ± 1.26	96.99 ± 1.79	408.37 ± 24.70	5.27 ± 0.26	12.19 ± 1.39
MH + PcGrad	1.23 ± 0.09	1.09 ± 0.04	10.32 ± 0.15	21.88 ± 0.63	41.39 ± 2.08	100.84 ± 2.33	401.48 ± 19.89	5.71 ± 0.07	**15.26 ± 0.24**
MM	0.49 ± 0.67	0.39 ± 0.55	10.23 ± 0.32	22.16 ± 0.26	40.55 ± 1.47	102.59 ± 3.12	449.14 ± 18.52	**5.78 ± 0.03**	**15.03 ± 0.35**
MM + RoMA	1.18 ± 0.25	**0.94 ± 0.53**	10.09 ± 0.23	21.79 ± 0.45	42.58 ± 0.75	97.88 ± 3.76	453.87 ± 17.95	5.70 ± 0.08	14.93 ± 0.31
MM + IOM	1.41 ± 0.05	1.34 ± 0.03	8.73 ± 1.91	20.87 ± 0.86	45.91 ± 1.00	**105.29 ± 2.84**	455.58 ± 25.13	5.53 ± 0.07	**15.30 ± 0.10**
MM + ICT	1.18 ± 0.03	0.64 ± 0.58	10.18 ± 0.34	21.12 ± 0.58	45.89 ± 1.08	**106.33 ± 0.74**	458.31 ± 20.57	**5.81 ± 0.02**	14.90 ± 0.39
MM + Tri-Mentor	1.20 ± 0.03	1.32 ± 0.05	10.44 ± 0.08	21.86 ± 0.60	43.21 ± 0.19	**103.01 ± 4.42**	479.70 ± 6.60	5.73 ± 0.05	**15.18 ± 0.24**
PCD	**1.48 ± 0.01**	1.36 ± 0.00	10.48 ± 0.34	21.03 ± 1.12	41.19 ± 0.33	**106.19 ± 1.04**	471.85 ± 3.44	5.58 ± 0.01	14.71 ± 0.03
ParetoFlow	**1.46 ± 0.02**	1.34 ± 0.02	9.55 ± 0.21	20.83 ± 0.53	42.97 ± 3.89	**105.73 ± 1.62**	428.92 ± 14.71	5.36 ± 0.16	14.64 ± 0.05
ParetoFlow+	1.47 ± 0.01	1.36 ± 0.01	10.20 ± 0.12	20.53 ± 0.47	42.26 ± 1.95	**105.66 ± 1.83**	421.89 ± 9.18	5.46 ± 0.17	14.74 ± 0.09
RFM-AN (ours)	**1.47 ± 0.01**	**1.39 ± 0.01**	**11.17 ± 0.04**	**23.65 ± 0.30**	**47.58 ± 0.55**	**105.79 ± 3.45**	**487.05 ± 7.86**	5.56 ± 0.03	14.43 ± 0.11
RFM-APG (ours)	**1.44 ± 0.06**	1.38 ± 0.00	**11.17 ± 0.01**	**23.80 ± 0.05**	**47.38 ± 0.27**	**105.08 ± 3.59**	**487.56 ± 6.51**	5.56 ± 0.03	14.46 ± 0.13
RFM-DAMG (ours)	**1.45 ± 0.06**	**1.38 ± 0.00**	**11.16 ± 0.05**	**23.67 ± 0.29**	**47.40 ± 0.92**	**105.60 ± 3.96**	**489.06 ± 4.07**	5.57 ± 0.03	14.44 ± 0.11
RFM-Cone (ours)	**1.45 ± 0.04**	**1.38 ± 0.01**	11.15 ± 0.01	**23.76 ± 0.33**	**47.29 ± 0.50**	**105.13 ± 3.42**	**492.66 ± 6.43**	5.62 ± 0.03	14.68 ± 0.17

Table 17: 75th-percentile hypervolume results for MO-NAS (Part 2, IN-1K/MOP). For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	IN-1K/MOP1	IN-1K/MOP2	IN-1K/MOP3	IN-1K/MOP4	IN-1K/MOP5	IN-1K/MOP6	IN-1K/MOP7	IN-1K/MOP8	IN-1K/MOP9
$\mathcal{D}$(best)	–	–	–	–	–	–	–	–	–
E2E	5.52 ± 0.23	5.66 ± 0.33	13.91 ± 0.20	**4.49 ± 0.21**	4.84 ± 0.04	11.83 ± 0.37	4.75 ± 0.15	9.31 ± 0.48	11.33 ± 0.91
E2E + GradNorm	5.26 ± 0.02	6.16 ± 0.09	13.23 ± 0.36	4.09 ± 0.12	4.31 ± 0.08	9.66 ± 0.14	4.54 ± 0.36	8.56 ± 0.23	10.29 ± 0.40
E2E + PcGrad	5.68 ± 0.13	6.25 ± 0.02	14.06 ± 0.26	4.44 ± 0.09	4.61 ± 0.04	11.77 ± 0.24	4.85 ± 0.11	9.52 ± 0.18	11.58 ± 0.15
MH	5.56 ± 0.05	5.87 ± 0.25	13.58 ± 0.04	4.48 ± 0.03	4.72 ± 0.03	10.98 ± 0.90	4.76 ± 0.03	9.33 ± 0.09	11.81 ± 0.44
MH + GradNorm	5.43 ± 0.23	6.22 ± 0.06	13.70 ± 0.43	3.22 ± 0.39	**4.41 ± 0.50**	8.52 ± 0.34	3.83 ± 0.24	8.66 ± 0.96	13.52 ± 1.90
MH + PcGrad	5.44 ± 0.09	6.27 ± 0.03	13.69 ± 0.30	4.38 ± 0.05	4.68 ± 0.05	11.38 ± 0.24	4.84 ± 0.10	9.42 ± 0.21	11.43 ± 0.47
MM	5.70 ± 0.13	5.91 ± 0.20	14.08 ± 0.27	**4.60 ± 0.06**	**4.90 ± 0.05**	**12.25 ± 0.20**	4.91 ± 0.03	9.68 ± 0.14	11.94 ± 0.36
MM + RoMA	5.69 ± 0.05	**5.88 ± 0.53**	13.87 ± 0.29	4.33 ± 0.23	4.57 ± 0.17	**12.03 ± 0.25**	4.96 ± 0.06	9.66 ± 0.11	10.91 ± 0.90
MM + IOM	5.99 ± 0.04	**6.34 ± 0.07**	**15.75 ± 0.03**	4.49 ± 0.05	4.64 ± 0.20	11.32 ± 0.18	4.88 ± 0.11	11.66 ± 0.23	16.02 ± 0.08
MM + ICT	**6.04 ± 0.05**	6.24 ± 0.08	14.72 ± 0.08	4.40 ± 0.04	4.76 ± 0.04	**11.52 ± 0.95**	**5.07 ± 0.12**	10.71 ± 0.20	13.27 ± 0.50
MM + Tri-Mentor	5.56 ± 0.04	**6.26 ± 0.15**	14.96 ± 0.05	4.36 ± 0.05	4.75 ± 0.06	11.73 ± 0.31	4.65 ± 0.04	9.33 ± 0.11	11.49 ± 0.40
PCD	5.86 ± 0.01	**6.31 ± 0.10**	15.70 ± 0.01	4.31 ± 0.04	4.67 ± 0.02	11.48 ± 0.14	5.05 ± 0.03	11.86 ± 0.13	15.58 ± 0.18
ParetoFlow	5.66 ± 0.07	6.26 ± 0.03	14.63 ± 0.57	4.35 ± 0.03	4.59 ± 0.05	11.71 ± 0.17	4.74 ± 0.14	11.61 ± 0.08	15.07 ± 0.18
ParetoFlow+	5.72 ± 0.02	**6.31 ± 0.04**	15.12 ± 0.36	4.35 ± 0.07	4.62 ± 0.05	11.72 ± 0.15	4.86 ± 0.09	11.73 ± 0.03	15.43 ± 0.23
RFM-AN (ours)	5.80 ± 0.05	**6.16 ± 0.25**	15.57 ± 0.11	4.33 ± 0.10	4.75 ± 0.03	11.76 ± 0.14	**5.12 ± 0.05**	**12.13 ± 0.06**	**16.04 ± 0.11**
RFM-APG (ours)	5.80 ± 0.04	**6.16 ± 0.26**	15.61 ± 0.07	4.35 ± 0.09	4.73 ± 0.03	11.72 ± 0.17	**5.11 ± 0.05**	**12.13 ± 0.07**	**16.06 ± 0.09**
RFM-DAMG (ours)	5.77 ± 0.07	**6.15 ± 0.26**	15.60 ± 0.09	4.34 ± 0.10	4.73 ± 0.02	11.77 ± 0.11	**5.11 ± 0.06**	**12.12 ± 0.07**	**16.01 ± 0.14**
RFM-Cone (ours)	5.80 ± 0.03	**6.27 ± 0.10**	15.65 ± 0.05	4.35 ± 0.08	4.72 ± 0.03	11.72 ± 0.12	**5.11 ± 0.09**	**12.12 ± 0.08**	**16.13 ± 0.06**

Table 18: 75th-percentile hypervolume results for MORL. For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	MO-Hopper	MO-Swimmer
$\mathcal{D}$(best)	–	–
E2E	4.98 ± 0.19	2.55 ± 0.29
E2E + GradNorm	5.20 ± 0.36	3.17 ± 0.30
E2E + PcGrad	4.95 ± 0.21	2.79 ± 0.06
MH	5.35 ± 0.20	2.95 ± 0.14
MH + GradNorm	**5.57 ± 0.46**	2.96 ± 0.09
MH + PcGrad	4.88 ± 0.15	2.93 ± 0.16
MM	5.00 ± 0.19	2.81 ± 0.09
MM + RoMA	**5.30 ± 0.35**	2.83 ± 0.09
MM + IOM	4.97 ± 0.12	2.86 ± 0.09
MM + ICT	5.33 ± 0.08	**3.61 ± 0.07**
MM + Tri-Mentor	5.12 ± 0.22	3.41 ± 0.13
PCD	5.08 ± 0.09	2.84 ± 0.08
ParetoFlow	**5.58 ± 0.03**	2.76 ± 0.49
ParetoFlow+	**5.59 ± 0.03**	2.87 ± 0.27
RFM-AN (ours)	5.06 ± 0.14	2.85 ± 0.15
RFM-APG (ours)	5.12 ± 0.17	2.83 ± 0.08
RFM-DAMG (ours)	5.21 ± 0.15	2.86 ± 0.07
RFM-Cone (ours)	5.05 ± 0.06	2.88 ± 0.14

Table 19: 75th-percentile hypervolume results for RE (Part 1). For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33
$\mathcal{D}$(best)	–	–	–	–	–	–	–	–
E2E	4.25 ± 0.31	**4.75 ± 0.21**	**4.72 ± 0.27**	4.27 ± 0.03	4.23 ± 0.14	**10.61 ± 0.06**	10.63 ± 0.01	10.61 ± 0.01
E2E + GradNorm	4.46 ± 0.07	**4.84 ± 0.00**	4.16 ± 0.03	4.31 ± 0.02	4.55 ± 0.02	9.93 ± 0.03	10.18 ± 0.33	9.30 ± 0.74
E2E + PcGrad	**4.59 ± 0.00**	**4.84 ± 0.00**	**4.84 ± 0.00**	3.97 ± 0.06	4.55 ± 0.06	**10.63 ± 0.03**	10.64 ± 0.00	9.70 ± 0.85
MH	**4.59 ± 0.00**	**4.84 ± 0.00**	**4.84 ± 0.00**	**4.52 ± 0.39**	**4.54 ± 0.67**	**10.65 ± 0.00**	10.63 ± 0.00	10.60 ± 0.01
MH + GradNorm	**4.35 ± 0.33**	**4.40 ± 0.60**	4.75 ± 0.04	**4.21 ± 0.96**	4.82 ± 0.02	**10.49 ± 0.35**	7.66 ± 1.09	10.58 ± 0.02
MH + PcGrad	**4.59 ± 0.00**	**4.73 ± 0.23**	**4.37 ± 0.73**	4.78 ± 0.04	**4.82 ± 0.04**	**10.60 ± 0.05**	10.63 ± 0.00	10.59 ± 0.02
MM	**4.58 ± 0.01**	**4.84 ± 0.00**	**4.84 ± 0.00**	4.42 ± 0.10	4.59 ± 0.22	**10.64 ± 0.01**	**10.64 ± 0.01**	10.58 ± 0.03
MM + RoMA	4.45 ± 0.07	**4.79 ± 0.08**	4.78 ± 0.05	4.69 ± 0.10	**4.82 ± 0.03**	**10.65 ± 0.00**	10.51 ± 0.10	10.57 ± 0.03
MM + IOM	**4.59 ± 0.00**	**4.84 ± 0.00**	4.75 ± 0.03	**4.63 ± 0.25**	**4.83 ± 0.02**	**10.65 ± 0.00**	**10.65 ± 0.00**	10.60 ± 0.01
MM + ICT	**4.59 ± 0.00**	**4.84 ± 0.00**	**4.83 ± 0.01**	**4.66 ± 0.18**	**4.84 ± 0.00**	**10.64 ± 0.01**	**10.65 ± 0.00**	10.59 ± 0.01
MM + Tri-Mentor	**4.59 ± 0.00**	**4.84 ± 0.00**	**4.84 ± 0.00**	**4.84 ± 0.00**	**4.84 ± 0.00**	10.64 ± 0.01	**10.65 ± 0.00**	10.61 ± 0.01
PCD	4.37 ± 0.01	4.83 ± 0.01	4.82 ± 0.00	**4.84 ± 0.00**	4.84 ± 0.00	10.63 ± 0.01	**10.65 ± 0.00**	10.60 ± 0.02
ParetoFlow	4.08 ± 0.07	4.71 ± 0.10	4.75 ± 0.00	4.53 ± 0.03	4.81 ± 0.01	10.38 ± 0.09	10.57 ± 0.07	10.34 ± 0.11
ParetoFlow+	4.10 ± 0.03	4.77 ± 0.04	4.75 ± 0.00	4.55 ± 0.00	4.80 ± 0.00	10.48 ± 0.05	10.57 ± 0.02	10.44 ± 0.04
RFM-AN (ours)	4.49 ± 0.09	**3.73 ± 2.11**	**4.84 ± 0.00**	4.60 ± 0.21	4.83 ± 0.00	6.42 ± 3.08	**10.65 ± 0.00**	**10.63 ± 0.00**
RFM-APG (ours)	**4.50 ± 0.10**	**4.84 ± 0.00**	4.84 ± 0.00	4.60 ± 0.21	4.83 ± 0.00	6.97 ± 3.11	10.64 ± 0.00	**10.63 ± 0.00**
RFM-DAMG (ours)	4.50 ± 0.08	**4.84 ± 0.00**	**4.84 ± 0.00**	4.41 ± 0.13	**4.83 ± 0.01**	5.83 ± 3.16	**10.49 ± 0.36**	**10.62 ± 0.00**
RFM-Cone (ours)	4.49 ± 0.08	**4.67 ± 0.38**	**4.84 ± 0.00**	**4.16 ± 0.89**	**4.83 ± 0.01**	5.82 ± 3.64	**10.49 ± 0.36**	**10.62 ± 0.00**

Table 20: 75th-percentile hypervolume results for RE (Part 2). For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	RE34	RE35	RE36	RE37	RE41	RE42	RE61
$\mathcal{D}$(best)	–	–	–	–	–	–	–
E2E	**10.09 ± 0.01**	10.52 ± 0.02	9.87 ± 0.12	**6.71 ± 0.03**	**20.74 ± 0.10**	**22.27 ± 0.39**	**108.63 ± 0.10**
E2E + GradNorm	**9.57 ± 0.73**	10.38 ± 0.02	6.06 ± 0.68	6.62 ± 0.02	17.41 ± 0.57	21.43 ± 0.20	100.45 ± 3.18
E2E + PcGrad	**10.08 ± 0.02**	**10.58 ± 0.00**	**10.03 ± 0.22**	**6.72 ± 0.01**	**20.79 ± 0.04**	**22.66 ± 0.10**	103.26 ± 1.44
MH	10.08 ± 0.01	**10.55 ± 0.02**	9.93 ± 0.04	**6.70 ± 0.02**	20.62 ± 0.05	22.44 ± 0.08	**108.31 ± 0.47**
MH + GradNorm	9.73 ± 0.03	10.35 ± 0.03	6.54 ± 1.27	6.65 ± 0.02	20.63 ± 0.06	22.13 ± 0.16	105.13 ± 3.20
MH + PcGrad	10.08 ± 0.01	**10.57 ± 0.01**	9.76 ± 0.21	**6.71 ± 0.03**	20.65 ± 0.05	22.47 ± 0.08	107.25 ± 0.11
MM	10.08 ± 0.01	10.52 ± 0.03	9.74 ± 0.20	**6.69 ± 0.04**	20.64 ± 0.03	22.36 ± 0.12	**108.40 ± 0.29**
MM + RoMA	9.95 ± 0.03	10.53 ± 0.03	**8.78 ± 1.44**	6.64 ± 0.03	20.58 ± 0.10	21.05 ± 0.49	108.07 ± 0.45
MM + IOM	10.07 ± 0.01	**10.58 ± 0.00**	**10.09 ± 0.10**	6.66 ± 0.03	20.42 ± 0.06	21.61 ± 0.53	107.32 ± 0.40
MM + ICT	10.05 ± 0.03	10.57 ± 0.00	9.86 ± 0.09	6.66 ± 0.03	20.62 ± 0.04	**22.66 ± 0.07**	108.21 ± 0.18
MM + Tri-Mentor	10.05 ± 0.02	10.35 ± 0.20	**10.11 ± 0.13**	**6.71 ± 0.02**	20.58 ± 0.12	**22.63 ± 0.04**	107.90 ± 0.71
PCD	9.69 ± 0.01	10.39 ± 0.01	8.17 ± 0.05	5.95 ± 0.02	18.65 ± 0.08	15.69 ± 0.04	101.00 ± 0.08
ParetoFlow	9.67 ± 0.05	10.40 ± 0.09	8.33 ± 0.24	5.36 ± 0.42	18.27 ± 0.44	17.33 ± 1.21	104.98 ± 1.53
ParetoFlow+	9.63 ± 0.02	10.31 ± 0.04	7.42 ± 0.42	5.60 ± 0.17	18.17 ± 0.09	16.25 ± 0.36	105.44 ± 1.48
RFM-AN (ours)	9.87 ± 0.03	**10.54 ± 0.04**	7.86 ± 0.38	**6.63 ± 0.14**	20.32 ± 0.09	21.65 ± 0.39	106.47 ± 1.34
RFM-APG (ours)	9.81 ± 0.14	10.53 ± 0.04	8.63 ± 0.84	6.70 ± 0.01	20.41 ± 0.06	22.08 ± 0.25	107.18 ± 0.59
RFM-DAMG (ours)	9.85 ± 0.03	10.53 ± 0.04	8.05 ± 0.59	**6.68 ± 0.04**	20.42 ± 0.04	21.81 ± 0.38	106.47 ± 1.12
RFM-Cone (ours)	9.88 ± 0.02	10.52 ± 0.04	8.10 ± 0.58	**6.57 ± 0.14**	20.54 ± 0.07	21.94 ± 0.33	106.55 ± 1.52

D.3 50TH PERCENTILE RESULTS

Tables 21–27 report the 50th-percentile hypervolume for DTLZ (Table 21), ZDT (Table 22), C-10/MOP (Table 23), IN-1K/MOP (Table 24), MORL (Table 25), and RE (Tables 26 and 27). $\mathcal{D}$(best) is reported only at the 100th percentile.

Table 21: 50th-percentile hypervolume results for DTLZ. For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
$\mathcal{D}$(best)	–	–	–	–	–	–	–
E2E	**10.64 ± 0.00**	12.43 ± 0.01	9.73 ± 0.12	**17.61 ± 0.03**	**21.05 ± 0.01**	**15.70 ± 0.02**	**10.79 ± 0.00**
E2E + GradNorm	10.62 ± 0.01	**12.40 ± 0.08**	9.83 ± 0.02	17.16 ± 0.30	**20.83 ± 0.40**	15.16 ± 0.22	10.59 ± 0.08
E2E + PcGrad	10.62 ± 0.01	**12.44 ± 0.00**	9.85 ± 0.02	17.25 ± 0.17	**21.06 ± 0.00**	15.62 ± 0.02	10.61 ± 0.08
MH	10.62 ± 0.01	**12.44 ± 0.01**	9.84 ± 0.03	**17.38 ± 0.42**	**21.03 ± 0.04**	14.31 ± 0.49	10.77 ± 0.02
MH + GradNorm	10.63 ± 0.01	12.42 ± 0.01	9.80 ± 0.06	17.25 ± 0.14	20.83 ± 0.16	15.46 ± 0.06	10.69 ± 0.03
MH + PcGrad	10.62 ± 0.01	**12.44 ± 0.00**	9.83 ± 0.01	16.98 ± 0.53	**21.04 ± 0.03**	15.49 ± 0.03	10.64 ± 0.09
MM	10.63 ± 0.01	12.43 ± 0.01	9.83 ± 0.03	17.50 ± 0.06	**21.05 ± 0.02**	15.57 ± 0.05	10.77 ± 0.01
MM + RoMA	10.61 ± 0.01	12.40 ± 0.01	9.78 ± 0.04	16.85 ± 0.16	20.86 ± 0.09	15.40 ± 0.05	10.09 ± 0.09
MM + IOM	10.62 ± 0.01	12.44 ± 0.00	**9.87 ± 0.01**	16.36 ± 0.71	21.05 ± 0.00	15.47 ± 0.01	10.13 ± 0.14
MM + ICT	10.62 ± 0.01	12.34 ± 0.03	9.85 ± 0.01	16.84 ± 0.17	20.62 ± 0.09	15.32 ± 0.08	10.45 ± 0.16
MM + Tri-Mentor	10.62 ± 0.01	12.42 ± 0.01	8.54 ± 0.16	17.03 ± 0.41	20.90 ± 0.07	15.51 ± 0.03	10.08 ± 0.31
PCD	10.46 ± 0.02	12.35 ± 0.01	9.74 ± 0.08	16.29 ± 0.15	20.65 ± 0.05	15.35 ± 0.03	9.14 ± 0.05
ParetoFlow	10.42 ± 0.15	12.17 ± 0.17	9.36 ± 0.23	14.20 ± 0.45	20.33 ± 0.39	15.19 ± 0.17	8.58 ± 0.14
ParetoFlow+	10.43 ± 0.03	12.06 ± 0.16	9.23 ± 0.35	14.00 ± 0.29	20.23 ± 0.18	14.75 ± 0.51	8.79 ± 0.06
RFM-AN (ours)	10.58 ± 0.02	8.66 ± 1.73	9.81 ± 0.01	13.44 ± 1.81	14.36 ± 0.28	14.97 ± 0.44	9.70 ± 0.37
RFM-APG (ours)	10.58 ± 0.02	9.20 ± 1.84	9.81 ± 0.01	14.32 ± 1.96	14.39 ± 0.26	15.01 ± 0.47	9.81 ± 0.42
RFM-DAMG (ours)	10.58 ± 0.02	8.43 ± 1.24	9.82 ± 0.01	13.68 ± 1.82	14.38 ± 0.23	15.02 ± 0.46	9.90 ± 0.39
RFM-Cone (ours)	10.58 ± 0.02	8.80 ± 1.70	9.81 ± 0.01	15.02 ± 1.22	14.65 ± 0.32	15.15 ± 0.37	9.91 ± 0.31

Table 22: 50th-percentile hypervolume results for ZDT. For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
$\mathcal{D}$(best)	–	–	–	–	–
E2E	**4.86 ± 0.00**	**5.70 ± 0.00**	**5.18 ± 0.64**	4.40 ± 0.32	**4.78 ± 0.01**
E2E + GradNorm	3.08 ± 0.10	5.07 ± 0.61	4.21 ± 0.07	4.65 ± 0.02	3.02 ± 0.90
E2E + PcGrad	4.86 ± 0.00	**5.70 ± 0.00**	5.06 ± 0.50	3.80 ± 0.20	1.82 ± 0.22
MH	4.84 ± 0.01	5.52 ± 0.11	**5.68 ± 0.07**	4.72 ± 0.11	4.75 ± 0.00
MH + GradNorm	**4.49 ± 0.79**	5.56 ± 0.10	4.94 ± 0.26	3.22 ± 0.13	3.21 ± 0.99
MH + PcGrad	4.82 ± 0.01	5.39 ± 0.01	5.49 ± 0.07	3.43 ± 0.34	**4.12 ± 0.85**
MM	4.84 ± 0.01	5.55 ± 0.09	**5.67 ± 0.10**	4.73 ± 0.19	4.75 ± 0.00
MM + RoMA	4.84 ± 0.00	5.50 ± 0.05	**5.57 ± 0.14**	**4.62 ± 0.49**	1.78 ± 0.06
MM + IOM	4.34 ± 0.07	4.89 ± 0.02	5.40 ± 0.05	**5.06 ± 0.14**	4.69 ± 0.05
MM + ICT	4.84 ± 0.00	5.53 ± 0.10	5.33 ± 0.14	3.26 ± 0.12	2.76 ± 1.05
MM + Tri-Mentor	4.43 ± 0.02	5.29 ± 0.05	5.18 ± 0.09	4.95 ± 0.08	2.27 ± 0.06
PCD	4.20 ± 0.01	4.77 ± 0.03	5.13 ± 0.02	4.72 ± 0.01	4.33 ± 0.01
ParetoFlow	4.13 ± 0.05	4.67 ± 0.06	5.08 ± 0.09	4.66 ± 0.13	4.44 ± 0.08
ParetoFlow+	4.17 ± 0.03	4.67 ± 0.03	5.13 ± 0.05	**5.08 ± 0.11**	4.48 ± 0.07
RFM-AN (ours)	4.46 ± 0.04	4.95 ± 0.07	5.51 ± 0.04	3.96 ± 0.12	2.44 ± 0.16
RFM-APG (ours)	4.50 ± 0.06	5.03 ± 0.03	5.50 ± 0.04	3.96 ± 0.13	3.01 ± 0.23
RFM-DAMG (ours)	4.49 ± 0.04	5.05 ± 0.03	5.52 ± 0.04	3.95 ± 0.11	2.82 ± 0.39
RFM-Cone (ours)	4.52 ± 0.06	5.05 ± 0.02	5.58 ± 0.07	3.99 ± 0.13	2.89 ± 0.52

Table 23: 50th-percentile hypervolume results for MO-NAS (Part 1, C-10/MOP). For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	C-10/MOP1	C-10/MOP2	C-10/MOP3	C-10/MOP4	C-10/MOP5	C-10/MOP6	C-10/MOP7	C-10/MOP8	C-10/MOP9
$\mathcal{D}$(best)	–	–	–	–	–	–	–	–	–
E2E	0.00 ± 0.00	0.00 ± 0.00	10.10 ± 0.25	21.02 ± 0.32	38.49 ± 2.38	99.40 ± 2.67	400.60 ± 24.57	5.72 ± 0.05	**15.22 ± 0.22**
E2E + GradNorm	1.20 ± 0.10	1.12 ± 0.08	9.66 ± 0.14	20.43 ± 0.69	37.02 ± 2.45	94.45 ± 4.46	377.97 ± 19.27	4.88 ± 0.39	**13.50 ± 1.79**
E2E + PcGrad	0.66 ± 0.61	0.26 ± 0.57	9.99 ± 0.17	20.50 ± 0.49	40.11 ± 2.06	97.21 ± 2.04	396.72 ± 30.39	**5.72 ± 0.06**	14.41 ± 0.32
MH	1.10 ± 0.04	0.95 ± 0.16	9.74 ± 0.31	19.56 ± 1.18	37.19 ± 0.65	99.48 ± 2.29	409.60 ± 20.37	**5.74 ± 0.07**	**14.87 ± 0.50**
MH + GradNorm	0.77 ± 0.55	**0.90 ± 0.50**	9.64 ± 0.53	20.41 ± 0.55	35.25 ± 3.28	94.56 ± 1.45	382.77 ± 13.02	5.17 ± 0.27	10.98 ± 1.23
MH + PcGrad	1.19 ± 0.14	1.09 ± 0.03	9.91 ± 0.33	20.47 ± 0.45	37.77 ± 0.76	95.53 ± 3.81	375.68 ± 7.65	5.67 ± 0.07	15.00 ± 0.18
MM	0.49 ± 0.67	0.52 ± 0.49	9.85 ± 0.36	20.65 ± 0.45	39.32 ± 1.57	100.55 ± 2.25	409.69 ± 20.16	5.73 ± 0.04	**14.94 ± 0.34**
MM + RoMA	1.15 ± 0.23	**0.92 ± 0.52**	9.69 ± 0.34	20.10 ± 0.27	40.86 ± 1.60	96.60 ± 2.73	435.07 ± 18.40	5.66 ± 0.06	14.65 ± 0.44
MM + IOM	1.40 ± 0.05	1.28 ± 0.06	7.12 ± 1.07	18.69 ± 1.47	**44.70 ± 1.75**	**101.71 ± 7.19**	405.89 ± 46.84	5.43 ± 0.11	**14.87 ± 0.58**
MM + ICT	1.17 ± 0.03	0.64 ± 0.58	9.72 ± 0.34	19.05 ± 0.98	43.73 ± 1.69	**103.68 ± 2.55**	406.07 ± 26.13	**5.77 ± 0.04**	14.55 ± 0.54
MM + Tri-Mentor	1.16 ± 0.04	1.30 ± 0.04	9.86 ± 0.63	20.10 ± 1.14	41.97 ± 0.33	99.88 ± 4.36	454.85 ± 10.03	5.70 ± 0.04	**15.06 ± 0.24**
PCD	1.45 ± 0.01	1.33 ± 0.02	10.21 ± 0.29	18.51 ± 1.80	40.37 ± 0.61	104.66 ± 0.36	451.35 ± 10.73	5.53 ± 0.02	14.49 ± 0.13
ParetoFlow	**1.46 ± 0.02**	1.31 ± 0.03	9.31 ± 0.37	19.55 ± 0.57	41.20 ± 3.55	**104.22 ± 1.98**	423.84 ± 12.43	5.29 ± 0.16	14.53 ± 0.12
ParetoFlow+	**1.46 ± 0.00**	1.36 ± 0.01	10.00 ± 0.15	20.07 ± 0.42	40.84 ± 0.77	**105.33 ± 1.78**	402.94 ± 18.31	5.45 ± 0.16	14.72 ± 0.08
RFM-AN (ours)	1.46 ± 0.00	**1.38 ± 0.01**	**10.98 ± 0.06**	**22.92 ± 0.79**	**45.57 ± 2.02**	**103.15 ± 3.89**	470.91 ± 4.14	5.50 ± 0.05	13.98 ± 0.10
RFM-APG (ours)	**1.43 ± 0.05**	**1.38 ± 0.01**	**10.94 ± 0.12**	**23.12 ± 0.77**	**45.53 ± 1.95**	**102.48 ± 3.56**	**471.76 ± 7.16**	5.49 ± 0.07	13.99 ± 0.15
RFM-DAMG (ours)	**1.43 ± 0.05**	**1.38 ± 0.00**	**10.94 ± 0.11**	**22.92 ± 0.81**	**45.55 ± 1.97**	**103.56 ± 3.85**	**470.85 ± 6.44**	5.48 ± 0.06	13.94 ± 0.19
RFM-Cone (ours)	1.41 ± 0.05	**1.38 ± 0.01**	**10.93 ± 0.12**	**23.06 ± 1.11**	**45.64 ± 2.26**	**102.19 ± 4.04**	**476.88 ± 12.48**	5.56 ± 0.04	14.41 ± 0.14

Table 24: 50th-percentile hypervolume results for MO-NAS (Part 2, IN-1K/MOP). For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	IN-1K/MOP1	IN-1K/MOP2	IN-1K/MOP3	IN-1K/MOP4	IN-1K/MOP5	IN-1K/MOP6	IN-1K/MOP7	IN-1K/MOP8	IN-1K/MOP9
$\mathcal{D}$(best)	–	–	–	–	–	–	–	–	–
E2E	5.49 ± 0.24	5.63 ± 0.30	13.84 ± 0.20	**4.44 ± 0.21**	4.64 ± 0.19	9.89 ± 0.71	4.58 ± 0.14	9.03 ± 0.45	10.89 ± 0.83
E2E + GradNorm	5.22 ± 0.03	6.13 ± 0.08	12.91 ± 0.30	3.96 ± 0.11	4.28 ± 0.10	9.42 ± 0.32	4.39 ± 0.36	8.24 ± 0.23	10.06 ± 0.74
E2E + PcGrad	5.55 ± 0.24	6.17 ± 0.05	13.94 ± 0.34	4.27 ± 0.14	4.43 ± 0.25	10.64 ± 1.21	4.75 ± 0.12	9.42 ± 0.22	11.49 ± 0.09
MH	5.51 ± 0.07	5.79 ± 0.22	13.37 ± 0.10	4.30 ± 0.19	4.46 ± 0.21	9.44 ± 0.30	4.63 ± 0.10	9.21 ± 0.09	11.74 ± 0.42
MH + GradNorm	5.37 ± 0.25	6.15 ± 0.05	13.40 ± 0.48	3.19 ± 0.37	**4.38 ± 0.49**	8.28 ± 0.19	3.65 ± 0.33	8.28 ± 0.88	13.28 ± 2.17
MH + PcGrad	5.30 ± 0.05	**6.19 ± 0.11**	13.43 ± 0.24	4.11 ± 0.26	4.31 ± 0.24	10.52 ± 0.93	4.72 ± 0.09	9.27 ± 0.16	11.04 ± 0.48
MM	5.59 ± 0.15	5.80 ± 0.21	13.74 ± 0.17	**4.55 ± 0.06**	**4.85 ± 0.08**	**12.18 ± 0.23**	4.81 ± 0.08	9.57 ± 0.23	11.84 ± 0.35
MM + RoMA	5.48 ± 0.12	**5.83 ± 0.53**	13.45 ± 0.14	4.04 ± 0.21	4.23 ± 0.22	11.87 ± 0.29	4.82 ± 0.08	9.19 ± 0.14	10.69 ± 0.87
MM + IOM	**5.96 ± 0.04**	**6.17 ± 0.17**	**15.60 ± 0.11**	4.41 ± 0.06	4.38 ± 0.46	11.11 ± 0.05	4.68 ± 0.22	10.57 ± 0.40	13.47 ± 1.61
MM + ICT	**5.96 ± 0.05**	**6.13 ± 0.20**	14.45 ± 0.25	4.28 ± 0.17	4.71 ± 0.06	11.01 ± 1.11	4.88 ± 0.11	10.26 ± 0.06	12.71 ± 0.52
MM + Tri-Mentor	5.49 ± 0.04	**6.07 ± 0.29**	14.56 ± 0.33	4.28 ± 0.04	4.66 ± 0.05	11.63 ± 0.27	4.58 ± 0.07	9.10 ± 0.28	10.62 ± 0.41
PCD	5.70 ± 0.02	6.15 ± 0.05	15.38 ± 0.20	4.19 ± 0.04	4.57 ± 0.03	11.10 ± 0.15	4.79 ± 0.13	11.37 ± 0.16	14.77 ± 0.49
ParetoFlow	5.64 ± 0.10	**6.22 ± 0.09**	14.48 ± 0.67	4.21 ± 0.13	4.57 ± 0.04	11.48 ± 0.14	4.57 ± 0.10	11.27 ± 0.25	14.75 ± 0.15
ParetoFlow+	5.69 ± 0.03	**6.28 ± 0.07**	15.07 ± 0.36	4.31 ± 0.06	4.59 ± 0.04	11.51 ± 0.03	4.80 ± 0.11	11.63 ± 0.06	15.23 ± 0.41
RFM-AN (ours)	5.70 ± 0.08	**6.07 ± 0.32**	**15.53 ± 0.11**	4.28 ± 0.09	4.69 ± 0.02	11.37 ± 0.27	**5.07 ± 0.06**	**12.07 ± 0.06**	**15.97 ± 0.08**
RFM-APG (ours)	5.69 ± 0.08	**6.11 ± 0.24**	**15.55 ± 0.09**	4.29 ± 0.09	4.68 ± 0.02	11.36 ± 0.24	**5.02 ± 0.09**	**12.08 ± 0.08**	**15.96 ± 0.16**
RFM-DAMG (ours)	5.70 ± 0.09	**6.08 ± 0.33**	**15.54 ± 0.11**	4.29 ± 0.09	4.68 ± 0.02	11.39 ± 0.20	**5.03 ± 0.08**	**12.02 ± 0.20**	**15.90 ± 0.11**
RFM-Cone (ours)	5.69 ± 0.04	**6.10 ± 0.19**	**15.62 ± 0.05**	4.30 ± 0.09	4.67 ± 0.02	11.46 ± 0.17	**5.02 ± 0.12**	**12.08 ± 0.07**	**15.95 ± 0.22**

Table 25: 50th-percentile hypervolume results for MORL. For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	MO-Hopper	MO-Swimmer
$\mathcal{D}$(best)	–	–
E2E	4.87 ± 0.13	2.20 ± 0.43
E2E + GradNorm	4.93 ± 0.27	2.89 ± 0.14
E2E + PcGrad	4.85 ± 0.21	2.58 ± 0.06
MH	5.00 ± 0.15	2.68 ± 0.11
MH + GradNorm	$\mathbf{5.41 \pm 0.39}$	2.66 ± 0.18
MH + PcGrad	4.77 ± 0.03	2.67 ± 0.11
MM	4.82 ± 0.04	2.57 ± 0.10
MM + RoMA	4.97 ± 0.24	2.58 ± 0.17
MM + IOM	4.83 ± 0.06	2.62 ± 0.10
MM + ICT	$\underline{5.14 \pm 0.04}$	$\mathbf{3.33 \pm 0.18}$
MM + Tri-Mentor	4.86 ± 0.19	$\underline{3.06 \pm 0.15}$
PCD	4.84 ± 0.04	2.45 ± 0.10
ParetoFlow	$\mathbf{5.23 \pm 0.44}$	2.34 ± 0.45
ParetoFlow+	5.05 ± 0.31	2.35 ± 0.33
RFM-AN (ours)	4.84 ± 0.04	2.36 ± 0.09
RFM-APG (ours)	4.83 ± 0.03	2.40 ± 0.15
RFM-DAMG (ours)	4.84 ± 0.04	2.36 ± 0.11
RFM-Cone (ours)	4.84 ± 0.04	2.34 ± 0.05

Table 26: 50th-percentile hypervolume results for RE (Part 1). For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	RE21	RE22	RE23	RE24	RE25	RE31	RE32	RE33
$\mathcal{D}$(best)	–	–	–	–	–	–	–	–
E2E	4.25 ± 0.31	4.46 ± 0.21	$\mathbf{4.71 \pm 0.27}$	4.27 ± 0.03	4.14 ± 0.28	$\mathbf{10.56 \pm 0.12}$	10.63 ± 0.01	10.60 ± 0.01
E2E + GradNorm	$\mathbf{4.33 \pm 0.28}$	$\mathbf{4.82 \pm 0.02}$	3.74 ± 0.05	4.20 ± 0.03	4.36 ± 0.05	9.78 ± 0.07	9.84 ± 0.39	8.19 ± 0.84
E2E + PcGrad	$\mathbf{4.59 \pm 0.00}$	$\mathbf{4.84 \pm 0.00}$	$\mathbf{4.84 \pm 0.00}$	3.53 ± 0.05	4.27 ± 0.21	$\mathbf{10.39 \pm 0.55}$	$\mathbf{10.64 \pm 0.00}$	7.85 ± 0.42
MH	$\mathbf{4.59 \pm 0.00}$	$\mathbf{4.77 \pm 0.15}$	$\mathbf{4.84 \pm 0.00}$	4.46 ± 0.36	$\mathbf{4.29 \pm 0.78}$	$\mathbf{10.65 \pm 0.00}$	$\underline{10.63 \pm 0.00}$	10.58 ± 0.03
MH + GradNorm	$\mathbf{4.01 \pm 0.92}$	$\mathbf{4.02 \pm 0.87}$	4.09 ± 0.14	3.39 ± 0.53	4.58 ± 0.24	$\mathbf{10.39 \pm 0.57}$	7.41 ± 0.94	$\mathbf{9.50 \pm 1.95}$
MH + PcGrad	$\mathbf{4.59 \pm 0.00}$	$\mathbf{4.61 \pm 0.34}$	4.09 ± 0.61	4.15 ± 0.37	$\mathbf{4.69 \pm 0.16}$	10.50 ± 0.15	10.39 ± 0.21	10.54 ± 0.08
MM	$\underline{4.57 \pm 0.02}$	$\mathbf{4.75 \pm 0.21}$	$\mathbf{4.84 \pm 0.00}$	4.41 ± 0.10	4.56 ± 0.20	10.49 ± 0.15	$\mathbf{10.64 \pm 0.01}$	10.57 ± 0.03
MM + RoMA	4.39 ± 0.09	4.36 ± 0.46	4.62 ± 0.10	$\underline{4.69 \pm 0.10}$	4.79 ± 0.03	$\mathbf{10.63 \pm 0.02}$	10.36 ± 0.05	10.21 ± 0.30
MM + IOM	$\mathbf{4.58 \pm 0.00}$	$\mathbf{4.84 \pm 0.00}$	4.70 ± 0.04	4.59 ± 0.24	$\mathbf{4.82 \pm 0.02}$	$\mathbf{10.65 \pm 0.00}$	$\mathbf{10.64 \pm 0.01}$	10.58 ± 0.03
MM + ICT	$\mathbf{4.59 \pm 0.00}$	$\mathbf{4.83 \pm 0.01}$	$\mathbf{4.83 \pm 0.01}$	4.30 ± 0.30	$\mathbf{4.84 \pm 0.00}$	$\mathbf{10.63 \pm 0.02}$	$\mathbf{10.53 \pm 0.26}$	10.55 ± 0.03
MM + Tri-Mentor	$\mathbf{4.59 \pm 0.00}$	$\mathbf{4.84 \pm 0.00}$	$\mathbf{4.84 \pm 0.00}$	$\mathbf{4.77 \pm 0.15}$	$\mathbf{4.84 \pm 0.00}$	$\underline{10.59 \pm 0.03}$	$\mathbf{10.64 \pm 0.01}$	10.60 ± 0.02
PCD	4.30 ± 0.02	4.75 ± 0.04	4.81 ± 0.00	$\mathbf{4.84 \pm 0.00}$	4.83 ± 0.00	10.58 ± 0.02	$\mathbf{10.64 \pm 0.00}$	$\underline{10.60 \pm 0.01}$
ParetoFlow	4.08 ± 0.07	4.55 ± 0.13	4.69 ± 0.01	4.46 ± 0.03	4.81 ± 0.01	10.29 ± 0.04	10.55 ± 0.08	10.33 ± 0.11
ParetoFlow+	4.08 ± 0.04	4.74 ± 0.04	4.71 ± 0.03	4.52 ± 0.01	4.79 ± 0.01	10.45 ± 0.06	10.54 ± 0.03	10.43 ± 0.05
RFM-AN (ours)	4.43 ± 0.15	$\mathbf{3.56 \pm 2.10}$	4.83 ± 0.01	4.60 ± 0.21	4.82 ± 0.00	6.42 ± 3.08	$\mathbf{10.48 \pm 0.37}$	$\mathbf{10.61 \pm 0.02}$
RFM-APG (ours)	$\mathbf{4.50 \pm 0.10}$	$\mathbf{4.84 \pm 0.00}$	4.84 ± 0.00	4.60 ± 0.21	4.83 ± 0.01	6.49 ± 2.85	$\mathbf{10.44 \pm 0.40}$	$\mathbf{10.61 \pm 0.03}$
RFM-DAMG (ours)	$\mathbf{4.46 \pm 0.16}$	$\mathbf{4.83 \pm 0.01}$	4.84 ± 0.00	4.41 ± 0.13	4.83 ± 0.00	5.83 ± 3.16	$\mathbf{10.30 \pm 0.47}$	$\mathbf{10.61 \pm 0.02}$
RFM-Cone (ours)	4.43 ± 0.13	$\mathbf{4.49 \pm 0.78}$	$\underline{4.84 \pm 0.00}$	$\mathbf{4.16 \pm 0.89}$	4.82 ± 0.00	5.82 ± 3.64	$\mathbf{10.29 \pm 0.48}$	$\mathbf{10.62 \pm 0.01}$

Table 27: 50th-percentile hypervolume results for RE (Part 2). For each task, methods within one standard deviation (or 5×10^{-4}) of the best are bolded and the best of the remaining methods is underlined; the shaded row marks the RFM variant with the most task-best results overall.

Methods	RE34	RE35	RE36	RE37	RE41	RE42	RE61
$\mathcal{D}$(best)	–	–	–	–	–	–	–
E2E	**10.07 ± 0.02**	10.49 ± 0.03	9.66 ± 0.17	**6.68 ± 0.04**	20.58 ± 0.11	21.83 ± 0.58	**108.30 ± 0.18**
E2E + GradNorm	7.21 ± 1.27	10.37 ± 0.01	5.42 ± 0.03	6.59 ± 0.02	16.90 ± 1.03	19.65 ± 2.26	96.02 ± 3.08
E2E + PcGrad	**10.07 ± 0.02**	**10.57 ± 0.00**	9.65 ± 0.26	**6.69 ± 0.01**	**20.69 ± 0.05**	**22.55 ± 0.15**	101.10 ± 2.70
MH	**10.06 ± 0.01**	**10.55 ± 0.02**	9.74 ± 0.09	**6.67 ± 0.03**	20.48 ± 0.07	22.35 ± 0.12	**107.94 ± 0.65**
MH + GradNorm	8.69 ± 0.80	10.06 ± 0.32	4.96 ± 1.15	6.62 ± 0.03	20.48 ± 0.10	21.47 ± 0.14	**104.36 ± 4.19**
MH + PcGrad	10.06 ± 0.01	**10.47 ± 0.20**	9.26 ± 0.16	**6.67 ± 0.03**	20.48 ± 0.08	**22.17 ± 0.41**	106.83 ± 0.24
MM	**10.05 ± 0.02**	10.51 ± 0.04	9.54 ± 0.13	6.65 ± 0.04	20.52 ± 0.07	22.22 ± 0.15	107.73 ± 0.49
MM + RoMA	9.56 ± 0.16	10.27 ± 0.17	7.00 ± 1.33	6.62 ± 0.04	20.39 ± 0.09	19.84 ± 0.57	107.46 ± 0.72
MM + IOM	10.05 ± 0.01	10.49 ± 0.05	**9.91 ± 0.14**	6.62 ± 0.03	20.28 ± 0.12	20.87 ± 0.42	106.23 ± 0.72
MM + ICT	10.03 ± 0.03	10.57 ± 0.00	9.10 ± 0.20	6.64 ± 0.04	20.51 ± 0.05	**22.53 ± 0.14**	107.74 ± 0.32
MM + Tri-Mentor	10.04 ± 0.02	10.30 ± 0.21	**9.62 ± 0.38**	**6.69 ± 0.02**	20.46 ± 0.14	**22.53 ± 0.03**	107.42 ± 0.58
PCD	9.51 ± 0.05	10.35 ± 0.01	7.91 ± 0.14	5.88 ± 0.06	18.51 ± 0.03	15.41 ± 0.03	100.67 ± 0.19
ParetoFlow	9.42 ± 0.21	10.33 ± 0.13	7.89 ± 0.26	5.27 ± 0.44	18.16 ± 0.46	16.44 ± 1.10	104.54 ± 1.57
ParetoFlow+	9.24 ± 0.10	10.29 ± 0.04	6.91 ± 0.31	5.45 ± 0.10	18.07 ± 0.10	16.06 ± 0.42	104.95 ± 1.69
RFM-AN (ours)	9.82 ± 0.02	10.50 ± 0.04	5.44 ± 0.12	**6.50 ± 0.25**	20.19 ± 0.09	21.01 ± 0.28	106.35 ± 1.22
RFM-APG (ours)	9.77 ± 0.12	10.50 ± 0.04	7.38 ± 1.25	6.65 ± 0.03	20.29 ± 0.09	21.26 ± 0.20	107.02 ± 0.59
RFM-DAMG (ours)	9.82 ± 0.01	10.50 ± 0.04	5.79 ± 0.78	6.64 ± 0.05	20.31 ± 0.09	21.15 ± 0.44	106.36 ± 1.10
RFM-Cone (ours)	9.85 ± 0.04	10.49 ± 0.03	5.47 ± 0.15	**6.46 ± 0.25**	20.39 ± 0.08	21.25 ± 0.32	106.44 ± 1.48

Table 3: Task metadata and hypervolume reference points for all 47 tasks. Reference points are in normalized maximization coordinates (negated from the evaluator's minimization convention). d is the number of design variables (categorical variables for NAS, whose logits have dimension D), $|\mathcal{D}|$ the offline dataset size, and `Cont.`/`Cat.` continuous/categorical.

Task	d	m	$\lvert\mathcal{D}\rvert$	Space	Reference point $\boldsymbol{r}$
DTLZ					
DTLZ1	7	3	60,000	Cont.	$(-2.2,\ -2.2,\ -2.2)$
DTLZ2	10	3	60,000	Cont.	$(-2.07826,\ -2.44443,\ -2.45812)$
DTLZ3	10	3	60,000	Cont.	$(-2.22594,\ -2.07323,\ -2.1453)$
DTLZ4	10	3	60,000	Cont.	$(-2.29461,\ -2.91967,\ -2.63431)$
DTLZ5	10	3	60,000	Cont.	$(-2.74136,\ -2.99806,\ -2.6272)$
DTLZ6	10	3	60,000	Cont.	$(-2.5467,\ -2.53751,\ -2.46451)$
DTLZ7	10	3	60,000	Cont.	$(-2.20001,\ -2.20001,\ -2.2)$
ZDT					
ZDT1	30	2	60,000	Cont.	$(-2.2,\ -2.2)$
ZDT2	30	2	60,000	Cont.	$(-2.2,\ -2.46332)$
ZDT3	30	2	60,000	Cont.	$(-2.2,\ -2.70741)$
ZDT4	10	2	60,000	Cont.	$(-2.20018,\ -2.52935)$
ZDT6	10	2	60,000	Cont.	$(-2.2,\ -2.21336)$
C-10/MOP					
C-10/MOP1	26	2	12,084	Cat.	$(-0.700653,\ -2.27345)$
C-10/MOP2	26	3	26,316	Cat.	$(-0.270857,\ -2.33973,\ -2.3436)$
C-10/MOP3	5	3	19,661	Cat.	$(-2.2,\ -2.2,\ -2.2)$
C-10/MOP4	5	4	19,661	Cat.	$(-2.2,\ -2.2,\ -2.2,\ -2.2)$
C-10/MOP5	6	5	9,375	Cat.	$(-2.2,\ \ldots,\ -2.2)$
C-10/MOP6	6	6	9,375	Cat.	$(-2.2,\ \ldots,\ -2.2)$
C-10/MOP7	6	8	9,375	Cat.	$(-2.2,\ \ldots,\ -2.2)$
C-10/MOP8	32	2	60,000	Cat.	$(-2.47149,\ -2.42687)$
C-10/MOP9	32	3	60,000	Cat.	$(-2.46312,\ -2.57324,\ -2.5169)$
IN-1K/MOP					
IN-1K/MOP1	25	2	60,000	Cat.	$(-2.85067,\ -2.46423)$
IN-1K/MOP2	25	2	60,000	Cat.	$(-2.8185,\ -2.77927)$
IN-1K/MOP3	25	3	60,000	Cat.	$(-2.82779,\ -2.57366,\ -2.5582)$
IN-1K/MOP4	34	2	60,000	Cat.	$(-2.13485,\ -2.31778)$
IN-1K/MOP5	34	2	60,000	Cat.	$(-2.14702,\ -2.39576)$
IN-1K/MOP6	34	3	60,000	Cat.	$(-2.21115,\ -2.43552,\ -2.44032)$
IN-1K/MOP7	21	2	60,000	Cat.	$(-2.97769,\ -2.29048)$
IN-1K/MOP8	21	3	60,000	Cat.	$(-2.95398,\ -2.26424,\ -2.27795)$
IN-1K/MOP9	21	4	60,000	Cat.	$(-2.2,\ -2.2,\ -2.2,\ -2.2)$
MORL					
MO-Hopper	10,184	2	4,500	Cont.	$(-2.45514,\ -3.41206)$
MO-Swimmer	9,734	2	8,571	Cont.	$(-2.40532,\ -1.76713)$
RE					
RE21	4	2	60,000	Cont.	$(-2.2,\ -2.2)$
RE22	3	2	60,000	Cont.	$(-2.2,\ -2.2)$
RE23	4	2	60,000	Cont.	$(-2.2,\ -2.20012)$
RE24	2	2	60,000	Cont.	$(-2.2,\ -2.2)$
RE25	3	2	60,000	Cont.	$(-2.2,\ -2.2)$
RE31	3	3	60,000	Cont.	$(-2.2,\ -2.2,\ -2.2)$
RE32	4	3	60,000	Cont.	$(-2.2,\ -2.2,\ -2.2)$
RE33	4	3	60,000	Cont.	$(-2.2,\ -2.2,\ -2.2)$
RE34	5	3	60,000	Cont.	$(-2.2,\ -2.2,\ -2.2)$
RE35	7	3	60,000	Cont.	$(-2.2,\ -2.2,\ -2.2)$
RE36	4	3	60,000	Cont.	$(-2.2,\ -2.2,\ -2.2)$
RE37	4	3	60,000	Cont.	$(-2.17549,\ -1.69879,\ -2.04536)$
RE41	7	4	60,000	Cont.	$(-2.2,\ -2.2,\ -2.2,\ -2.2)$
RE42	6	4	60,000	Cont.	$(-2.2,\ -2.2,\ -2.2,\ -2.2)$
RE61	3	6	60,000	Cont.	$(-2.2,\ \ldots,\ -2.2)$