TWO-TIME-SCALE WEIGHT ADAPTATION AND EVOLUTIONARY GENOME SEARCH FOR PARETO FRONT COVERAGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Many problems in science and engineering require minimising several conflicting objectives at once. In settings where gradients of the objectives are available, e.g. multi-task learning, preference alignment, and adjoint-based design, scalarisation reduces the problem to a gradient-friendly family of weighted sub-problems. However, it shifts the problem onto weight selection: the weight distribution that spreads solutions evenly over the Pareto front is governed by the solution map that is non-injective and often discontinuous. We propose TWARD (Two-time-scale Weight Adaptation and Repulsion Descent), a two-time-scale method for gradient-based Multi-objective optimisation. On the fast scale, the decision variables descend an entropic reference-point scalarisation, a smoothed Tchebycheff achievement function whose reference point is parametrised by the log of the weights, with a parameter τ which controls coverage. On the slow scale, entropic mirror descent moves the weights on the simplex under a repulsion measured between solutions in objective space, promoting spreading of solutions on the Pareto front, translated into a weight update in closed form by inverting the scalarisation rather than the solution map. Where a single gradient flow is confined to one connected piece of the front, an evolutionary outer loop varies the m-dimensional scalarisation genome (λ, τ). At a matched evaluation budget, we see superior performance compared to established baseline approaches over a wide range of synthetic and real-world tasks.

1 INTRODUCTION

Multi-objective optimization (MOO) asks for the simultaneous minimization of m conflicting objectives $f = (f_1, \ldots, f_m) : \mathbb{R}^d \to \mathbb{R}^m$. No single point minimizes all of them, so the goal is the *Pareto front* $\mathcal{PF}$, the image of the non-dominated set, and in practice a finite set of solutions that *covers* it evenly.[1] A standard approach is *scalarization* (Miettinen, 1999), which replaces the vector problem by a parametrized family of scalar sub-problems $\min_x S_\lambda(x)$ indexed by a weight $\lambda \in \Delta_{m-1}$ (Section 3), known as a preference. When the objectives are differentiable this is fast and amenable to gradient based methods.

However the selection of weights for the scalarised objective is challenging. Sampling $\lambda \sim q$ and pushing it through the solution map $\Phi(\lambda) = \arg\min_x S_\lambda(x)$ induces a front density $P(y) = q(\lambda)\ |\det \nabla_\lambda \Phi(\lambda)|^{-1}$, so a uniform covering requires sampling weights from

$$q^*(\lambda) \ \propto\ |\det \nabla_\lambda \Phi(\lambda)| \ =\ \sqrt{\det\big(J(\lambda)^\top J(\lambda)\big)}, \tag{1}$$

the Jacobian determinant of the solution map. The difficulty is that the solution-map Jacobian $J = -H_\lambda^{-1}(x^*)\partial_\lambda \nabla_x S_\lambda(x^*)$ requires computing Hessian and gradients at the unknown optimum x^*, and assumes invertibility of H_λ. Moreover, the choice of scalarisation affects what is reachable, e.g., a linear scalarization $\sum_i \lambda_i f_i$ reaches only the convex hull of the front, missing every concave or disconnected piece at any weight (Miettinen, 1999).

[1] By "evenly" we mean a controlled spread in objective space where uniformity is one target among others.

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.

Due to such issues, practitioners often resort to *population-based evolutionary* methods (NSGA-II/III Deb et al., 2002; Deb & Jain, 2014, MOEA/D Zhang & Li, 2007 and its weight-adapting descendants Li & Yao, 2020), which handle disconnected fronts without gradients but scale poorly in the decision dimension d. Our objective in our proposed approach is to retain the advantages of gradient-based optimisation while enjoying the enhanced coverage characteristic of evolutionary methods. To this end, we couple two sets of dynamics. First, the weights are dynamically evolved via mirror descent on a *slow* time-scale, while particles minimise an associated soft achievement scalarisation function (ASF) (Wierzbicki, 1980; Beck & Teboulle, 2012) at a faster time-scale. The weight enters the ASF as an adaptive reference $\bar{z}_i = z_i - \tau s_i \log \lambda_i$, thus coupling the two systems. This guarantees that every λ defines a Pareto point. To ensure even spread, an objective-space repulsion pushes nearby particles' weights apart on the simplex (in contrast to decision-space Stein repulsion (Liu & Wang, 2016; Liu et al., 2021b). This recovers the structure of ρ^* without forming $J(\lambda)$ of (1), by inverting the soft ASF (Section 4.2). Second, an evolutionary outer loop searches the genome (λ, τ) with convergence and spacing tests, encouraging the particles to move across disconnected patches of the pareto front.

Across DTLZ (Deb et al., 2005), WFG (Huband et al., 2006), ZCAT (Zapotecas-Martínez et al., 2023) and ZDT3 (Zitzler et al., 2000), the scheme matches or improves strong LibMOON (Zhang et al., 2024) baselines at a matched Jacobian budget (Section 5).

2 Related work

Scalarization reduces a multi-objective problem to a family of single-objective sub-problems (Miettinen, 1999). The weighted sum is differentiable and cheap but only recovers points on the convex hull of the front, so it misses concave and disconnected regions. The classical weighted Tchebycheff and ℓ_∞ scalarizations, also standard in Miettinen (1999), are Pareto-complete for arbitrary fronts, yet they are non-smooth, so not easily amenable to gradient-based optimisation. The modified Tchebycheff variant (Ma et al., 2018) improves the spread of solutions but keeps the same non-smoothness. Wierzbicki (1980) generalized these into the *ASF*, whose minima are weak Pareto-optimal for *any* reference point, attainable or not. A differentiable surrogate is obtained by replacing the inner $\max$ with its log-sum-exp soft-max, which Boyd & Vandenberghe (2004) already present as a classical differentiable approximation of the pointwise maximum. The same construction sits inside the nonsmooth-optimization smoothing framework of Nesterov (2005); Beck & Teboulle (2012). Applied with the preference *outside* the exponent, this yields the soft achievement form we use, whose front normal is the bare softmax π and whose λ is an adaptive shift of the reference point. Lin et al. (2024) apply a similar smoothing to the *weighted* Tchebycheff (preference inside the exponent instead and the normal $\pi \odot \lambda$) and prove Pareto and complexity properties for gradient-based MOO.

The MGDA (Désidéri, 2012) finds a common descent direction and provably converges to a Pareto-stationary point, and Sener & Koltun (2018) cast multi-task learning as MOO with a scalable MGDA upper bound. A large family of gradient-manipulation methods follows this line, including PCGrad (Yu et al., 2020), CAGrad (Liu et al., 2021a), Nash-MTL (Navon et al., 2022) and SDMGrad (Xiao et al., 2023). All of these target a single balanced solution per run, whereas we aim to cover the whole front. A second family attaches a preference to each run so that a set of preferences yields a set of solutions. EPO (Mahapatra & Rajan, 2020) descends with controlled ascent to hit an exact preference ray, Pareto MTL (Lin et al., 2019) partitions objective space into cones and solves a constrained sub-problem per cone, PMGDA (Zhang et al., 2026) enforces a preference as an equality constraint on the MGDA direction, and hypervolume-gradient ascent (Deist et al., 2023) moves a population along the gradient of the hypervolume indicator. These are the closest gradient baselines to our setting, and we compare against them through the LibMOON library (Zhang et al., 2024). The distinction is that their preferences are supplied and held fixed, while ours are optimised.

A common mechanism for spreading solutions across the Pareto front is repulsion. Various forms have been proposed. MOO-SVGD (Liu et al., 2021b), repels particles in decision space and adds the force to the MGDA update, so it needs pairwise RBF kernels in R^d and couples repulsion to the MGDA descent. Repulsion dynamics for uniform front approximation (Borghi, 2023) and its consensus-based extension (Borghi et al., 2023) instead make the scalarization parameters interact through a repulsive energy which is minimised using gradient-free methods. We adopt a similar approach, computing the repulsion force in objective space, with the same RBF kernel on $F_k \in$

$\mathbb{R}^m$, and apply it to the weights. Translating an objective-space force into a weight step would ordinarily require the solution-map Jacobian $\partial F/\partial\lambda$ evaluated at the unknown optimum. Crucially, in our reference-point scalarisation, the weights enter only as an additive shift of the reference, so a displacement of F corresponds to a displacement of $\log\lambda$ and the step is available in closed form. Related approaches, include MOEA/D (Zhang & Li, 2007), NSGA-II/III (Deb et al., 2002; Deb & Jain, 2014) and AdaW (Li & Yao, 2020) which evolve x under fixed or adapted weights. We include the associated fixed-weight aggregations from LibMOON as gradient baselines to our experiments. Quality-diversity methods (Mouret & Clune, 2015; Pierrot et al., 2022) again vary x. Our outer loop varies the genome (λ, τ) and promoting particles whose gradient is close to stationary and that are well spaced.

3 Background

3.1 Multi-objective optimization and Pareto optimality

We minimize m objectives jointly over a compact decision set,

$$\min_{x\in\mathcal{D}} f(x) = \big(f_1(x), \dots, f_m(x)\big), \qquad \mathcal{D}\subseteq\mathbb{R}^d, \quad f:\mathcal{D}\to\mathbb{R}^m, \tag{2}$$

and write $F = f(x) \in \mathbb{R}^m$ for the image of a point in objective space. Because there is no total order on $\mathbb{R}^m$, optimality is defined through *dominance*. For $x, y \in \mathcal{D}$ we say x *dominates* y if $f_i(x) \le f_i(y)$ for all i with at least one strict inequality, and x *strictly dominates* y if $f_i(x) < f_i(y)$ for all i (Miettinen, 1999; Ehrgott, 2005).

Definition 1 (Pareto optimality)**.** A point $x^* \in \mathcal{D}$ is *Pareto optimal* (efficient) if no $x \in \mathcal{D}$ dominates it, and *weakly Pareto optimal* if no $x \in \mathcal{D}$ strictly dominates it. Every Pareto optimal point is weakly Pareto optimal, but not conversely. The *Pareto set* $\mathcal{PS} = \{x^* \in \mathcal{D} : x^* \text{ is Pareto optimal}\}$ maps under f to the *Pareto front* $\mathcal{PF} = f(\mathcal{PS}) \subset \mathbb{R}^m$.

For differentiable objectives w.l.o.g assume a linear scalarisation then the first-order necessary condition for weak Pareto optimality is *Pareto stationarity*,

$$\exists\,\lambda\in\Delta_{m-1}:\ \sum_{i=1}^{m}\lambda_i\nabla f_i(x^*) = 0, \qquad \Delta_{m-1} = \Big\{\lambda\in\mathbb{R}^m_{\ge 0} : \textstyle\sum_i\lambda_i = 1\Big\}, \tag{3}$$

equivalently $\min_{\lambda\in\Delta_{m-1}}\|\sum_i\lambda_i\nabla f_i(x^*)\| = 0$ (Fliege & Svaiter, 2000; Désidéri, 2012). This is the condition our convergence tests (Section 4.3). It is sufficient for Pareto optimality when every f_i is convex on a convex $\mathcal{D}$ (Boyd & Vandenberghe, 2004) and only necessary otherwise, i.e. for the concave, disconnected and degenerate fronts.

3.2 Scalarisation and Soft Achievement Scalarising Function

Scalarization reduces (2) to a family of single-objective problems $\{\min_{x\in\mathcal{D}} S_\lambda(x) : \lambda\in\Delta_{m-1}\}$ (Miettinen, 1999). The *weighted sum* is smooth but reaches only the convex hull of the front (Das & Dennis, 1997; Ehrgott, 2005). The *weighted Tchebycheff* form against a reference z is Pareto-complete (i.e. every Pareto-optimal point of (2) is a global minimiser of S_θ for some θ), but non-smooth in x (Miettinen, 1999). We use the *ASF* of Wierzbicki (1980) with log-sum-exp smoothing (Beck & Teboulle, 2012): for reference $z\in\mathbb{R}^m$, scale $s\in\mathbb{R}^m_{>0}$, sharpness $\tau>0$, and $\lambda\in\Delta_{m-1}$,

$$S_{\lambda,\tau}(x) = \tau\log\Big(\textstyle\sum_{i=1}^m \lambda_i\, e^{(f_i(x)-z_i)/(\tau s_i)}\Big) \tag{4}$$

smooths $\max_i(f_i-\bar z_i)/s_i$, with $\bar z_i = z_i - \tau s_i\log\lambda_i$, so that steering λ moves an *adaptive reference* $\bar z(\lambda)$. Its gradient is a softmax-weighted combination of objective gradients,

$$\nabla_x S_{\lambda,\tau}(x) = \sum_{i=1}^m \frac{\pi_i(x)}{s_i}\,\nabla f_i(x), \qquad \pi_i = \frac{\lambda_i e^{(f_i-z_i)/(\tau s_i)}}{\sum_j \lambda_j e^{(f_j-z_j)/(\tau s_j)}}, \tag{5}$$

with $\pi(x)\in\Delta_{m-1}$. As $\tau\downarrow 0$, (4) epi-converges to the hard ASF, and as $\tau\to\infty$ it tends to a linear scalarization. Since every $\lambda_i > 0$, $S_{\lambda,\tau}$ is strictly increasing in each f_i, so every global minimiser

of (4) is Pareto optimal for any λ in the interior of Δ^{m-1} and any $\tau > 0$; local minimisers are Pareto-stationary with multiplier π. The hard ASF, by contrast, guarantees only weak Pareto optimality (Wierzbicki, 1980). The same monotonicity argument applies to the smooth weighted Tchebycheff of Lin et al. (2024); the two differ in the role of λ. front, (3).

The scalarisation (4) falls within a larger family. Indeed, setting $g_i = (f_i - z_i)/s_i$, we can define

$$S(x) = \sup_{p \in \Delta_{m-1}} \Big(\langle p,\, A_\lambda\, g \rangle - \tau\, \Omega(p) \Big) \;+\; \rho\, \langle w,\, g \rangle, \tag{6}$$

for $\rho \geq 0$, $\tau > 0$ and $A_\lambda \in \mathbb{R}^{m \times m}$, and $\Omega : \Delta_{m-1} \to \mathbb{R}$. Different selections of A_λ, and Ω yield familiar choices. For example, taking $A_\lambda = I$ and $\Omega = D_{\mathrm{KL}}(\cdot \| \lambda)$ with $\rho = 0$ recovers our form (4), where the preference λ enters as the KL prior. Placing the weights inside the map, $A_\lambda = \mathrm{diag}(\lambda)$ with Ω the negative entropy, gives the smooth (weighted) Tchebycheff scalarization of Lin et al. (2024), which letting $\tau \downarrow 0$ epi-converges to the hard Tchebycheff. Taking $\rho > 0$ adds the linear term that yields the *augmented* ASF of Wierzbicki (1980). The gradient of S can be expressed as $\nabla_x S = \sum_i (w_i/s_i)\, \nabla f_i$ where the weights $w \in \Delta_{m-1}$ are known as the *front normal*: at a stationary point $\nabla_x S = 0$ it is the multiplier of Pareto stationarity (3), the supporting direction of the front.

$$\nabla_x S_{\lambda,\tau,\rho} = \sum_{i=1}^{m} \frac{\pi_i + \rho \lambda_i}{s_i}\, \nabla f_i. \tag{7}$$

4 METHOD

We maintain N particles $(x_k, \lambda_k) \in \mathbb{R}^d \times \Delta_{m-1}$ and write $F_k = f(x_k)$, $f_{k,i} = f_i(x_k)$, with $\odot$ the element-wise product. **TWARD-S** is the inner two-time-scale solver (Algorithm 1). **TWARD** wraps it in an outer evolutionary search over genomes (λ, τ) (Algorithm 2).

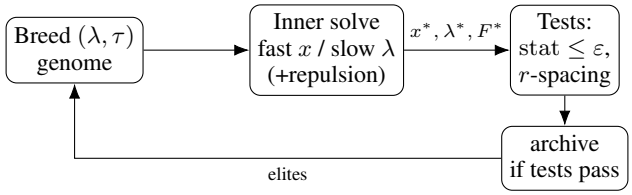


Figure 1: The bilevel loop.

4.1 FAST TIME-SCALE DYNAMICS

For a frozen λ the inner loop is defined by the gradient flow $\dot{x}(t) = -\nabla_x S_{\lambda,\tau}\big(x(t)\big)$, so $S_{\lambda,\tau}$, stationary point $x^*(\lambda_k)$. In practice, this is discretised using a forward Euler scheme with step-size η_x, yielding a sequence $x_k^{t+1} = x_k^t - \eta_x \nabla_x S_{\lambda_k,\tau}(x_k^t)$ The ASF $S_{\lambda,\tau}$ decreases along trajectories, and its equilibria are Pareto-stationary points (3) with multiplier proportional to $\pi(x^*)/s$. Each weight therefore selects a front point through the solution map $\lambda \to x^*(\lambda)$ with image $F^*(\lambda) = f(x^*(\lambda))$.

With $g_i = (f_i - z_i)/s_i$, differentiating (5) once more gives

$$\nabla_x^2 S_{\lambda,\tau} = \mathbb{E}_\pi\big[\nabla^2 g_I\big] + \tfrac{1}{\tau}\, \mathrm{Cov}_\pi\big(\nabla g_I\big), \qquad I \sim \pi, \tag{8}$$

The first term is the average curvature of the objectives. The second arises from the soft-max, which vanishes when one objective dominates and is largest when π is split between objectives whose gradients disagree. This term determines the Lipschitz constant, and hence the admissible step η_x, and on τ as shown in Lemma 1.

Assumption 1 (Smooth, bounded-gradient objectives)**.** Each $f_i \in C^2$ on a compact convex set $\mathcal{D}$ with $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i \|x - y\|$ and $\|\nabla f_i(x)\| \leq G_i$. Set $L_f = \max_i L_i$, $G_f = \max_i G_i$.

Lemma 1 (τ-dependent smoothness)**.** *Under Assumption 1, $\nabla_x S_{\lambda,\tau}$ is Lipschitz on $\mathcal{D}$ with constant at most $L'(\tau) = L_f + G_f^2/\tau$.*

The proof (Appendix A) splits the Hessian into a convex combination of objective Hessians (bounded by L_f) and $\frac{1}{\tau}$ times a softmax so the covariance of gradients (bounded by G_f^2), matching the smoothing form of Beck & Teboulle (2012); Lin et al. (2024). With $\eta_x = 1/L'(\tau)$ the descent lemma gives $S(x^{t+1}) \le S(x^t) - \frac{\eta_x}{2} \|\nabla_x S(x^t)\|^2$, so the admissible step shrinks like τ/G_f^2 as $\tau \downarrow 0$.

Theorem 1 (ε-stationarity, uniformly in λ)**.** *Under Assumption 1, let $g_i = (f_i - z_i)/s_i$ and $B = \max_{i,\, x\in\mathcal{D}} g_i(x) - \min_{i,\, x\in\mathcal{D}} g_i(x)$. For every $\lambda \in \Delta^{m-1}$, gradient descent on $S_{\lambda,\tau}$ with $\eta_x = 1/L'(\tau)$ reaches $\|\nabla_x S_{\lambda,\tau}(x^t)\| < \varepsilon$ for some $t < T$, where*

$$T \le \left\lceil \frac{2B\,(L_f + G_f^2/\tau)}{\varepsilon^2} \right\rceil. \tag{9}$$

For $S_{\lambda,\tau,\rho}$, the same bound holds with L_f replaced by $(1+\rho)L_f$ and B by $(1+\rho)B$ (see Lemma 2).

The proof is in Appendix A.3. The key observation is that the bound depends on τ but not on λ. Moreover, since $\pi \in \Delta^{m-1}$, $\|\nabla_x S_{\lambda,\tau}(x)\| < \varepsilon$ certifies the Pareto-stationarity residual (3) of the scaled objectives with multiplier $\pi(x)$, so the tolerance ε has the same meaning for every particle. While this exposition and the theory assumes gradient descent, in practice we use Adam (Kingma & Ba, 2015) for robustness to the stiffness of small-τ blocks.

4.2 Slow time-scale dynamics

The weights λ enter the ASF $S_{\lambda,\tau}$ as $\log\lambda$, so we require positive weights. Moreover, the objective $S_{\lambda,\tau}$ is invariant under multiplicative scaling of λ (up to a additive constant). Based on this, we impose that the weights λ lies on the unit simplex Δ_{m-1}, and are evolved under entropic mirror descent (Beck, 2017) with its KL-proximal step (see Appendix A.5 for construction). By separating the rates, so that $\eta_\lambda \ll \eta_x$ (Borkar, 2008) the slow update treats x as equilibrated, while the fast updates treat λ as frozen.

This yields a weight update defined through the entropic mirror step

$$\lambda_k^+ = \frac{\lambda_k \odot \exp(-\eta_\lambda\, \tilde{g}_k)}{\langle \lambda_k,\ \exp(-\eta_\lambda\, \tilde{g}_k)\rangle}, \qquad \text{equivalently} \qquad \log\lambda_k^+ = \log\lambda_k - \eta_\lambda\, \tilde{g}_k - c_k\mathbf{1}, \tag{10}$$

where the scalar c_k is the renormalisation onto Δ^{m-1}. It removes only the gauge direction $\mathbf{1}$, which leaves $S_{\lambda,\tau}$ unchanged. The exponent is

$$\tilde{g}_k = \underbrace{\gamma\, \frac{\hat{r}_k/(\tau s)}{\bar{r}}}_{\text{repulsion}} + \underbrace{\kappa\big(\log\lambda_k - \overline{\log\lambda_k}\,\mathbf{1}\big)}_{\text{centering}}, \qquad \overline{\log\lambda_k} = \tfrac{1}{m}\textstyle\sum_i \log\lambda_{k,i}, \quad \bar{r} = \tfrac{1}{N}\textstyle\sum_k \|\hat{r}_k/(\tau s)\|\,, \tag{11}$$

where $\gamma, \kappa \ge 0$ are tuning parameters. The update direction $\tilde{g}_k$ must balance two competing pressures: *repulsion*, which encourages even coverage of the front, and *centering*, a quadratic restoring force which pulls the reference point $\bar{z}$ back towards the ray emerging from z.

Repulsion. The slow scale must adapt the weights so that the points $F^*(\lambda_k)$ spread over the front. A naive direct approach would use the Jacobian $\partial F^*/\partial\lambda$, the same object that determines the ideal density (1), but this would require computing H_λ^{-1}, the inverse Hessian of S at an optimum that is not yet known, which is ill-conditioned on the fronts we target. We avoid both by inverting the scalarisation rather than the solution map.

More specifically, with $\tilde{F}_k = (F_k - \mu_F)/(\sigma_F + \epsilon)$ and RBF weights $\omega_{kj} = \exp(-\left\|\tilde{F}_k - \tilde{F}_j\right\|^2/h^2)$,

$$\hat{r}_k = \frac{1}{N-1}\sum_{j\neq k} \frac{2}{h^2}\,\omega_{kj}\,(\tilde{F}_k - \tilde{F}_j) \ \in \mathbb{R}^m, \tag{12}$$

defines the (normalized) negative gradient of the local repulsion energy $\sum_{j\neq k}\omega_{kj}$ w.r.t. $\tilde{F}_k$, which discourages crowding in objective space. We need to compute λ_k-update that moves the particle's objective vector F_k by the repulsion force $\hat{r}_k$. To this end, we use the fact that at a stationary point,

the softmax weights π_k of (5) are the normal to the front, i.e. they are determined by where F_k sits on the front, not directly by λ_k. The softmax relates the three quantities explicitly,

$$\pi_{k,i} = \frac{\lambda_{k,i}\, e^{u_{k,i}}}{Z_k}, \qquad u_{k,i} = \frac{f_{k,i} - z_i}{\tau s_i}, \qquad Z_k = \sum_l \lambda_{k,l} e^{u_{k,l}}, \tag{13}$$

and taking logarithms and solving for $f_{k,i}$ gives $f_{k,i} = z_i + \tau s_i\big(\log \pi_{k,i} - \log \lambda_{k,i} + \log Z_k\big)$.

Holding the normal π_k fixed and differentiating gives $\delta f_{k,i} = -\tau s_i\,(\delta \log \lambda_{k,i} - \delta \log Z_k)$. Since $\delta \log Z_k$ is common to all i, inverting yields, up to an additive constant that is removed by renormalisation onto Δ^{m-1},

$$\delta \log \lambda_{k,i} = -\frac{\delta f_{k,i}}{\tau s_i}, \tag{14}$$

Conversely, applying the log-weight step $-v/\tau$ with $v = \hat{r}_k/s$ does not in general move F_k by $\hat{r}_k$: the particle must stay on the front. On a locally flat patch with π_k fixed, (14) and $\pi_k^\top \delta g_k = 0$ give $\delta g_k = (I - \mathbf{1}\pi_k^\top)v$ with $g = (F - z)/s$, so the component of v along the front normal is discarded and the particle moves by the projection of v onto the tangent space, taken along the ray direction $\mathbf{1}$. On curved fronts, where π_k changes, this holds only approximately.

Why this scalarisation? The lift (14) inverts the softmax weights π of (5) at fixed π, because π is the front normal at a stationary point, i.e. for the $A_\lambda = I$, $\rho = 0$ member of (6), the coefficients n in $\nabla_x S = \sum_i n_i \nabla f_i / s_i$ are exactly π. Rescaling λ changes S only by a constant, so the inverse is diagonal in $\log \lambda$ up to a constant that renormalisation removes. The smooth weighted Tchebycheff of Lin et al. (2024), $\tau \log \sum_i e^{\lambda_i g_i/\tau}$ with softmax weights p, has normal $n \propto p \odot \lambda$. Holding this normal fixed gives $\delta \log \lambda_i = (\tau c - \lambda_i\, \delta g_i)/(\tau + \lambda_i g_i)$, where the scalar c is fixed by the simplex constraint. The feedback therefore depends on each particle's position through $\lambda_i g_i/\tau$, which converges to $-\delta g_i/g_i$ as $\tau \to 0$ it approaches, which is ill-conditioned near the reference point. More importantly, the coefficients $p \odot \lambda$ sum to $\sum_i p_i \lambda_i \le 1$: particles with small weights descend more slowly under a common η_x, and $\|\nabla_x S\| \le \varepsilon$ certifies only a Pareto-stationarity residual of $\varepsilon / \sum_i p_i \lambda_i$, whereas for (4) it certifies ε for every particle (Theorem 1), which makes the evolutionary approach feasible.

4.3 Outer Evolutionary Loop

The outer loop promotes exploration across disconnected fronts with fresh restarts. It searches over the scalarisation genome $(\lambda, \tau) \in \Delta^{m-1} \times \mathcal{T}$, an m-dimensional space independent of d, and keeps an archive $\mathcal{A}$ of converged, well-separated solutions, generated after G generations.

One generation.

1. *Breed.* Draw C child genomes from $\mathcal{A}$, uniformly at random in the first generation.
2. *Restart and solve.* Initialise each child at a fresh $x_0 \sim \mathrm{Unif}(D)$ with its genome's (λ, τ), and run Algorithm 1 for T_{in} steps on the C children jointly, so that repulsion acts among siblings. Each child returns (x^*, λ^*, F^*), and its final weights λ^* become its genome in the archive.
3. *Convergence test.* Discard the child unless $\mathrm{stat}(x^*) \le \varepsilon$, where stat is the gradient norm $\|\nabla_x S_{\lambda^*,\tau}(x^*)\|$ on the interior of D and the projected (KKT) residual at active box constraints (Appendix D.5). Every archived point is approximately Pareto-stationary.
4. *Spacing test.* Let $\mathcal{N} = \{a \in \mathcal{A} : d_s(F^*, F_a) < r\}$ with $d_s(F_a, F_b) = \|(F_a - F_b)/s\|$ and s fixed after the first generation. If $\mathcal{N} = \emptyset$, add the child. If the child is more stationary than every member of $\mathcal{N}$, it replaces all of $\mathcal{N}$. Otherwise discard it. In particular, r controls the maximum size of $\mathcal{A}$: a larger radius keeps fewer, more spaced particles (Appendix D.1).

5 Experiments

We evaluate **TWARD-S** (solver alone) and **TWARD** (outer search over (λ, τ)). Baselines are LibMOON preference methods (EPO, PMGDA, MGDA-UB, PMTL), MOO-SVGD, HVGrad and four

Algorithm 1 TWARD-S (inner)

Require: $f, N, T, \eta_x, \eta_\lambda, \tau, \gamma, \kappa, h$
1: init $x_k \in \mathcal{D}$, λ_k on Δ_{m-1}
2: **for** $t = 1, \dots, T$ **do**
3: $F_k \leftarrow f(x_k)$; $\pi_k \leftarrow$softmax (5)
4: $x_k \leftarrow x_k - \eta_x \sum_i (\pi_{k,i}/s_i) \nabla f_i(x_k)$ ▷ fast
5: $\hat{r}_k \leftarrow$(12); $\tilde{g}_k \leftarrow$(11)
6: $\lambda_k \leftarrow \lambda_k \odot e^{-\eta_\lambda \tilde{g}_k}$; renormalize ▷ slow
7: **end for**
8: **return** $\{(x_k, \lambda_k, F_k)\}$

Algorithm 2 TWARD (outer)

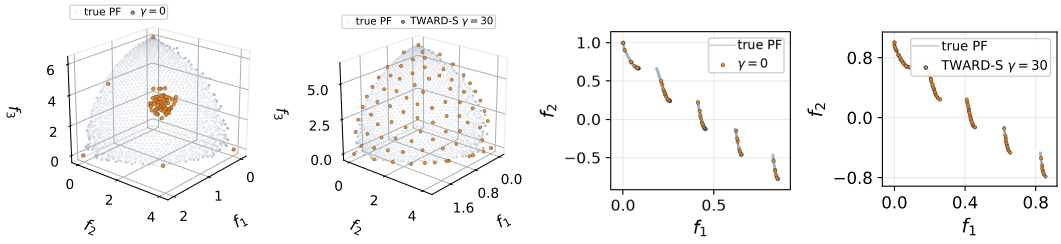
Require: $f, \mathcal{T}, C, r, \varepsilon, T_{\text{in}}, G$
1: $A \leftarrow \emptyset$
2: **for** $g = 1, \dots, G$ **do** (in parallel)
3: breed C genomes from A (random if empty)
4: **for** each genome **do**
5: restart x_0; run Alg. 1 $\rightarrow (x^*, \lambda^*, F^*)$
6: **if** $\text{stat}(x^*) > \varepsilon$ **then reject**
7: **else** update A by r-spacing on d_s (add / replace / reject)
8: **end for**
9: **end for**
10: **return** A

aggregations (Zhang et al., 2024). Benchmarks are ZDT (Zitzler et al., 2000), DTLZ (Deb et al., 2005), WFG (Huband et al., 2006) and ZCAT (Zapotecas-Martínez et al., 2023), scored by IGD ↓ and HV ↑ against the pymoo true fronts (Blank & Deb, 2020).

Fairness is counted in Jacobian evaluations of F. TWARD-S and every LibMOON baseline return N=100 solutions over T=1600 steps ($N \times T$ Jacobians). TWARD uses short inner trajectories of length $T = 400$ and matches that budget at G=4 with n_{init}=n_{children}=100. Most problems are m=3 in decision dimension $d \in \{12, 22, 30\}$ (ZDT3 is m=2, d=30). Full (d, m) and position/distance splits are in Table 4 with geometries ZDT3 (disconnected), DTLZ5/6 (degenerate), DTLZ7 (patches), WFG7 (concave), and ZCAT1/14/15 (irregular).

5.1 Repulsion is necessary

Without objective-space repulsion our scalarised approach collapses onto a small trade-off even when the same budget is spent. Thus we demonstrate by freezing every hyper-parameter except γ at identical cost (N=100, T=1600, TWARD-S) that repulsion is necessary to obtain an *evenly* spaced Pareto front. Figure 2 shows two examples of the attained fronts WFG7 and ZDT3, showing how repulsion $\gamma > 0$ promotes coverage. More examples can be found in Appendix D.4.



(a) WFG7, γ=0 **(b)** WFG7, γ>0 **(c)** ZDT3, γ=0 **(d)** ZDT3, γ>0

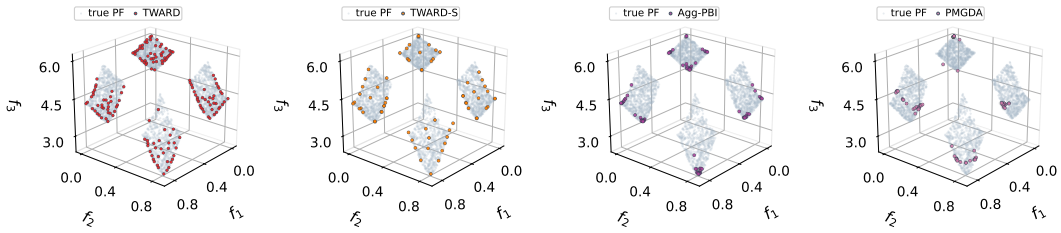
Figure 2: No Repulsion vs Repulsion , TWARD-S

5.2 Coverage against baselines

Figure 3 shows DTLZ7, a heavily non-convex disconnected benchmark with m=3 objectives and d=22 decision variables (Table 4). The four panels are the IGD top four among all twelve methods: (a) TWARD, (b) TWARD-S, (c) PBI, (d) PMGDA. Ranks are global among the twelve. TWARD and TWARD-S take ranks 1–2 and cover the patches. PBI and PMGDA are next and leave gaps. Full twelve-method fronts and tables are in Appendices D.6 and D.11.

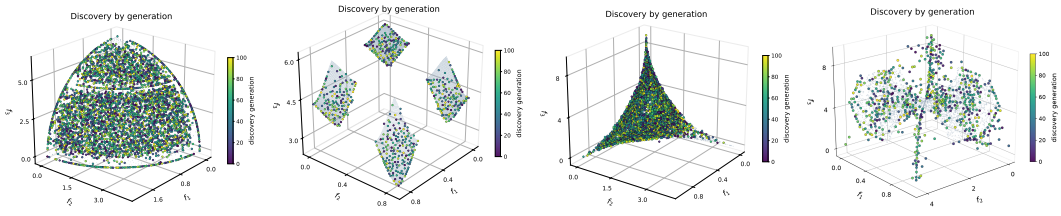
5.3 Running TWARD for more generations

Every method in the tables is tuned and run at one matched cost, which for TWARD means only G=4 generations. We see that more steps only refine the region discovered but do not open new



(a) #1 TWARD **(b)** #2 TWARD-S **(c)** #3 PBI **(d)** #4 PMGDA

Figure 3: DTLZ7: both of ours plus the two best baselines (IGD rank).



(a) WFG7 **(b)** DTLZ7 **(c)** ZCAT2 **(d)** ZCAT18

Figure 4: TWARD tuned at $G = 4$ run for G=100.

ones (see Appendix D.10). TWARD therefore spends extra budget on fresh genomes and restarts rather than longer trajectories.

We run the outer loop to G=100 at the same inner horizon T_{in}=400 (25× the matched cost). IGD falls monotonically on every problem, and most of the gain arrives by G=25. For G=0 (TWARD-S), 25 and 100, IGD is $0.235 \to 0.089 \to 0.053$ on WFG7, $0.096 \to 0.036 \to 0.024$ on DTLZ7, $0.142 \to 0.047 \to 0.035$ on ZCAT2 and $0.297 \to 0.096 \to 0.069$ on ZCAT18. two to four times below the $G = 4$ cost entries of Tables 1–2 (held-out seeds).

Figure 4 shows where the gain comes from. Early generations already touch every component of the front, and with TWARD-S all four DTLZ7 patches are hit (Appendix D.9). Later generations (yellow) sit between earlier ones across the whole front rather than in new regions. Extra generations fill the gaps between existing particles and replace less stationary ones through the spacing contest. The problems are chosen because each defeats a single gradient flow for a different reason. WFG7 has a concave front with a parameter-dependent bias, so uniform weights and restarts crowd into part of the front. DTLZ7 splits the front into four patches, and a descent path cannot cross from one patch to another. The ZCAT problems have d=30, objectives scaled 1:4:9 and a curved Pareto set, so descent must follow a nonlinear valley in x. ZCAT2 adds long thin arms toward the extreme points, and these are covered to their tips. On ZCAT18 the front switches abruptly between two 2-D wings and a 1-D curve, depending on which region x lies in.

5.4 Quantitative comparison

Tables 1–2 report IGD/HV (mean±std). Twelve-method fronts for every tabulated problem are in Appendix D.6. TWARD and TWARD-S dominate disconnected and degenerate DTLZ fronts (DTLZ5–7), while HVGrad still edges smooth ZDT3. DTLZ6 is harder twin of DTLZ5 (a steeper g), and every other gradient baseline stays off the front. Only TWARD-S/TWARD reach it. On WFG/ZCAT, TWARD is first on most rows by IGD. The runner-up is often HVGrad or TWARD-S, not always the solver alone. Robustness across seeds is discussed in Appendix D.2.

5.5 A real-data check: accuracy vs. fairness on UCI Adult

We train a small MLP (47→64→32→1, GELU, D=5185 weights) on UCI Adult and minimise three smooth objectives on a fixed training subsample: cross-entropy (accuracy proxy), demographic-parity gap on `sex`, and equalized-odds gap. Every returned network is re-scored on the held-out test split. With no true PF we report dominated hypervolume against a shared reference, the number

Problem	TWARD-S		TWARD		HVGrad		MOO-SVGD		soft-Tche		EPO	
	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV
ZDT3	0.010 ±0.000	1.094 ±0.000	0.010 ±0.001	1.093 ±0.000	**0.005** ±0.000	**1.096** ±0.000	0.088 ±0.013	1.031 ±0.033	0.033 ±0.003	1.084 ±0.002	0.046 ±0.005	1.060 ±0.010
DTLZ2	0.055 ±0.001	0.751 ±0.001	**0.029** ±0.001	**0.779** ±0.000	0.089 ±0.004	0.711 ±0.003	0.065 ±0.002	0.735 ±0.002	0.060 ±0.001	0.715 ±0.002	0.055 ±0.000	0.682 ±0.001
DTLZ5	**0.007** ±0.000	0.133 ±0.000	0.007 ±0.001	**0.133** ±0.000	0.233 ±0.174	0.091 ±0.031	0.016 ±0.000	0.128 ±0.000	0.009 ±0.001	0.132 ±0.000	0.010 ±0.001	0.124 ±0.001
DTLZ6	**0.008** ±0.001	**0.133** ±0.000	0.022 ±0.011	0.125 ±0.006	3.234 ±0.003	0.000 ±0.000	3.228 ±0.001	0.000 ±0.000	3.165 ±0.000	0.000 ±0.000	3.174 ±0.003	0.000 ±0.000
DTLZ7	0.092 ±0.004	1.402 ±0.004	**0.090** ±0.014	**1.416** ±0.009	0.296 ±0.034	1.010 ±0.079	0.281 ±0.028	0.995 ±0.017	0.235 ±0.019	1.248 ±0.024	0.360 ±0.035	0.871 ±0.027
WFG1	**1.514** ±0.233	**17.265** ±4.082	1.525 ±0.163	17.083 ±2.960	1.704 ±0.003	10.167 ±0.080	2.086 ±0.414	5.357 ±4.374	1.709 ±0.004	9.969 ±0.087	1.693 ±0.002	10.625 ±0.064
WFG2	0.467 ±0.082	54.982 ±3.365	**0.287** ±0.034	59.011 ±0.265	0.304 ±0.009	**59.494** ±0.057	0.329 ±0.018	58.539 ±0.208	0.373 ±0.044	58.174 ±0.390	0.486 ±0.072	55.860 ±0.472
WFG3	**0.022** ±0.001	**6.646** ±0.004	0.035 ±0.006	6.571 ±0.020	0.043 ±0.003	6.534 ±0.014	0.092 ±0.026	6.237 ±0.163	0.052 ±0.001	6.202 ±0.002	0.066 ±0.005	5.985 ±0.056
WFG4	0.640 ±0.092	21.245 ±0.536	0.482 ±0.026	24.529 ±0.417	0.492 ±0.021	23.648 ±0.703	**0.459** ±0.022	**25.418** ±0.733	0.580 ±0.026	22.305 ±0.819	0.597 ±0.034	21.460 ±0.768
WFG5	0.236 ±0.012	30.240 ±0.200	**0.215** ±0.012	**30.578** ±0.466	0.310 ±0.017	29.017 ±0.023	0.418 ±0.056	26.685 ±0.746	0.245 ±0.005	29.012 ±0.411	0.240 ±0.006	28.552 ±0.350
WFG6	0.220 ±0.004	35.744 ±0.063	**0.168** ±0.001	34.907 ±0.079	0.293 ±0.009	**36.094** ±0.057	0.202 ±0.003	35.764 ±0.020	0.265 ±0.002	32.949 ±0.277	0.247 ±0.004	32.207 ±0.131
WFG7	0.232 ±0.003	35.908 ±0.054	**0.160** ±0.006	**37.000** ±0.057	0.317 ±0.008	36.064 ±0.050	0.211 ±0.002	35.717 ±0.080	0.260 ±0.004	32.726 ±0.305	0.249 ±0.007	32.187 ±0.178
WFG8	0.438 ±0.012	27.863 ±0.450	**0.313** ±0.008	**29.959** ±0.156	0.425 ±0.024	29.066 ±0.526	0.439 ±0.039	26.273 ±0.784	0.462 ±0.013	24.472 ±0.487	0.426 ±0.017	25.321 ±0.303

Table 1: ZDT3/DTLZ and WFG: IGD↓/HV↑. Gold/silver/bronze = 1st/2nd/3rd.

Problem	TWARD-S		TWARD		HVGrad		MOO-SVGD		soft-Tche		EPO	
	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV
ZCAT1	0.243 ±0.013	36.003 ±0.212	**0.178** ±0.010	**36.991** ±0.099	0.239 ±0.012	36.767 ±0.122	0.581 ±0.052	31.968 ±0.673	0.280 ±0.010	35.600 ±0.176	0.687 ±0.071	24.539 ±0.875
ZCAT2	0.180 ±0.006	45.653 ±0.064	**0.086** ±0.003	**45.863** ±0.039	0.162 ±0.012	45.801 ±0.018	0.234 ±0.038	45.271 ±0.079	0.397 ±0.049	45.075 ±0.089	1.058 ±0.224	41.394 ±0.391
ZCAT3	0.204 ±0.004	23.657 ±0.025	**0.130** ±0.002	**24.183** ±0.029	0.200 ±0.003	23.722 ±0.079	0.284 ±0.025	22.196 ±0.266	0.349 ±0.024	18.846 ±0.535	0.448 ±0.002	18.527 ±0.154
ZCAT4	0.252 ±0.008	24.469 ±0.256	**0.149** ±0.004	**25.904** ±0.062	0.213 ±0.010	25.444 ±0.085	0.348 ±0.028	23.665 ±0.340	0.319 ±0.017	23.647 ±0.322	0.431 ±0.012	20.336 ±0.440
ZCAT5	0.169 ±0.003	42.891 ±0.045	**0.110** ±0.005	43.173 ±0.035	0.154 ±0.007	**43.208** ±0.045	0.196 ±0.017	43.063 ±0.081	0.257 ±0.020	42.146 ±0.230	0.495 ±0.050	37.774 ±0.538
ZCAT6	0.235 ±0.014	26.759 ±0.091	**0.160** ±0.011	**27.747** ±0.049	0.247 ±0.028	27.166 ±0.084	0.312 ±0.026	23.025 ±1.112	0.356 ±0.028	25.318 ±0.186	0.368 ±0.003	23.681 ±0.485
ZCAT7	0.231 ±0.008	25.044 ±0.260	**0.148** ±0.005	25.898 ±0.068	0.378 ±0.017	**25.911** ±0.039	0.322 ±0.010	24.655 ±0.279	0.386 ±0.029	22.413 ±0.101	0.372 ±0.017	19.727 ±0.304
ZCAT8	0.235 ±0.017	2.980 ±0.090	**0.143** ±0.013	**3.298** ±0.060	0.410 ±0.032	2.441 ±0.035	0.240 ±0.030	2.996 ±0.074	0.383 ±0.023	2.460 ±0.024	0.383 ±0.016	2.196 ±0.123
ZCAT9	0.284 ±0.015	13.144 ±0.150	**0.209** ±0.008	13.187 ±0.081	0.280 ±0.008	**13.260** ±0.070	0.429 ±0.062	11.335 ±0.459	0.416 ±0.023	12.353 ±0.191	0.501 ±0.041	10.364 ±0.596
ZCAT10	0.119 ±0.003	**6.041** ±0.042	**0.093** ±0.002	6.007 ±0.032	0.143 ±0.005	5.994 ±0.033	0.229 ±0.020	5.440 ±0.191	0.323 ±0.029	5.520 ±0.076	0.237 ±0.006	5.291 ±0.249
ZCAT11	0.264 ±0.014	23.436 ±0.150	**0.186** ±0.006	**24.055** ±0.106	0.236 ±0.016	24.051 ±0.313	0.329 ±0.035	22.767 ±0.573	0.431 ±0.023	21.898 ±0.384	0.532 ±0.030	19.277 ±0.988
ZCAT12	0.236 ±0.005	11.125 ±0.091	**0.140** ±0.006	**11.724** ±0.044	0.213 ±0.006	11.253 ±0.111	0.272 ±0.006	10.414 ±0.151	0.388 ±0.023	9.727 ±0.132	0.451 ±0.038	7.779 ±0.389
ZCAT13	0.353 ±0.053	29.336 ±0.797	**0.157** ±0.005	31.366 ±0.072	0.230 ±0.010	**31.427** ±0.042	0.333 ±0.035	29.625 ±0.479	0.511 ±0.081	26.466 ±0.653	0.834 ±0.036	25.544 ±0.734
ZCAT14	0.162 ±0.021	17.406 ±0.078	0.103 ±0.004	17.332 ±0.070	**0.102** ±0.015	**17.636** ±0.050	0.288 ±0.037	16.432 ±0.268	0.126 ±0.009	17.422 ±0.092	0.258 ±0.022	16.158 ±0.196
ZCAT15	0.066 ±0.007	**21.521** ±0.102	**0.062** ±0.006	21.449 ±0.047	0.111 ±0.020	21.348 ±0.286	0.237 ±0.048	20.382 ±0.247	0.143 ±0.022	21.116 ±0.110	0.263 ±0.037	19.449 ±0.397
ZCAT16	0.201 ±0.032	20.305 ±0.109	**0.130** ±0.015	**20.583** ±0.022	0.238 ±0.080	20.144 ±0.410	0.317 ±0.055	19.935 ±0.144	0.316 ±0.034	19.466 ±0.283	0.413 ±0.026	18.135 ±0.190
ZCAT17	0.242 ±0.023	42.506 ±0.081	0.229 ±0.142	**43.215** ±0.339	**0.203** ±0.019	42.806 ±0.201	0.213 ±0.021	42.410 ±0.420	0.259 ±0.019	42.805 ±0.125	0.382 ±0.036	40.867 ±0.458
ZCAT18	0.268 ±0.032	23.938 ±0.780	**0.126** ±0.005	26.223 ±0.045	0.172 ±0.009	**26.238** ±0.094	0.239 ±0.015	25.568 ±0.215	0.288 ±0.026	24.884 ±0.184	0.345 ±0.045	21.712 ±1.025
ZCAT19	0.197 ±0.015	23.054 ±0.132	**0.127** ±0.006	**24.004** ±0.097	0.180 ±0.027	23.667 ±0.208	0.353 ±0.039	21.824 ±0.195	0.204 ±0.018	23.090 ±0.165	0.316 ±0.035	19.618 ±0.630
ZCAT20	0.326 ±0.132	23.440 ±0.128	**0.161** ±0.008	24.400 ±0.181	0.191 ±0.010	**24.495** ±0.161	0.227 ±0.036	24.473 ±0.299	0.456 ±0.086	23.873 ±0.177	0.403 ±0.024	18.741 ±0.821

Table 2: ZCAT: IGD↓/HV↑. Gold/silver/bronze = 1st/2nd/3rd.

of non-dominated points and IGD to the pooled non-dominated set. Table 3 summarises the result. TWARD-S attains the highest test HV with the densest front (n=30), HVGrad is the closest baseline. TWARD packs a more uniform archive at slightly lower HV (c.f. Appendix D.12).

Table 3: UCI Adult, m=3 (CE, DP, EO). Test-scored fronts. Fronts in Appendix D.12.

Method	HV ↑	n	IGD_{pool} ↓
TWARD-S	1.634	30	0.013
TWARD	1.604	11	0.016
HVGrad	1.627	28	0.010
Agg-LS	1.627	17	0.013
MOO-SVGD	1.614	14	0.024

6 Conclusion

In this paper, we introduced TWARD, a Multi-Objective Optimisation method which performs gradient descent in the decision variables and global optimisation over scalarisation parameters. The approach leverages a novel entropic reference-point scalarisation, where the weights enter only as an additive shift to the reference point, yielding closed form weight updates without requiring solution-map Jacobians. The resulting two-time-scale solver spreads a population across the front by descent alone, and an evolutionary outer loop over (λ, τ), with stationarity and spacing tests, enables covering across disconnected fronts. At a matched budget of Jacobian evaluations, TWARD attains the best IGD on most ZDT, DTLZ, WFG and ZCAT problems against gradient-based baselines, with the largest margins on disconnected and degenerate fronts, and remains competitive on an accuracy–fairness task with a neural network. The method has clear limitations. The closed-form lift holds the front normal fixed, so it is only exact when fronts are locally flat. While TWARD is competitive, Hypervolume gradient ascent remains stronger on smooth low-dimensional fronts and challenging multi-modal fronts such as DTLZ1/3 defeat every gradient method we tested, including ours.

ETHICS STATEMENT

Most experiments use synthetic benchmark problems. The one real-data experiment uses the public UCI Adult dataset, which contains de-identified 1994 US census records, including demographic attributes. We use the sex attribute only to define the demographic-parity and equalized-odds objectives. No new data were collected, and no human subjects were involved. These fairness objectives are smooth group-level surrogates on a single dataset. A point on the resulting front does not certify that a model is fair in deployment, and choosing a trade-off between accuracy and fairness remains a decision for the practitioner. Multi-objective optimisation is a general-purpose tool; its downstream uses, including ones we would not endorse, are determined by the objectives a user supplies rather than by the method.

REPRODUCIBILITY STATEMENT

The method is specified by Algorithms 1 and 2, and the weight update and its derivation are given in the main text and appendices. The proofs of Lemma 1 and Theorem 1 are in Appendix A. The Appendices documens the objective implementations and their agreement with the reference implementations, the dominance tolerance used for scoring, the budget matching between method families, and the search grids. Hyper-parameters are listed for every problem and method. The experiments use Adam for the fast-scale step. Hyper-parameters are tuned on seed 0, and all reported results are means and standard deviations over held-out seeds 1–5. All problems, solvers and metrics are deterministic given a seed, and every number in the paper comes from a single scripted pipeline. Source code for the solver, the benchmark implementations and the plotting pipeline is provided as anonymised supplementary material.

AI USE STATEMENT

Generative AI tools were used as writing assistants (editing for clarity and compressing draft prose to fit the page limit) and to help format tables and LaTeX. They were not used to invent experimental results, theorems, or claims. All technical content, experiments, proofs and citations were produced and checked by the authors, who take full responsibility for the final manuscript.

REFERENCES

Amir Beck. *First-Order Methods in Optimization*. SIAM, 2017. doi: 10.1137/1.9781611974997.

Amir Beck and Marc Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012. doi: 10.1137/100818327.

Julian Blank and Kalyanmoy Deb. pymoo: Multi-objective optimization in Python. *IEEE Access*, 8: 89497–89509, 2020. doi: 10.1109/ACCESS.2020.2990567.

Giacomo Borghi. Repulsion dynamics for uniform Pareto front approximation in multi-objective optimization problems. *Proceedings in Applied Mathematics and Mechanics*, 23(1):e202200285, 2023. doi: 10.1002/pamm.202200285.

Giacomo Borghi, Michael Herty, and Lorenzo Pareschi. An adaptive consensus based method for multi-objective optimization with uniform Pareto front approximation. *Applied Mathematics & Optimization*, 88(2):58, 2023. doi: 10.1007/s00245-023-10036-y.

Vivek S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Texts and Readings in Mathematics. Hindustan Book Agency, 2008. doi: 10.1007/978-93-86279-38-5.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511804441. URL https://web.stanford.edu/~boyd/cvxbook/.

Indraneel Das and John E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63–69, 1997. doi: 10.1007/BF01197559.

Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014. doi: 10.1109/TEVC.2013.2281535.

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. doi: 10.1109/4235.996017.

Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In Ajith Abraham, Lakhmi Jain, and Robert Goldberg (eds.), *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, pp. 105–145. Springer, 2005. doi: 10.1007/1-84628-137-7_6.

Timo M. Deist, Monika Grewal, Frank J. W. M. Dankers, Tanja Alderliesten, and Peter A. N. Bosman. Multi-objective learning using HV maximization. In *Evolutionary Multi-Criterion Optimization*, volume 13970 of *Lecture Notes in Computer Science*, pp. 103–117. Springer, 2023. doi: 10.1007/978-3-031-27250-9_8.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathématique*, 350(5–6):313–318, 2012. doi: 10.1016/j.crma.2012.03.014.

Matthias Ehrgott. *Multicriteria Optimization*. Springer, 2 edition, 2005. doi: 10.1007/3-540-27659-9.

Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000. doi: 10.1007/s001860000043.

Simon Huband, Philip Hingston, Luigi Barone, and Lyndon While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006. doi: 10.1109/TEVC.2005.861417.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Miqing Li and Xin Yao. What weights work for you? adapting weights for any Pareto front shape in decomposition-based evolutionary multiobjective optimisation. *Evolutionary Computation*, 28 (2):227–253, 2020. doi: 10.1162/evco_a_00269.

Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. Pareto multi-task learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/685bfde03eb646c27ed565881917c71c-Abstract.html.

Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth Tchebycheff scalarization for multi-objective optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 30479–30509. PMLR, 2024. URL https://proceedings.mlr.press/v235/lin24y.html.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021a. URL https://proceedings.neurips.cc/paper/2021/hash/9d27fdf2477ffbff837d73ef7ae23db9-Abstract.html.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/b3ba8f1bee1238a2f37603d90b58898d-Abstract.html.

Xingchao Liu, Xin Tong, and Qiang Liu. Profiling Pareto front with multi-objective Stein variational gradient descent. In *Advances in Neural Information Processing Systems*, volume 34, 2021b. URL https://proceedings.neurips.cc/paper/2021/hash/7bb16972da003e87724f048d76b7e0e1-Abstract.html.

Xiaoliang Ma, Qingfu Zhang, Guangdong Tian, Junshan Yang, and Zexuan Zhu. On Tchebycheff decomposition approaches for multiobjective evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 22(2):226–244, 2018. doi: 10.1109/TEVC.2017.2704118.

Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in Pareto optimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6597–6607. PMLR, 2020. URL https://proceedings.mlr.press/v119/mahapatra20a.html.

Kaisa Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research & Management Science*. Springer, 1999. doi: 10.1007/978-1-4615-5563-6.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. arXiv:1504.04909, 2015. URL https://arxiv.org/abs/1504.04909.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16428–16446. PMLR, 2022. URL https://proceedings.mlr.press/v162/navon22a.html.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Springer, 2004. doi: 10.1007/978-1-4419-8853-9.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103 (1):127–152, 2005. doi: 10.1007/s10107-004-0552-5.

Thomas Pierrot, Guillaume Richard, Karim Beguir, and Antoine Cully. Multi-objective quality diversity optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 139–147. ACM, 2022. doi: 10.1145/3512290.3528823.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/432aca3a1e345e339f35a30c8f65edce-Abstract.html.

Dilin Wang, Ziyang Tang, Chandrajit Bajaj, and Qiang Liu. Stein variational gradient descent with matrix-valued kernels. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5dcd0ddd3d918c70d380d32bce4e733a-Abstract.html.

Andrzej P. Wierzbicki. The use of reference objectives in multiobjective optimization. In Günter Fandel and Tomas Gal (eds.), *Multiple Criteria Decision Making Theory and Application*, volume 177 of *Lecture Notes in Economics and Mathematical Systems*, pp. 468–486. Springer Berlin Heidelberg, 1980. doi: 10.1007/978-3-642-48782-8_32.

Peiyao Xiao, Hao Ban, and Kaiyi Ji. Direction-oriented multi-objective learning: Simple and provable stochastic algorithms. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/0e5b96f97c1813bb75f6c28532c2ecc7-Abstract-Conference.html.

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*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Saúl Zapotecas-Martínez, Carlos A. Coello Coello, Hernán E. Aguirre, and Kiyoshi Tanaka. Challenging test problems for multi- and many-objective optimization. *Swarm and Evolutionary Computation*, 81:101350, 2023. doi: 10.1016/j.swevo.2023.101350.

Qingfu Zhang and Hui Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007. doi: 10.1109/TEVC.2007.892759.

Xiaoyuan Zhang, Liang Zhao, Yingying Yu, Xi Lin, Yifan Chen, Han Zhao, and Qingfu Zhang. LibMOON: A gradient-based multiobjective optimization library in PyTorch. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/03ce71224741e6596059b8680a62978b-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Xiaoyuan Zhang, Xi Lin, and Qingfu Zhang. PMGDA: A preference-based multiple gradient descent algorithm. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 10(2): 1098–1110, 2026. doi: 10.1109/TETCI.2025.3526459.

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

A PROOFS

Throughout this appendix we absorb the per-objective scale into the objectives, i.e. we write f_i for f_i/s_i with $s \in \mathbb{R}^m_{>0}$ fixed. This is without loss of generality. A fixed positive scale replaces the constants of Assumption 1 by L_i/s_i and G_i/s_i and changes nothing else. The proofs are written for $\rho = 0$. Lemma 2 puts a frozen augmentation back. With $u_i = (f_i - z_i)/\tau$,

$$S_{\lambda,\tau,\rho}(x) = \tau \log \sum_{j=1}^{m} \lambda_j \, e^{u_j(x)} \; + \; \rho \sum_{j=1}^{m} \lambda_j \, g_j(x), \qquad g_j = f_j - z_j, \tag{15}$$

and $\pi_i = \lambda_i e^{u_i} / \sum_j \lambda_j e^{u_j}$. On the fast scale λ is held fixed. The $\rho = 0$ member is $S_{\lambda,\tau}$.

A.1 REDUCING $\rho > 0$ TO $\rho = 0$

Lemma 2 (Augmentation at frozen λ)**.** *Fix $\lambda \in \Delta_{m-1}$ and $\rho \geq 0$. As a function of x,*

$$S_{\lambda,\tau,\rho}(x) = S_{\lambda,\tau,0}(x) + \rho \sum_{j=1}^{m} \lambda_j \, g_j(x), \tag{16}$$

so

$$\nabla_x S_{\lambda,\tau,\rho} = \nabla_x S_{\lambda,\tau,0} + \rho \sum_{j=1}^{m} \lambda_j \, \nabla g_j, \tag{17}$$

$$\nabla^2_x S_{\lambda,\tau,\rho} = \nabla^2_x S_{\lambda,\tau,0} + \rho \sum_{j=1}^{m} \lambda_j \, \nabla^2 g_j. \tag{18}$$

Under Assumption 1, $\|\sum_j \lambda_j \nabla^2 g_j\| \leq L_f$. Hence if $\nabla_x S_{\lambda,\tau,0}$ is $L'(\tau)$-Lipschitz then $\nabla_x S_{\lambda,\tau,\rho}$ is $L'(\tau) + \rho L_f$ Lipschitz, i.e. replace L_f by $(1+\rho)L_f$ in $L_f + G_f^2/\tau$. Differentiating (15) in the weight gives

$$\frac{\partial S_{\lambda,\tau,\rho}}{\partial \lambda_i} = \frac{\tau \, \pi_i}{\lambda_i} + \rho \, g_i. \tag{19}$$

Proof. Equation (16) is (15) with the $\rho = 0$ term named. Differentiating in x once, then again, yields (17) and (18). For the norm, $\lambda \in \Delta_{m-1}$ and $\|\nabla^2 g_j\| \leq L_j$, so $\|\sum_j \lambda_j \nabla^2 g_j\| \leq \sum_j \lambda_j L_j \leq L_f$. Multiply by ρ and apply the triangle inequality. For (19), ∂_{λ_i} of the log-sum-exp is $\tau \cdot e^{u_i} / \sum_j \lambda_j e^{u_j} = \tau \pi_i / \lambda_i$. The linear term contributes ρg_i. □

A.2 Proof of Lemma 1 (τ-dependent smoothness)

Proof. Differentiating $S_{\lambda,\tau}$ once gives $\nabla_x S = \sum_i \pi_i \nabla f_i$, as in (5). From $\log \pi_i = \log \lambda_i + u_i - S/\tau$ and $\partial u_i/\partial x = \nabla f_i/\tau$,

$$\nabla_x \pi_i = \pi_i \, \nabla_x \log \pi_i = \frac{\pi_i}{\tau}\big(\nabla f_i - \nabla_x S\big). \tag{20}$$

Differentiating $\nabla_x S = \sum_i \pi_i \nabla f_i$ with the product rule and substituting (20) gives

$$\nabla_x^2 S_{\lambda,\tau} = \underbrace{\sum_{i=1}^m \pi_i \, \nabla^2 f_i}_{\text{(a)}} + \frac{1}{\tau} \underbrace{\Big(\sum_{i=1}^m \pi_i \nabla f_i \nabla f_i^\top - \nabla_x S \, (\nabla_x S)^\top \Big)}_{\text{(b)} = \mathrm{Cov}_\pi(\nabla f) \succeq 0}. \tag{21}$$

Form (b) is also $\sum_i \pi_i (\nabla f_i - \nabla_x S)(\nabla f_i)^\top$, using $\nabla_x S = \sum_j \pi_j \nabla f_j$.

For (a), $\pi \in \Delta_{m-1}$ and $\|\nabla^2 f_i\| \le L_i$ give $\big\|\sum_i \pi_i \nabla^2 f_i\big\| \le \sum_i \pi_i L_i \le L_f$. For (b), $0 \preceq \mathrm{Cov}_\pi(\nabla f) \preceq \sum_i \pi_i \nabla f_i \nabla f_i^\top$ in the PSD order, and the second-moment matrix has spectral norm at most $\sum_i \pi_i \|\nabla f_i\|^2 \le G_f^2$, since $\|\nabla f_i\| \le G_f$ implies $\|\nabla f_i\|^2 \le G_f^2$. By the triangle inequality, $\big\|\nabla_x^2 S_{\lambda,\tau}\big\| \le L_f + G_f^2/\tau = L'(\tau)$ on the compact convex set $\mathcal{D}$. A uniform Hessian-norm bound on a convex set is Lipschitz continuity of the gradient with that constant. □

A.3 Proof of Theorem 1 (ε-stationarity)

Proof. Write $S = S_{\lambda,\tau}$ and $L' = L'(\tau)$. By Lemma 1 and the descent lemma for L'-smooth functions (Nesterov, 2004), the step $x^{t+1} = x^t - \eta_x \nabla S(x^t)$ with $0 < \eta_x \le 1/L'$ satisfies

$$S(x^{t+1}) \le S(x^t) - \eta_x \Big(1 - \tfrac{\eta_x L'}{2}\Big) \big\|\nabla S(x^t)\big\|^2 \le S(x^t) - \frac{\eta_x}{2} \big\|\nabla S(x^t)\big\|^2. \tag{22}$$

The decrease coefficient is largest at $\eta_x = 1/L'$. Summing over $t = 0, \dots, T-1$ and telescoping, $\frac{\eta_x}{2} \sum_{t<T} \|\nabla S(x^t)\|^2 \le S(x^0) - S(x^T) \le V^0$ with $V^0 = S(x^0) - \min_x S$. Hence $\min_{t<T} \|\nabla S(x^t)\|^2 \le 2V^0/(\eta_x T)$, so $\min_{t<T} \|\nabla S(x^t)\| < \varepsilon$ as soon as $T > 2V^0/(\eta_x \varepsilon^2)$. Since $\sum_i \lambda_i = 1$, $\sum_i \lambda_i g_i \le S_{\lambda,\tau} \le \max_i g_i$ pointwise, so $V^0 \le \max_i g_i(x^0) - \min_{i,x} g_i(x) \le B$. Therefore, taking $\eta_x = 1/L'$ gives $T \le \lceil 2V^0 L'/\varepsilon^2 \rceil \le \lceil 2B(L_f + G_f^2/\tau)/\varepsilon^2 \rceil$, as required.

For $S_{\lambda,\tau,\rho}$ the same proof applies with L' replaced by $L' + \rho L_f$ (Lemma 2). □

Remark 1 (Projected steps). With box constraints the fast scale uses projected gradients. The same descent argument applies to the residual stat of (28), which vanishes iff x is a KKT point of $\min_{x \in [x_{\min}, x_{\max}]} S_{\lambda,\tau}$ (Def. 4).

A.4 The solution map

Let $x^*(\lambda)$ be a local minimizer of $x \mapsto S_{\lambda,\tau,\rho}(x)$ at which $H_\lambda = \nabla_x^2 S_{\lambda,\tau,\rho}(x^*)$ is invertible, and write S for $S_{\lambda,\tau,\rho}$.

Proposition 1 (Solution-map Jacobian)**.**

$$\frac{\partial x^*}{\partial \lambda_i} = -H_\lambda^{-1} \frac{\partial}{\partial \lambda_i} \nabla_x S \Big|_{x^*}. \tag{23}$$

Every sensitivity therefore passes through H_λ^{-1} at an unknown optimum, so the method never forms $J(\lambda)$.

Proof. Stationarity says $\nabla_x S(x^*(\lambda), \lambda) = 0$ for all λ near the point of interest. Differentiating in λ_i by the chain rule,

$$H_\lambda \, \frac{\partial x^*}{\partial \lambda_i} + \frac{\partial}{\partial \lambda_i} \nabla_x S \Big|_{x^*} = 0, \tag{24}$$

which rearranges to (23). Invertibility of H_λ is the step that fails on non-convex fronts. At $\rho = 0$ one has $\nabla_x S = \sum_j \pi_j \nabla g_j$ with $\pi = \mathrm{softmax}(u + \log \lambda)$, so $\partial \pi_j / \partial \lambda_i = (\pi_j/\lambda_i)(\delta_{ij} - \pi_i)$ and

$$\frac{\partial}{\partial \lambda_i} \nabla_x S = \frac{\pi_i}{\lambda_i} \Big(\nabla g_i - \sum_{j=1}^m \pi_j \nabla g_j \Big). \tag{25}$$

At a stationary point this collapses to $(\pi_i/\lambda_i)\nabla g_i$, still premultiplied by H_λ^{-1}. For $\rho > 0$, Lemma 2 adds $\rho\nabla g_i$ to the column. The H_λ^{-1} obstruction remains. □

A.5 Entropic mirror descent

The weights lie in Δ_{m-1}. Given $\tilde{g}_k \in \mathbb{R}^m$ and $\eta_\lambda > 0$, entropic mirror descent is the KL-proximal step

$$\lambda^+ = \arg\min_{\lambda \in \Delta_{m-1}} \Big\{ \langle \tilde{g}_k, \lambda \rangle + \frac{1}{\eta_\lambda} D_{\mathrm{KL}}(\lambda \| \lambda_k) \Big\}, \tag{26}$$

with $D_{\mathrm{KL}}(\lambda\|\lambda_k) = \sum_i \lambda_i \log(\lambda_i/\lambda_{k,i})$.

Proposition 2 (Closed form)**.** *The minimizer of* (26) *is*

$$\lambda_i^+ = \frac{\lambda_{k,i} \exp(-\eta_\lambda [\tilde{g}_k]_i)}{\sum_{j=1}^m \lambda_{k,j} \exp(-\eta_\lambda [\tilde{g}_k]_j)}. \tag{27}$$

If λ_k is in the relative interior, so is λ^+.

Proof. Up to a λ-independent constant the objective is $\sum_i \lambda_i [\tilde{g}_k]_i + (1/\eta_\lambda) \sum_i \lambda_i \log(\lambda_i/\lambda_{k,i})$ over $\lambda \in \Delta_{m-1}$. A multiplier ν for $\sum_i \lambda_i = 1$ gives the stationarity condition $[\tilde{g}_k]_i + (1/\eta_\lambda)(1 + \log \lambda_i - \log \lambda_{k,i}) + \nu = 0$, hence $\lambda_i \propto \lambda_{k,i} \exp(-\eta_\lambda [\tilde{g}_k]_i)$. Normalizing onto the simplex is (27). The exponential keeps every coordinate positive. □

B Experimental details

B.1 Objective transcriptions and their verification

Every benchmark is re-implemented as a differentiable JAX function so that the same objective is seen by our solver and, through a shared wrapper, by the gradient baselines. Each transcription is checked against the reference implementation (`pymoo` for ZDT/DTLZ and WFG, the published definition for ZCAT) on uniformly random points in the box the maximum absolute deviation is $\leq 5 \times 10^{-5}$ in float32 and $\sim 10^{-14}$ in float64, the float32 figure being consistent with round-off on objectives of magnitude up to 10^3. True fronts are taken from the reference implementation, so all methods are scored against identical $\mathcal{PF}$ and identical hypervolume reference points. Table 4 records the (d, m) and position/distance split used throughout. ZDT/DTLZ follow the standard pymoo defaults (Zitzler et al., 2000; Deb et al., 2005); every WFG instance uses $(k, \ell) = (4, 8)$ at $(d, m) = (12, 3)$ (Huband et al., 2006); every ZCAT instance uses $d = 10m = 30$ with $m = 3$, complicated PS and level 1 (Zapotecas-Martínez et al., 2023). The difficulty tags follow those sources (bias, multimodality, non-separability, degeneracy, disconnection).

Table 4: Problem dimensions used in all reported runs.

Problem	d	m	Position / distance	Difficulty (source)
ZDT3	30	2	pos. x_1; dist. 29	disconnected
DTLZ2	12	3	$k{=}d{-}m{+}1{=}10$	concave sphere
DTLZ5, DTLZ6	12	3	$k{=}10$	degenerate curve (DTLZ6: biased g)
DTLZ7	22	3	$k{=}20$	disconnected patches
DTLZ1, DTLZ3†	12	3	$k{=}10$	multimodal g (Appendix B.7)
WFG1–9	12	3	$(k, \ell){=}(4, 8)$	Huband suite (bias / disc. / deg. / multi. / deceptive / nonsep.)
ZCAT1–20	30	3	PS dim. $m{-}1$ (1 on ZCAT14–16)	complicated PS; varied PF shapes

†Excluded from the main comparison tables.

B.2 Dominance Testing

Every reported indicator is computed on the non-dominated subset of an attained set. Therefore, the dominance test itself will affect the outcome, and thus worth exploring in more detail. Objectives are

evaluated in float32, whose relative resolution is $\approx 1.2 \times 10^{-7}$, and on problems whose objectives reach 10^3 away from the front (DTLZ1, DTLZ3) that puts the rounding error at $\sim 10^{-4}$ in absolute terms. An exact comparison then lets a point that is worse by 10^3 in one objective survive on a 10^{-5} advantage in another, purely because the two values were rounded differently: we observed attained sets in which points at $\|F\| = 1.8 \times 10^3$ were retained next to points at $\|F\| = 53$ whose first two objectives agreed to five decimals. We therefore treat differences below 10^{-6} times the largest objective magnitude in the set as ties, one order above the float32 resolution.

The scale is taken over the whole set rather than per objective, which matters for exactly the problems that provoke the issue. On DTLZ2/3/5/6 and the WFG and ZCAT families every objective carries a shared factor such as $1 + g$, so the absolute rounding error of an objective whose own value is near zero is still governed by that factor: DTLZ3 attained sets contain $f_1 = -7.7 \times 10^{-5}$, negative where the objective is non-negative by construction. A per-objective tolerance would measure f_1 against its own magnitude, miss that error, and keep the outlier.

Fortunately, this choice is robust. Indeed, varying the tolerance from 10^{-7} to 10^{-4} leaves IGD and HV unchanged to four decimals on every problem while removing the inflation from GD (on DTLZ6, $0.658 \to 0.001$, and on DTLZ3, $80.8 \to 37.0$), and only at 10^{-3} do genuine trade-offs start to merge. IGD and HV are insensitive because a numerically tied outlier neither covers a new part of the front nor dominates any region a retained point does not already dominate. Then GD is sensitive precisely because it averages distances over the retained set.

B.3 Budget and scoring

Every method in the main comparison is run at the same budget in terms of Jacobian calculations. TWARD matches the budget through 4 generations and T=400 and initializing $N_i = 100$ for each generation i, similarly every other method uses N=100 returned solutions and T=1600 inner iterations. Each LibMOON baseline is given K=N preference rays stepped T times, so both families return the same number of solutions and take the same number of gradient steps. Our fast scale is a single fused JAX step over the whole population, whereas the LibMOON solvers loop once per ray, so wall-clock measures the implementation and the matched step count is what the comparison rests on. Every indicator is computed on the non-dominated subset of each attained set (Appendix B.2), scored against the true front for IGD and a fixed reference point per problem for HV.

B.4 Protocol for Experiments

The ablation (Section 5.1) fixes all hyper-parameters except the repulsion strength γ, using one population of N=100 particles with no evolutionary layer and the same temperature ladder, learning rates and seeds, so the difference between γ=0 and the tuned $\gamma^\star$ cannot be attributed to other confounding factors. The tables (Section 5) fix the returned count and the step budget across methods at N=100 and T=1600, so every row is a like-for-like comparison at fixed budget. The *fronts* (Section 5.2) use that same budget and show the attained set itself (Appendix D.7–D.6).

B.5 Role of τ

For any fixed $\tau > 0$, the construction below represents Pareto points admitting strictly positive stationarity multipliers for the scaled objectives as stationary points of the scalarisation.

Definition 2 (τ-reachable set). $R(\tau) = \overline{\{f(x^*) : x^* \text{ stationary for } S_{\lambda,\tau},\ \lambda \in \Delta_{m-1}\}} \cap \mathcal{PF}$.

For a fixed temperature $\tau > 0$, every Pareto point with an interior normal is a stationary point of $S_{\lambda,\tau}$ for some λ: given multipliers π, set $\lambda_i \propto \pi_i\, e^{-(f_i - z_i)/(\tau s_i)}$ and the softmax recovers π, so $\nabla_x S = 0$ (Section 3.2; Miettinen, 1999; Wierzbicki, 1980). Hence $R(\tau) = \mathcal{PF}$ up to closure for every τ. Thus a single τ is sufficient to ensure completeness.

However, there is no guarantee that these stationary points are local minima or reachable by gradient descent. This is where multiple τ values offers and advantage. Changing the scalarization can change the location, stability, and attraction basins of stationary points. A point that is inaccessible to descent under one configuration might become accessible under another Lemma 1 gives $L'(\tau) = L_f + G_f^2/\tau$. Large τ descends easily but softens toward linear scalarisation; small τ is sharper and

stiffer. With only N weights and T steps, which front points you actually hit depends on τ. Thus, the same λ can lead descent to different places, and some regions are easy at one τ and hard at another. Running $\{\tau_1, \ldots, \tau_K\}$ (one λ-block each) mixes those conditionings in one population.

B.6 Kernel metric alternatives

The repulsion (12) uses a Euclidean RBF kernel in standardized objective space. Two alternatives are natural. A Mahalanobis or local-kNN covariance adapts to anisotropic fronts and avoids mixing the geometry of different segments of a disconnected front (Wang et al., 2019). A Jacobian-pullback metric $d_J^2 = (F_k - F_j)^\top (J_f J_f^\top)^+ (F_k - F_j)$ measures distance along the tangent space of the front, which is the geometrically correct notion but costs a pseudo-inverse per particle. Only the Euclidean kernel was evaluated in this work.

B.7 Where Gradient methods fail: Multimodal distance terms

DTLZ1 and DTLZ3 deserve a separate note, because their scores look like a failure of the method when they are in fact a property of the landscape. Both multiply the front geometry by $1 + g$ with a distance term $g = 100\big(k + \sum_{j \geq m}[(x_j - 0.5)^2 - \cos(20\pi(x_j - 0.5))]\big)$, $k = d - m + 1$. Each tail coordinate therefore has 11 local basins spaced 0.1 apart in x, and only the central one, $x_j = 0.5$ for every $j \geq m$, lies on the front. A descent started from a uniform draw converges to the *nearest* basin, whose image sits at radius $1 + \sum_j j_i^2$ where j_i indexes the basin coordinate i started in: a median of about 85 for $k = 10$, against a front at radius 1. Reaching the front requires all k coordinates to start in the central basin, an event of probability 10^{-k}.

Three consequences follow, and they apply to every gradient method in our comparison, ours included. First, the score on these problems measures the best basin among the restarts a method can afford rather than convergence, and it improves only logarithmically in that number: on DTLZ3 we measure IGD 22.5 from 1008 restarts and 12.6 from 3888, against a combinatorial floor (the best $\sum_i j_i^2$ among N uniform draws) of 22 and 9 respectively. Spending the same budget as one larger population instead of more restarts changes nothing (512 particles in a single pass score the same 25.5 as 128), because what is being sampled is the basin, not the descent. Second, the baselines draw x_0 uniformly from the same box and consequently sit at a comparable distance, which is why every gradient method lands within a factor of two of the others there. Third, the step size must be small enough for a descent to settle *inside* a basin: with an Adam step larger than the 0.05 basin half-width the iterate steps across basins indefinitely and ends worse than the basin it started in (median g of 125 at $\eta_x = 0.02$ against 87 at $\eta_x = 0.002$, the latter matching the nearest-basin prediction exactly), and at $\eta_x = 0.05$ it simply saturates against the box bounds. The temperature ladder matters as much as the step size here: a single warm τ scores IGD 50.5 where the cold ladder scores 25.5 at an identical budget. Starting the same solver inside the global basin reaches the true front to numerical accuracy (median $\|F\| = 1.0000$, IGD 0.29), which confirms that the machinery is intact and the obstruction is the landscape.

C SUPPLEMENTARY MATERIAL

C.1 WHY A WEIGHTED SUM CANNOT COVER A CONCAVE FRONT

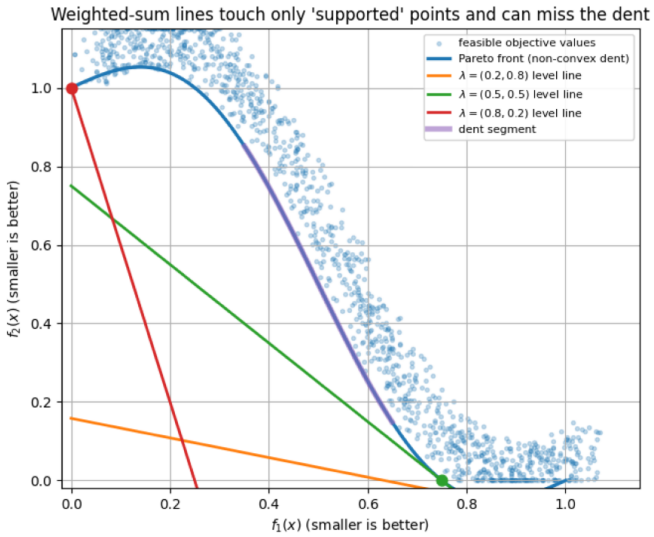


Figure 5: A weighted sum can only touch the front where a supporting hyperplane does, so concave and disconnected pieces are unreachable at any weight. This is the obstruction that (1) quantifies and that the soft-Tchebycheff scalarization of Section 3.2 removes.

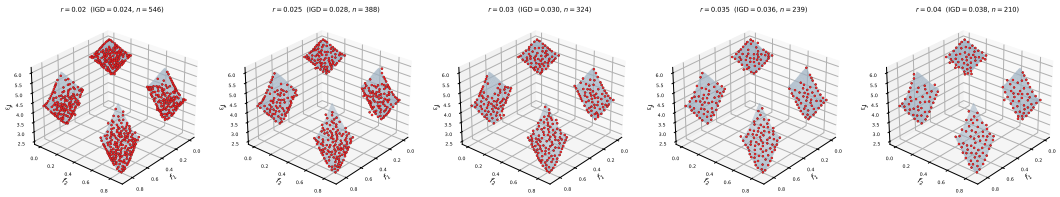
C.2 FURTHER RELATED WORK

Unlike Borghi (2023) and Borghi et al. (2023), which are gradient-free and diffuse the scalarization parameters, we keep gradient descent on x and a principled mirror descent on λ, and our outer search evolves the *pair* (λ, τ), while restarting x to cross basins. Unlike NSGA-II/III, MOEA/D and MAP-Elites, the genome is the scalarization itself rather than the decision vector, so the search space is m-dimensional regardless of d, and acceptance is decided by the gradient-based certificates of Def. 4 and Def. 5 rather than by dominance ranking. Compared with the preference-based gradient solvers (EPO, Pareto MTL, PMGDA), the preference is not supplied: it is a state variable with its own dynamics, and the temperature is searched alongside it. Wang et al. (2019) is the closest work on the metric question, and the kernel alternatives it suggests are discussed in Appendix B.

D ADDITIONAL RESULTS

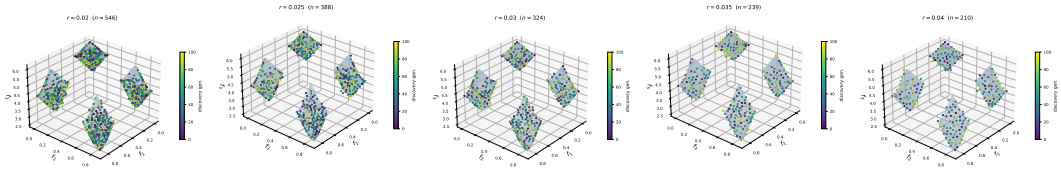
D.1 ARCHIVE GEOMETRY AND THE TWO TIME SCALES

Spacing test sensitivity. The archive is r-separated by the spacing test of Def. 5: a candidate whose objective vector falls within radius r of an existing elite is rejected (or contests the slot). That packing bound makes r a density parameter, with the outcome being sensitive to small changes. Around each problem's tuned default we therefore sweep five nearby values in steps of 0.005 (defaults r=0.01 on WFG7/ZCAT2/ZCAT19 and r=0.02 on DTLZ7/ZCAT18), all at G=100 with the TWARD winners and seed 0. For each problem, Figures 6–15 show the attained front and the discovery-coloured archive across the five radii; Figure 16 then summarizes IGD and archive size against r. Larger r thins the archive and can raise IGD once the front is under-sampled; too small an r packs redundant stacked points without improving coverage.



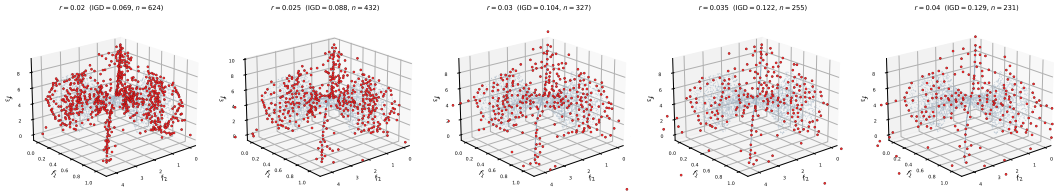
(a) $r = 0.02$ (IGD= 0.024) **(b)** $r = 0.025$ (IGD= 0.028) **(c)** $r = 0.03$ (IGD= 0.030) **(d)** $r = 0.035$ (IGD= 0.036) **(e)** $r = 0.04$ (IGD= 0.038)

Figure 6: DTLZ7: attained front across the spacing-radius sweep .



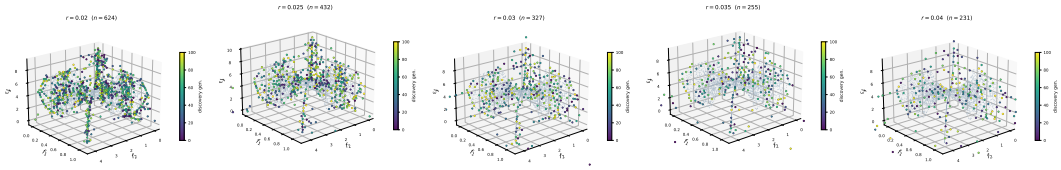
(a) $r = 0.02\,(n = 546)$ **(b)** $r = 0.025\,(n = 388)$ **(c)** $r = 0.03\,(n = 324)$ **(d)** $r = 0.035\,(n = 239)$ **(e)** $r = 0.04\,(n = 210)$

Figure 7: DTLZ7: discovery by generation across the same radii (colour = entry generation).



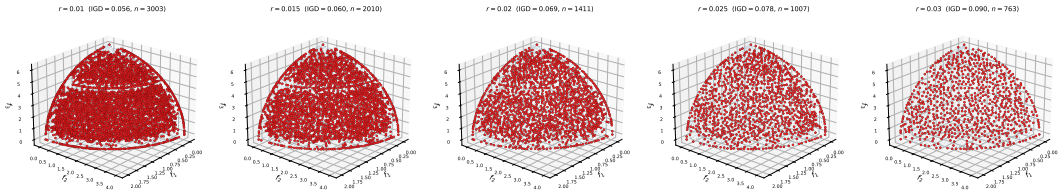
(a) $r = 0.02$ (IGD= 0.069) **(b)** $r = 0.025$ (IGD= 0.088) **(c)** $r = 0.03$ (IGD= 0.104) **(d)** $r = 0.035$ (IGD= 0.122) **(e)** $r = 0.04$ (IGD= 0.129)

Figure 8: ZCAT18: attained front across the spacing-radius sweep .



(a) $r = 0.02\,(n = 624)$ **(b)** $r = 0.025\,(n = 432)$ **(c)** $r = 0.03\,(n = 327)$ **(d)** $r = 0.035\,(n = 255)$ **(e)** $r = 0.04\,(n = 231)$

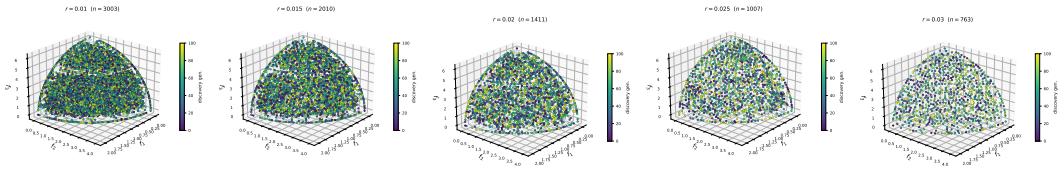
Figure 9: ZCAT18: discovery by generation across the same radii (colour = entry generation).



(a) $r = 0.01$ (IGD= 0.056) **(b)** $r = 0.015$ (IGD= 0.060) **(c)** $r = 0.02$ (IGD= 0.069) **(d)** $r = 0.025$ (IGD= 0.078) **(e)** $r = 0.03$ (IGD= 0.090)

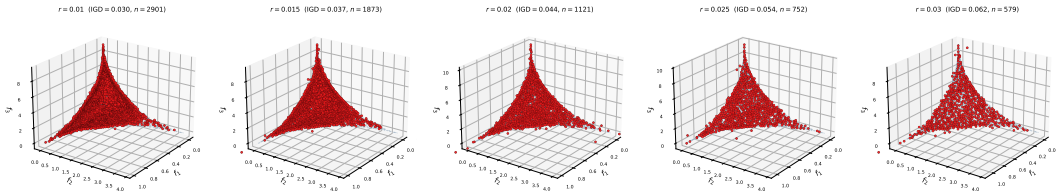
Figure 10: WFG7: attained front across the spacing-radius sweep .

The two scales are visibly separated. Figure 82 plots λ and x on a shared step axis for a TWARD-S run. The decision variables settle within a few hundred inner steps while the weights are still drifting thousands of steps later, which is the condition under which the fast-scale analysis of Section 4.1



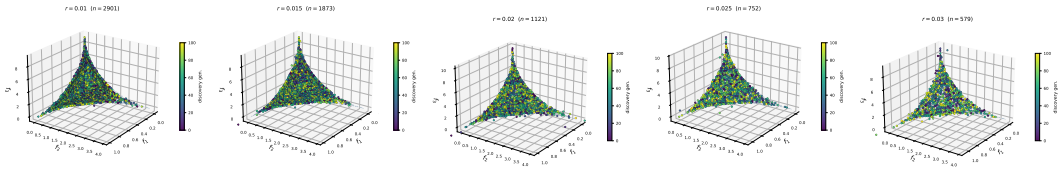
(a) $r = 0.01$ $(n = 3003)$ **(b)** $r = 0.015$ $(n = 2010)$ **(c)** $r = 0.02$ $(n = 1411)$ **(d)** $r = 0.025$ $(n = 1007)$ **(e)** $r = 0.03$ $(n = 763)$

Figure 11: WFG7: discovery by generation across the same radii (colour = entry generation).



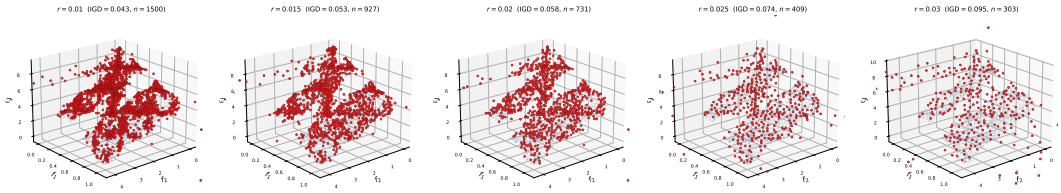
(a) $r = 0.01$ (IGD$=0.030$) **(b)** $r = 0.015$ (IGD$=0.037$) **(c)** $r = 0.02$ (IGD$=0.044$) **(d)** $r = 0.025$ (IGD$=0.054$) **(e)** $r = 0.03$ (IGD$=0.062$)

Figure 12: ZCAT2: attained front across the spacing-radius sweep .



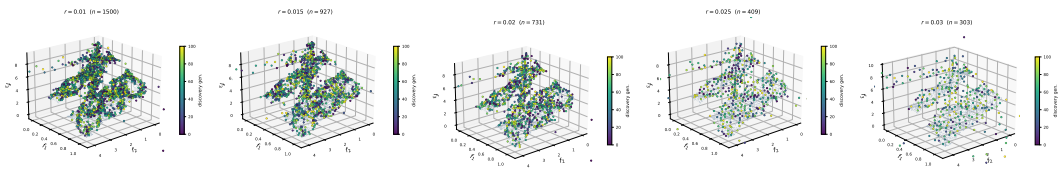
(a) $r = 0.01$ $(n = 2901)$ **(b)** $r = 0.015$ $(n = 1873)$ **(c)** $r = 0.02$ $(n = 1121)$ **(d)** $r = 0.025$ $(n = 752)$ **(e)** $r = 0.03$ $(n = 579)$

Figure 13: ZCAT2: discovery by generation across the same radii (colour = entry generation).



(a) $r = 0.01$ (IGD$=0.043$) **(b)** $r = 0.015$ (IGD$=0.053$) **(c)** $r = 0.02$ (IGD$=0.058$) **(d)** $r = 0.025$ (IGD$=0.074$) **(e)** $r = 0.03$ (IGD$=0.095$)

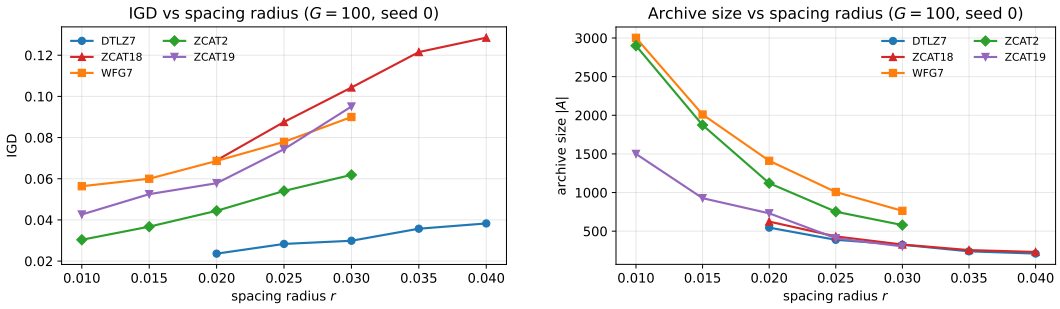
Figure 14: ZCAT19: attained front across the spacing-radius sweep .



(a) $r = 0.01$ $(n = 1500)$ **(b)** $r = 0.015$ $(n = 927)$ **(c)** $r = 0.02$ $(n = 731)$ **(d)** $r = 0.025$ $(n = 409)$ **(e)** $r = 0.03$ $(n = 303)$

Figure 15: ZCAT19: discovery by generation across the same radii (colour = entry generation).

applies at frozen λ. When the two move on the same timescale, η_λ is too large for the η_x in use, and the panel shows it directly.



(a) IGD vs r

(b) Archive size vs r

Figure 16: Spacing-test summary at G=100, seed 0 (TWARD with $\kappa = \rho = 0$).

D.2 TUNING: TRAIN SEED VERSUS HELD-OUT SEEDS

Hyper-parameters for both TWARD-S and TWARD are selected on a single *train* seed (0). The main-text tables then re-evaluate the frozen winners on held-out seeds 1–5 (mean±std). Table 6 reports IGD for both splits.

Training seeds. On seed 0, TWARD (evo) already beats TWARD-S on $28/31$ problems. The solver wins only on ZDT3, DTLZ6 and WFG3. So the train seed does *not* favour the solver alone. What it *does* show is that a single-seed ranking can overstate gaps: on DTLZ7, evo looks far ahead on seed 0 (0.060 vs. 0.090) but the two are nearly tied on seeds 1–5 (0.090 vs. 0.092).

Robustness on the test seeds. The train-seed winner matches the multiseed winner on $30/31$ problems (only DTLZ5 flips, within noise). Median relative IGD degradation from seed 0 to the 1–5 mean is about 13% for TWARD-S and 18% for TWARD, so both transfer, and the outer loop's advantage is not an artefact of the tuning seed.

Table 5: Selected TWARD-S / TWARD configuration per problem (train seed 0). Columns: temperature ladder, repulsion (γ, h), restoring force κ, augmentation ρ, and Adam rates.

Problem	Method	τ	γ	h	κ	ρ	η_x	η_λ
ZDT3	TWARD-S	[0.15, 0.2, 0.7]	7.5	0.08	0.5	0.2	8.0e-3	1.0e-4
ZDT3	TWARD	[0.08, 0.5]	10	0.1	0.5	0.01	8.0e-3	5.0e-4
DTLZ2	TWARD-S	[0.05]	12.5	0.3	0.5	0	1.0e-3	1.0e-3
DTLZ2	TWARD	[0.05, 0.08]	12.5	0.4	0	0	5.0e-3	2.0e-3
DTLZ4	TWARD	[0.05, 0.5]	15	0.12	0	0	5.0e-3	2.0e-3
DTLZ5	TWARD-S	[0.05]	50	0.2	0	0	2.0e-3	2.0e-4
DTLZ5	TWARD	[0.05, 0.1]	50	0.6	0.5	0	5.0e-3	2.0e-4
DTLZ6	TWARD-S	[0.05]	5	0.12	0	0	0.02	1.0e-3
DTLZ6	TWARD	[0.03, 0.15]	5	0.1	0	0	3.0e-3	2.0e-3
DTLZ7	TWARD-S	[0.15]	7.5	0.2	0.5	0.1	2.0e-3	2.0e-3
DTLZ7	TWARD	[0.08, 0.1, 0.7]	3	0.08	0	0.1	3.0e-3	2.0e-3
WFG1	TWARD-S	[0.3, 0.9]	3	0.25	2	0.2	8.0e-3	5.0e-4
WFG1	TWARD	[0.7]	15	0.2	0	0	0.02	5.0e-3
WFG2	TWARD-S	[0.15, 0.4, 1.2]	40	0.4	0	0.01	0.02	2.0e-3
WFG2	TWARD	[0.3, 1.2]	15	0.4	0.5	0.05	0.05	1.0e-3
WFG3	TWARD-S	[0.05, 0.5]	1	0.08	0	0.05	0.05	2.0e-3
WFG3	TWARD	[0.3]	20	0.12	1	0.01	0.05	5.0e-4
WFG4	TWARD-S	[0.7]	40	0.03	2	0.05	0.05	5.0e-3
WFG4	TWARD	[0.4, 0.7]	5	1	2	0.05	0.05	0.01
WFG5	TWARD-S	[0.05]	30	0.35	0.5	0	0.01	2.0e-4
WFG5	TWARD	[0.2, 0.3]	4	0.6	2	0.05	0.03	2.0e-3
WFG6	TWARD-S	[0.05]	30	0.3	1	0.01	0.03	1.0e-3
WFG6	TWARD	[0.05]	7.5	0.35	0	0	0.05	5.0e-3
WFG7	TWARD-S	[0.05]	20	0.3	0.5	0	0.02	2.0e-3
WFG7	TWARD	[0.05]	10	0.3	0	0	0.05	5.0e-3
WFG8	TWARD-S	[0.05]	5	0.35	0	0.01	0.03	2.0e-3
WFG8	TWARD	[0.1]	12.5	0.15	0	0	0.05	0.01
WFG9	TWARD-S	[0.08]	12.5	0.5	0	0.05	0.05	5.0e-4
WFG9	TWARD	[0.1, 0.2]	4	0.08	0	0	0.05	0.01
ZCAT1	TWARD-S	[0.03, 0.05, 0.2]	15	0.2	0	0.01	0.05	5.0e-3
ZCAT1	TWARD	[0.08, 0.5]	20	0.35	2	0	0.05	1.0e-3
ZCAT2	TWARD-S	[0.2, 0.9, 1.2]	5	0.15	1	0.2	0.03	1.0e-3
ZCAT2	TWARD	[0.5, 0.9]	5	0.1	0.5	0.1	0.05	2.0e-3
ZCAT3	TWARD-S	[0.05, 0.15]	3	0.2	0	0.1	0.05	5.0e-3
ZCAT3	TWARD	[0.4]	2	0.4	0.5	0.05	0.05	0.01
ZCAT4	TWARD-S	[0.05, 0.2]	1.5	0.25	0	0.05	0.05	0.01
ZCAT4	TWARD	[0.3, 0.4, 1.2]	1.5	0.3	0.5	0.01	0.05	0.01
ZCAT5	TWARD-S	[0.08]	10	0.3	0	0	0.05	5.0e-3
ZCAT5	TWARD	[0.7, 0.9, 1.2]	5	0.4	0.5	0.2	0.05	0.01
ZCAT6	TWARD-S	[0.03, 0.05, 0.08]	30	0.2	0	0.01	0.05	5.0e-3
ZCAT6	TWARD	[0.1]	40	0.35	0.5	0.01	0.05	1.0e-3
ZCAT7	TWARD-S	[0.03, 0.4]	40	0.4	1	0.2	0.02	5.0e-3
ZCAT7	TWARD	[0.08, 1.2]	10	0.35	0.5	0.01	0.05	5.0e-3
ZCAT8	TWARD-S	[0.4]	1	0.4	2	0.05	0.01	5.0e-3
ZCAT8	TWARD	[0.02, 0.05, 0.3]	3	0.25	0.5	0.05	0.05	0.01
ZCAT9	TWARD-S	[0.2, 0.5]	5	0.4	2	0.01	0.05	0.01
ZCAT9	TWARD	[0.03, 0.1, 0.4]	7.5	0.4	1	0.05	0.05	0.01
ZCAT10	TWARD-S	[0.1, 0.15, 0.2]	7.5	0.25	2	0.05	0.05	0.01
ZCAT10	TWARD	[0.15, 0.2, 0.3]	20	0.2	2	0.05	0.05	2.0e-4
ZCAT11	TWARD-S	[0.7, 0.9, 1.2]	1	0.35	0.5	0.1	0.03	0.01
ZCAT11	TWARD	[0.2, 0.7, 0.9]	4	0.5	1	0.1	0.05	0.01
ZCAT12	TWARD-S	[0.2, 1.2]	2	0.4	0.5	0.2	0.05	0.01
ZCAT12	TWARD	[0.4, 0.9, 1.2]	2	0.4	0.5	0	0.05	0.01
ZCAT13	TWARD-S	[0.05, 0.15, 0.2]	2	0.25	0	0.01	0.03	5.0e-3
ZCAT13	TWARD	[0.4, 0.9, 1.2]	10	0.35	1	0	0.05	5.0e-3
ZCAT14	TWARD-S	[0.5, 1.2]	2	0.9	1	0.1	0.02	0.01
ZCAT14	TWARD	[0.4, 1.2]	20	0.01	0.5	0.01	0.05	0.01
ZCAT15	TWARD-S	[0.9]	1.5	0.08	1	0.2	0.03	0.01
ZCAT15	TWARD	[0.7, 0.9, 1.2]	3	0.01	1	0.01	0.05	0.01
ZCAT16	TWARD-S	[0.01, 0.08, 0.7]	20	0.35	1	0.1	0.03	5.0e-3
ZCAT16	TWARD	[1.2]	4	0.8	0.5	0.01	0.05	2.0e-3
ZCAT17	TWARD-S	[0.2, 0.4, 0.5]	15	0.9	1	0.05	0.05	5.0e-4
ZCAT17	TWARD	[0.7]	1	0.6	0.5	0.05	0.05	0.01
ZCAT18	TWARD-S	[0.03, 0.05, 0.15]	50	0.35	1	0	0.03	1.0e-3
ZCAT18	TWARD	[0.9]	12.5	0.25	2	0.1	0.05	5.0e-3
ZCAT19	TWARD-S	[0.7, 0.9, 1.2]	0.5	0.25	1	0.2	0.01	0.01
ZCAT19	TWARD	[1.2]	7.5	0.02	1	0.1	0.05	0.01
ZCAT20	TWARD-S	[0.9]	4	0.07	0.5	0.01	0.03	0.01
ZCAT20	TWARD	[0.2, 1.2]	4	0.7	1	0.2	0.05	0.01

Problem	TWARD-S$_0$	TWARD$_0$	TWARD-S$_{1:5}$	TWARD$_{1:5}$	win$_0$	win$_{1:5}$
ZDT3	**0.008**	0.008	**0.010** $\pm$0.000	0.010 $\pm$0.001	S	S
DTLZ2	0.052	**0.027**	0.055 $\pm$0.001	**0.029** $\pm$0.001	E	E
DTLZ5	0.006	**0.005**	**0.007** $\pm$0.000	0.007 $\pm$0.001	E	S
DTLZ6	**0.007**	0.008	**0.008** $\pm$0.001	0.022 $\pm$0.011	S	S
DTLZ7	0.090	**0.060**	0.092 $\pm$0.004	**0.090** $\pm$0.014	E	E
WFG2	0.387	**0.254**	0.467 $\pm$0.082	**0.287** $\pm$0.034	E	E
WFG3	**0.019**	0.022	**0.022** $\pm$0.001	0.035 $\pm$0.006	S	S
WFG4	0.492	**0.407**	0.640 $\pm$0.092	**0.482** $\pm$0.026	E	E
WFG5	0.217	**0.182**	0.236 $\pm$0.012	**0.215** $\pm$0.012	E	E
WFG6	0.220	**0.155**	0.220 $\pm$0.004	**0.168** $\pm$0.001	E	E
WFG7	0.225	**0.134**	0.232 $\pm$0.003	**0.160** $\pm$0.006	E	E
ZCAT1	0.209	**0.133**	0.243 $\pm$0.013	**0.178** $\pm$0.010	E	E
ZCAT2	0.142	**0.077**	0.180 $\pm$0.006	**0.086** $\pm$0.003	E	E
ZCAT3	0.205	**0.119**	0.204 $\pm$0.004	**0.130** $\pm$0.002	E	E
ZCAT4	0.239	**0.140**	0.252 $\pm$0.008	**0.149** $\pm$0.004	E	E
ZCAT5	0.162	**0.103**	0.169 $\pm$0.003	**0.110** $\pm$0.005	E	E
ZCAT6	0.204	**0.137**	0.235 $\pm$0.014	**0.160** $\pm$0.011	E	E
ZCAT7	0.205	**0.133**	0.231 $\pm$0.008	**0.148** $\pm$0.005	E	E
ZCAT8	0.186	**0.115**	0.235 $\pm$0.017	**0.143** $\pm$0.013	E	E
ZCAT9	0.263	**0.178**	0.284 $\pm$0.015	**0.209** $\pm$0.008	E	E
ZCAT10	0.118	**0.081**	0.119 $\pm$0.003	**0.093** $\pm$0.002	E	E
ZCAT11	0.245	**0.158**	0.264 $\pm$0.014	**0.186** $\pm$0.006	E	E
ZCAT12	0.224	**0.137**	0.236 $\pm$0.005	**0.140** $\pm$0.006	E	E
ZCAT13	0.240	**0.142**	0.353 $\pm$0.053	**0.157** $\pm$0.005	E	E
ZCAT14	0.101	**0.078**	0.162 $\pm$0.021	**0.103** $\pm$0.004	E	E
ZCAT15	0.057	**0.049**	0.066 $\pm$0.007	**0.062** $\pm$0.006	E	E
ZCAT16	0.158	**0.090**	0.201 $\pm$0.032	**0.130** $\pm$0.015	E	E
ZCAT17	0.206	**0.109**	0.242 $\pm$0.023	**0.229** $\pm$0.142	E	E
ZCAT18	0.209	**0.116**	0.268 $\pm$0.032	**0.126** $\pm$0.005	E	E
ZCAT19	0.182	**0.113**	0.197 $\pm$0.015	**0.127** $\pm$0.006	E	E
ZCAT20	0.227	**0.143**	0.326 $\pm$0.132	**0.161** $\pm$0.008	E	E

Table 6: IGD↓ on the tuning seed (0) versus held-out seeds 1–5 (mean$\pm$std). Bold marks the better of TWARD-S vs. TWARD in each block. The train-seed winner matches the multiseed winner on $29/31$ problems.

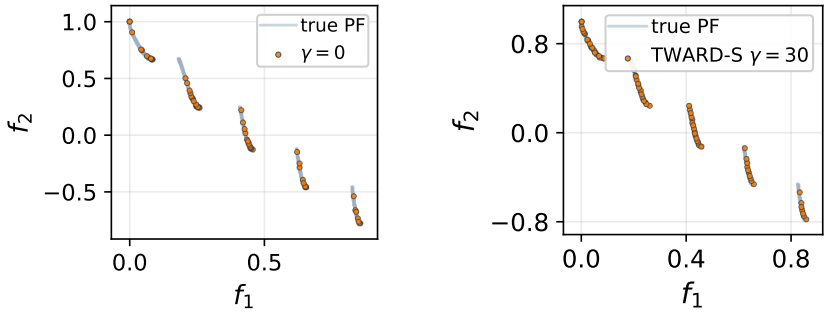
Where the method leads and where it does not. On ZDT/DTLZ geometries (disconnected patches, degenerate curves, concave surfaces), TWARD and TWARD-S are at or near the best IGD, with a large margin on DTLZ6. On WFG and ZCAT the same pattern holds in Tables 1 and 2, with HVGrad or MOO-SVGD occasionally second. Multimodal distance problems (DTLZ1/3) remain hard for every gradient method tested (Appendix B.7).

D.3 Hyper-parameter sensitivty

The spacing radius r is swept in Appendix D.1 (fronts and discovery rows, then Figure 16): near the tuned default, steps of 0.005 already move IGD and archive size. The repulsion strength γ is contrasted at off versus tuned in Figure 2 and Appendix D.4. The bandwidth h and the test tolerance ε are fixed per problem by the tuning grid of Appendix B.

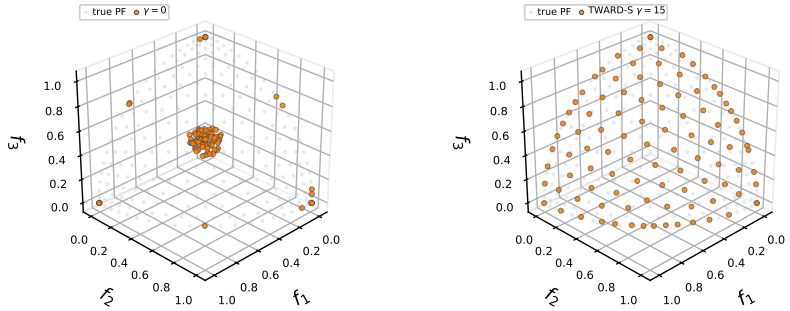
D.4 Repulsion ablation, all problems

Figure 2 isolates γ on WFG7 and ZDT3 in the main text. Below, every tabulated problem gets the same TWARD-S comparison at matched cost (N=100, T=1600): repulsion off (γ=0) versus the tuned $\gamma^\star$.



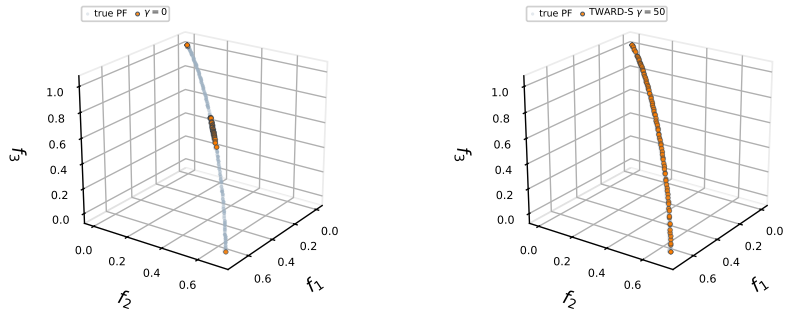
(a) γ=0 (0.020) **(b)** $\gamma^\star$ (0.008)

Figure 17: ZDT3: repulsion off vs on (TWARD-S, N=100, T=1600).



(a) γ=0 (0.214) **(b)** $\gamma^\star$ (0.053)

Figure 18: DTLZ2: repulsion off vs on (TWARD-S, N=100, T=1600).

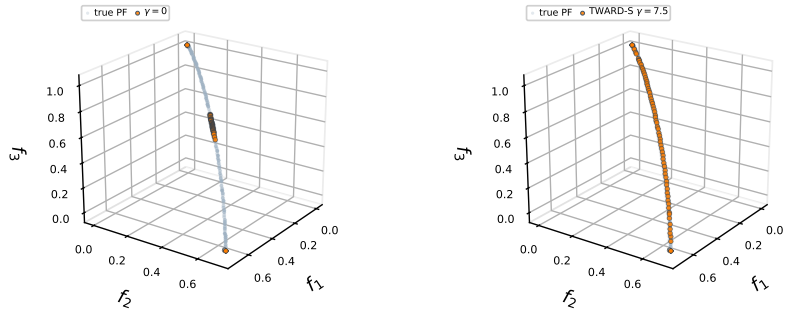


(a) γ=0 (0.149) **(b)** $\gamma^\star$ (0.005)

Figure 19: DTLZ5: repulsion off vs on (TWARD-S, N=100, T=1600).

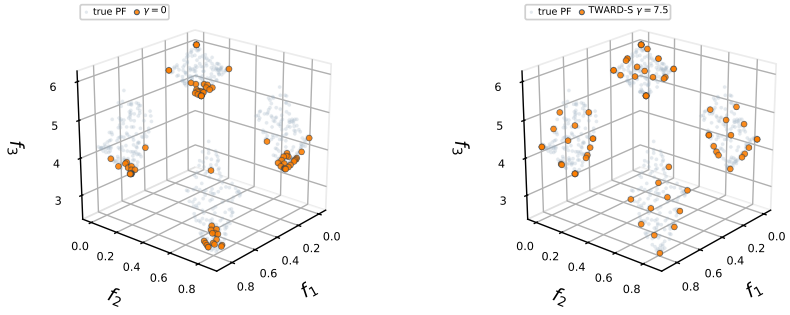
D.5 Details of the Outer Evolutionary Search

Definition 3 (Genome space and random inner map)**.** The genome space is $\mathcal{G} = \Delta_{m-1} \times \mathcal{T}$ with $\mathcal{T} = \{\tau_1 < \cdots < \tau_K\}$. The random inner map $\Phi : \mathcal{G} \times \Omega \to \mathbb{R}^d \times \Delta_{m-1} \times \mathbb{R}^m$ runs Algorithm 1 for T_{in} steps from a fresh restart $x_0(\omega) \sim \text{Unif}([x_{\min}, x_{\max}]^d)$ with weight initialized at λ and temperature fixed at τ, returning (x^*, λ^*, F^*) and the residual $\text{stat}(x^*)$ below.



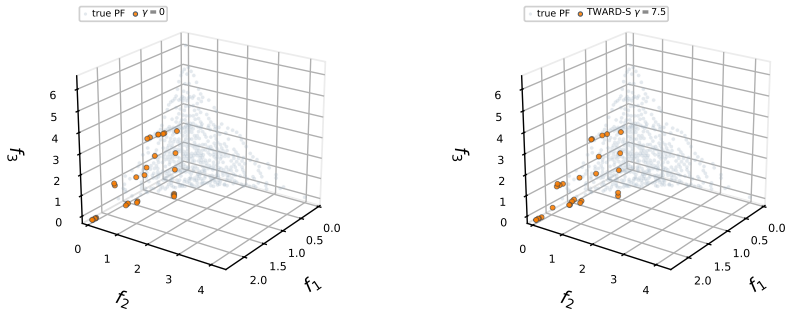
(a) γ=0 (0.154) **(b)** $\gamma^\star$ (0.007)

Figure 20: DTLZ6: repulsion off vs on (TWARD-S, N=100, T=1600).



(a) γ=0 (0.206) **(b)** $\gamma^\star$ (0.091)

Figure 21: DTLZ7: repulsion off vs on (TWARD-S, N=100, T=1600).



(a) γ=0 (1.370) **(b)** $\gamma^\star$ (1.360)

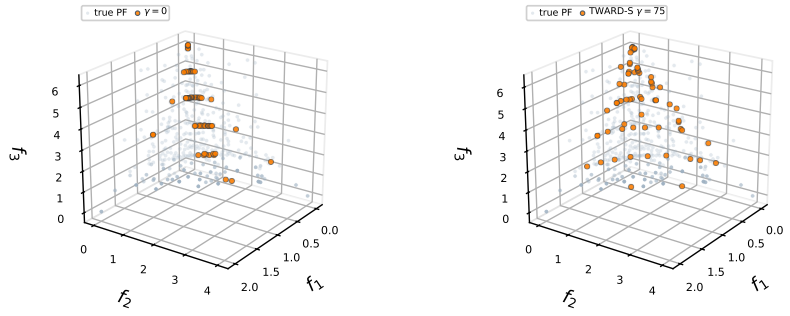
Figure 22: WFG1: repulsion off vs on (TWARD-S, N=100, T=1600).

Definition 4 (Convergence Test)**.** With $\nabla_x S_{\lambda,\tau}$ from (5), the box-projected (approximate-KKT) residual is

$$\mathrm{stat}(x) = \left\| x - \mathrm{clip}_{[x_{\min}, x_{\max}]}\big(x - \nabla_x S_{\lambda,\tau}(x)\big) \right\|, \tag{28}$$

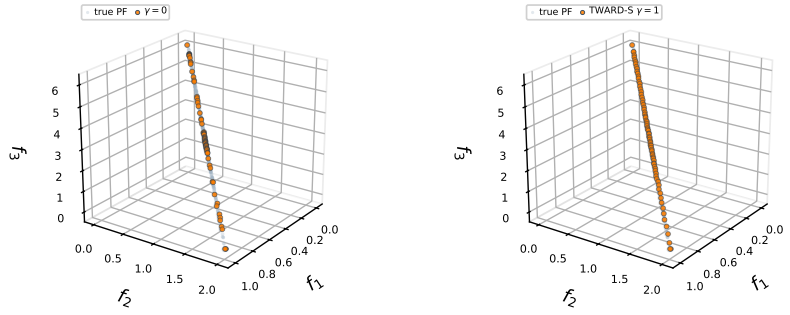
and the trial passes iff $\mathrm{stat}(x^*) \le \varepsilon$. On the interior this reduces to $\|\nabla_x S_{\lambda,\tau}(x^*)\| \le \varepsilon$ (Theorem 1). The projection makes it a correct KKT test at active box bounds.

Definition 5 (Spacing Test)**.** In objective space normalized by the population range s, two images are r-close iff $d_s(F_a, F_b) = \|(F_a - F_b)/s\| < r$. The radius r sets the minimum archive spacing.



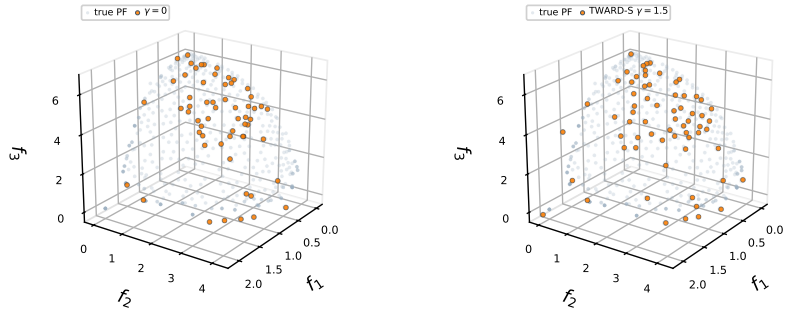
(a) γ=0 (0.604) **(b)** $\gamma^\star$ (0.403)

Figure 23: WFG2: repulsion off vs on (TWARD-S, N=100, T=1600).



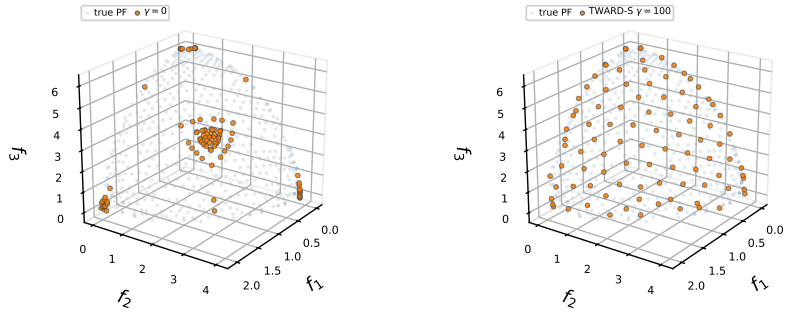
(a) γ=0 (0.052) **(b)** $\gamma^\star$ (0.021)

Figure 24: WFG3: repulsion off vs on (TWARD-S, N=100, T=1600).



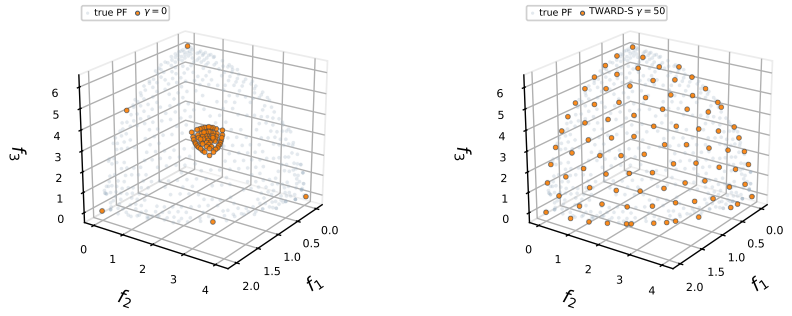
(a) γ=0 (0.660) **(b)** $\gamma^\star$ (0.519)

Figure 25: WFG4: repulsion off vs on (TWARD-S, N=100, T=1600).



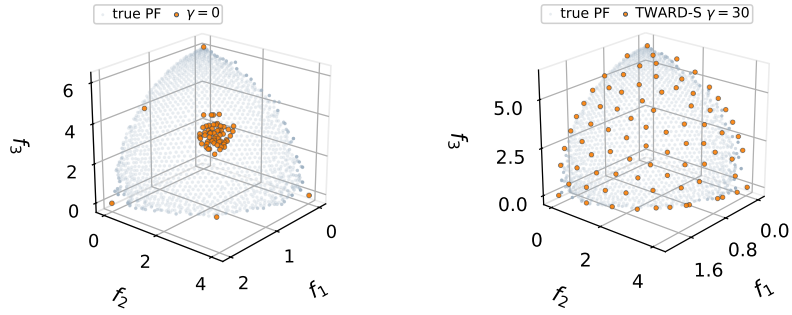
(a) γ=0 (0.598) **(b)** $\gamma^\star$ (0.219)

Figure 26: WFG5: repulsion off vs on (TWARD-S, N=100, T=1600).



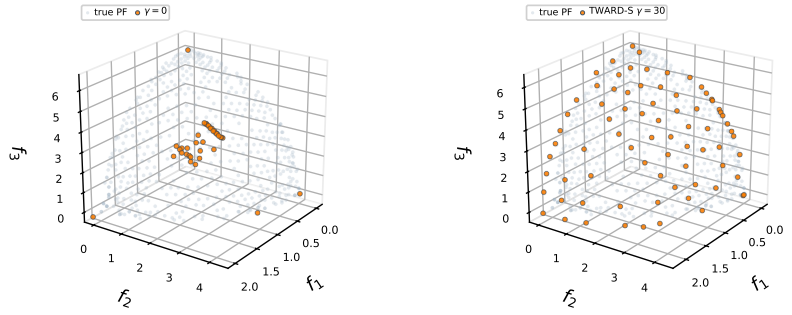
(a) γ=0 (0.981) **(b)** $\gamma^\star$ (0.220)

Figure 27: WFG6: repulsion off vs on (TWARD-S, N=100, T=1600).



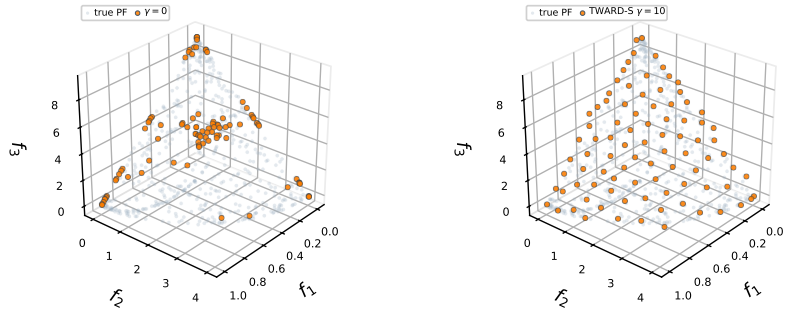
(a) γ=0 (0.924) **(b)** $\gamma^\star$ (0.236)

Figure 28: WFG7: repulsion off vs on (TWARD-S, N=100, T=1600).



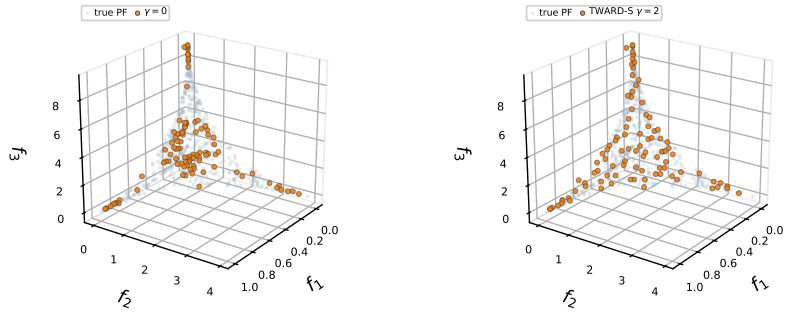
(a) γ=0 (1.017) **(b)** $\gamma^\star$ (0.385)

Figure 29: WFG8: repulsion off vs on (TWARD-S, N=100, T=1600).



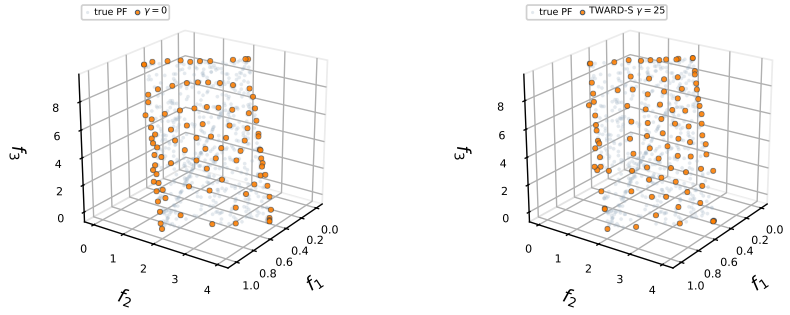
(a) γ=0 (0.592) **(b)** $\gamma^\star$ (0.210)

Figure 30: ZCAT1: repulsion off vs on (TWARD-S, N=100, T=1600).



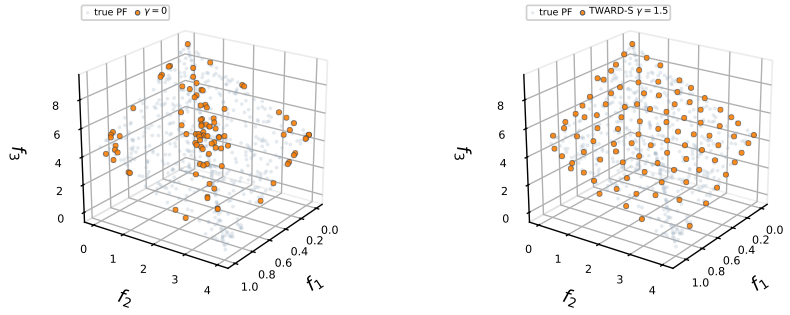
(a) γ=0 (0.265) **(b)** $\gamma^\star$ (0.144)

Figure 31: ZCAT2: repulsion off vs on (TWARD-S, N=100, T=1600).



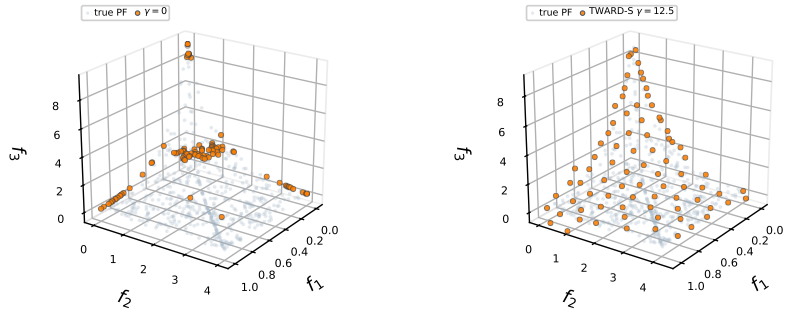
(a) γ=0 (0.261) **(b)** $\gamma^\star$ (0.236)

Figure 32: ZCAT3: repulsion off vs on (TWARD-S, N=100, T=1600).



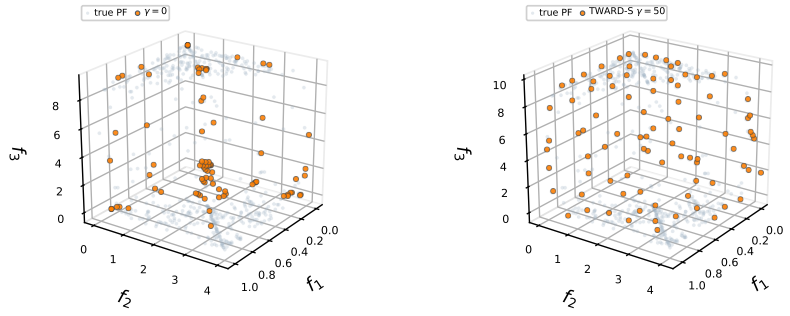
(a) γ=0 (0.341) **(b)** $\gamma^\star$ (0.209)

Figure 33: ZCAT4: repulsion off vs on (TWARD-S, N=100, T=1600).



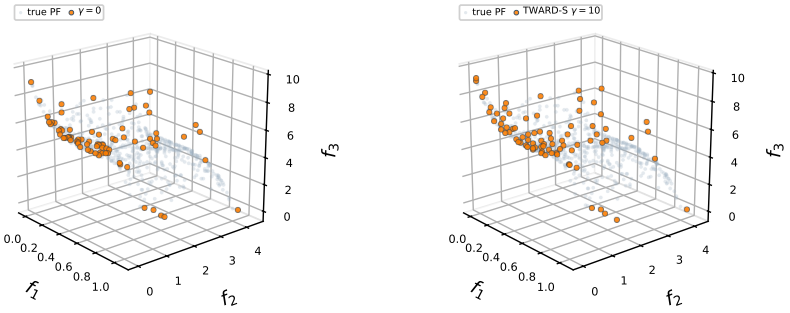
(a) γ=0 (0.470) **(b)** $\gamma^\star$ (0.166)

Figure 34: ZCAT5: repulsion off vs on (TWARD-S, N=100, T=1600).



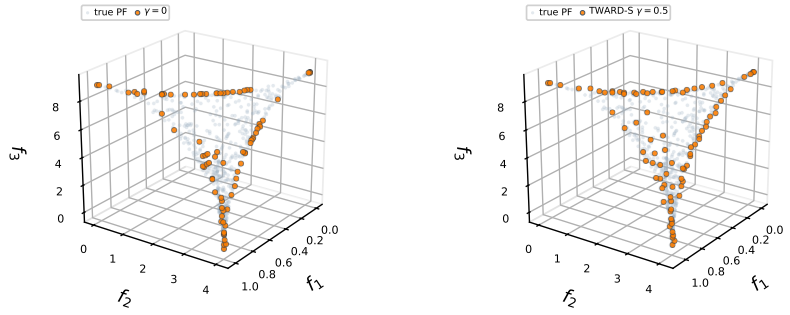
(a) γ=0 (0.363) **(b)** $\gamma^\star$ (0.210)

Figure 35: ZCAT6: repulsion off vs on (TWARD-S, N=100, T=1600).



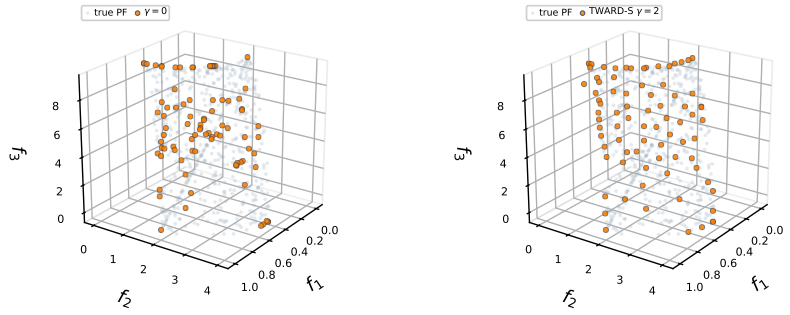
(a) γ=0 (0.445) **(b)** $\gamma^\star$ (0.386)

Figure 36: ZCAT7: repulsion off vs on (TWARD-S, N=100, T=1600).



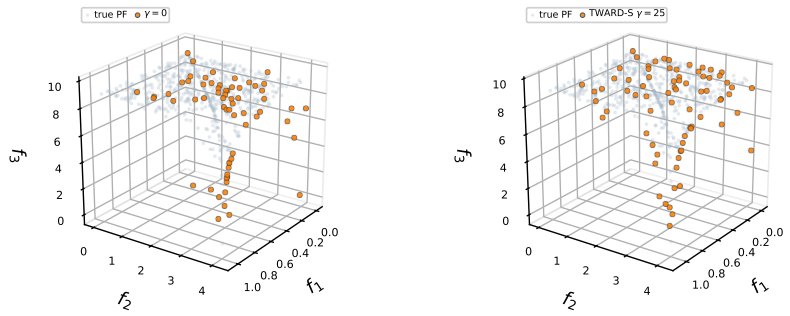
(a) γ=0 (0.225) **(b)** $\gamma^\star$ (0.182)

Figure 37: ZCAT8: repulsion off vs on (TWARD-S, N=100, T=1600).



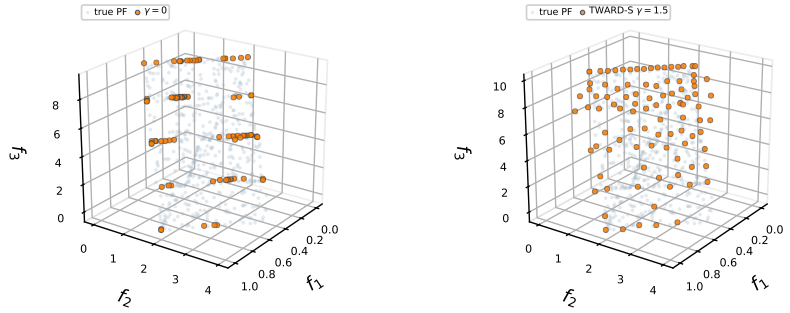
(a) γ=0 (0.347) **(b)** $\gamma^\star$ (0.268)

Figure 38: ZCAT9: repulsion off vs on (TWARD-S, N=100, T=1600).



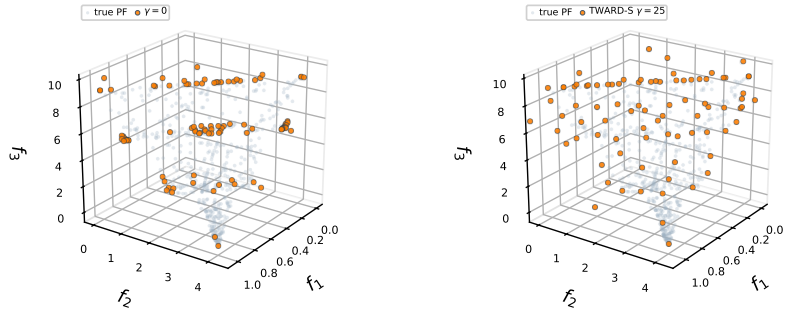
(a) γ=0 (0.230) **(b)** $\gamma^\star$ (0.184)

Figure 39: ZCAT10: repulsion off vs on (TWARD-S, N=100, T=1600).



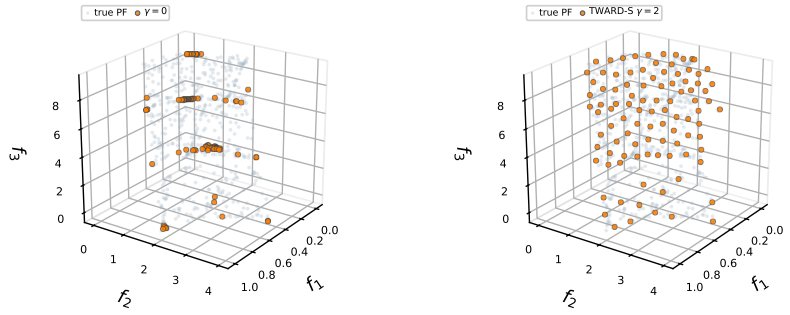
(a) γ=0 (0.496) **(b)** $\gamma^\star$ (0.249)

Figure 40: ZCAT11: repulsion off vs on (TWARD-S, N=100, T=1600).



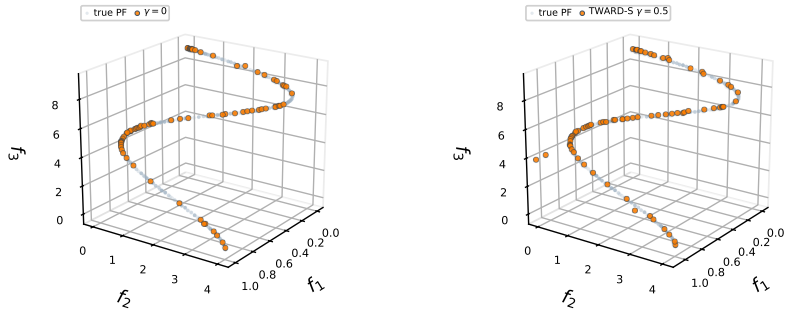
(a) γ=0 (0.486) **(b)** $\gamma^\star$ (0.309)

Figure 41: ZCAT12: repulsion off vs on (TWARD-S, N=100, T=1600).



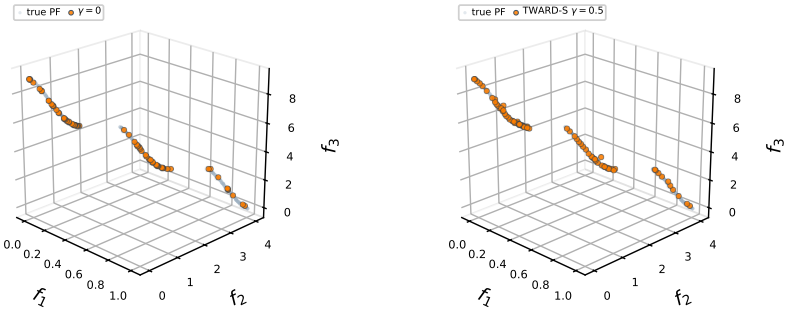
(a) γ=0 (0.549) **(b)** $\gamma^\star$ (0.215)

Figure 42: ZCAT13: repulsion off vs on (TWARD-S, N=100, T=1600).



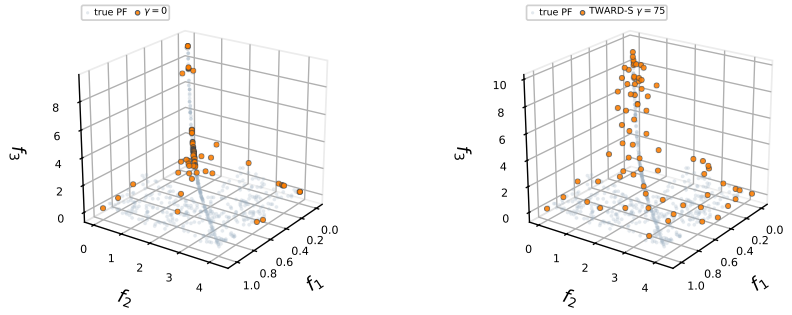
(a) γ=0 (0.108) **(b)** $\gamma^\star$ (0.088)

Figure 43: ZCAT14: repulsion off vs on (TWARD-S, N=100, T=1600).



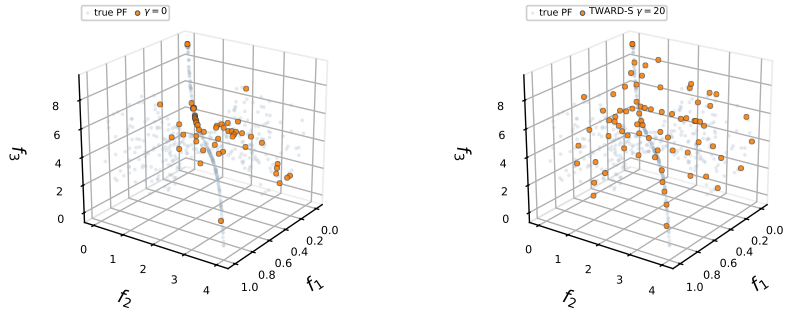
(a) γ=0 (0.082) **(b)** $\gamma^\star$ (0.057)

Figure 44: ZCAT15: repulsion off vs on (TWARD-S, N=100, T=1600).



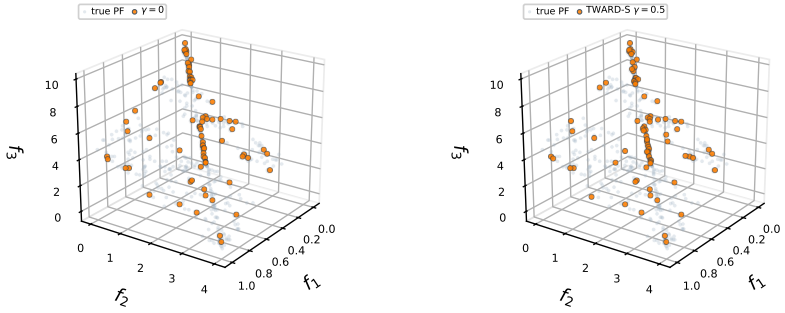
(a) γ=0 (0.453) **(b)** $\gamma^\star$ (0.206)

Figure 45: ZCAT17: repulsion off vs on (TWARD-S, N=100, T=1600).



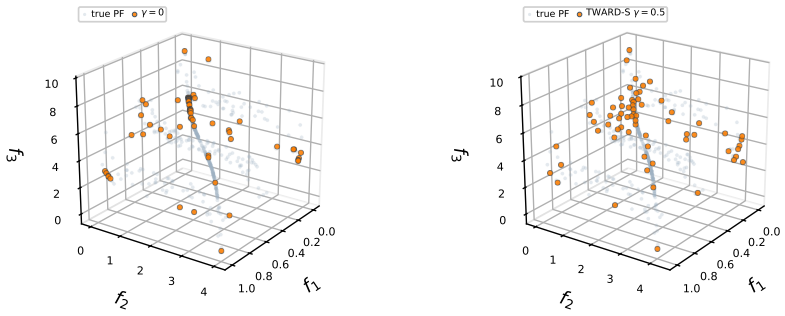
(a) γ=0 (0.373) **(b)** $\gamma^\star$ (0.208)

Figure 46: ZCAT18: repulsion off vs on (TWARD-S, N=100, T=1600).



(a) γ=0 (0.196) **(b)** $\gamma^\star$ (0.185)

Figure 47: ZCAT19: repulsion off vs on (TWARD-S, N=100, T=1600).



(a) γ=0 (0.322) **(b)** $\gamma^\star$ (0.230)

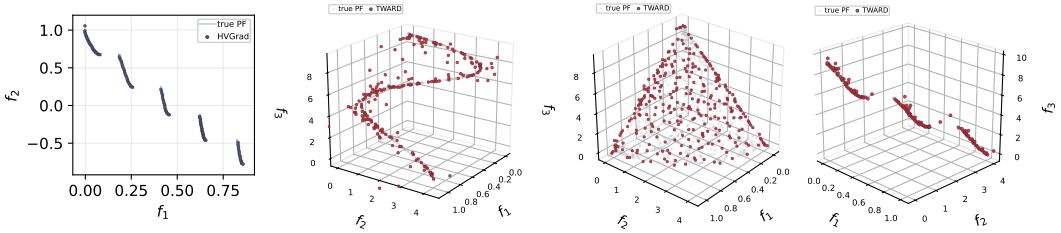
Figure 48: ZCAT20: repulsion off vs on (TWARD-S, N=100, T=1600).

D.6 Per-problem Pareto fronts, all methods

Same attained fronts as the main comparison (seed 0, matched budget), not a re-run. Every tabulated problem (33 problems: ZDT3, DTLZ2/5/6/7, WFG1–8, ZCAT1–20) gets one figure with all twelve methods, sorted by IGD, four panels per row over the true PF. The main text shows only the DTLZ7 top four; this section is the full gallery. Marker shape identifies the method; colour is shared.

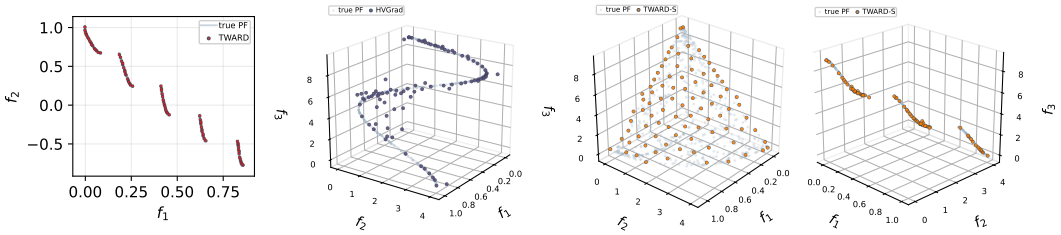
D.7 Rank-1 and rank-2 fronts

Figures 80 and 81 collect the best and second-best IGD panels across representative geometries (the main text shows the matched top-four on DTLZ7 only).



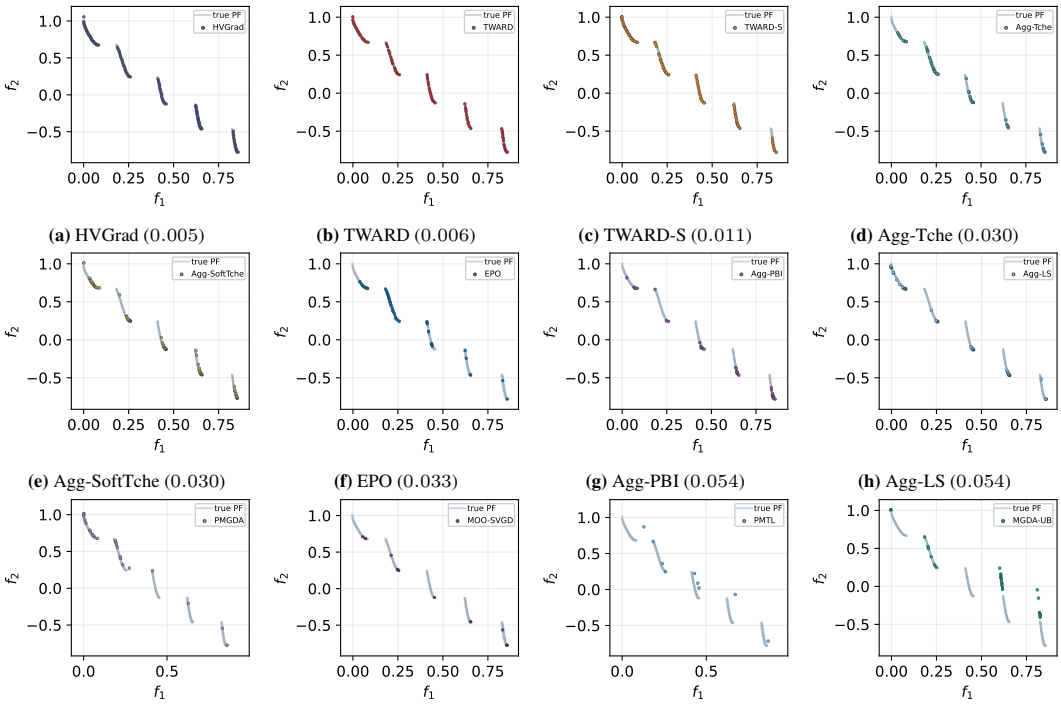
(a) ZDT3: HVGrad (0.005) **(b)** ZCAT14: TWARD (0.067) **(c)** ZCAT1: TWARD (0.137) **(d)** ZCAT15: TWARD (0.040)

Figure 80: Best IGD on each geometry (rank 1). Circles for every method; colour = identity.



(a) ZDT3: TWARD (0.006) **(b)** ZCAT14: HVGrad (0.102) **(c)** ZCAT1: TWARD-S (0.218) **(d)** ZCAT15: TWARD-S (0.072)

Figure 81: Runner-up IGD on the same geometries (rank 2).

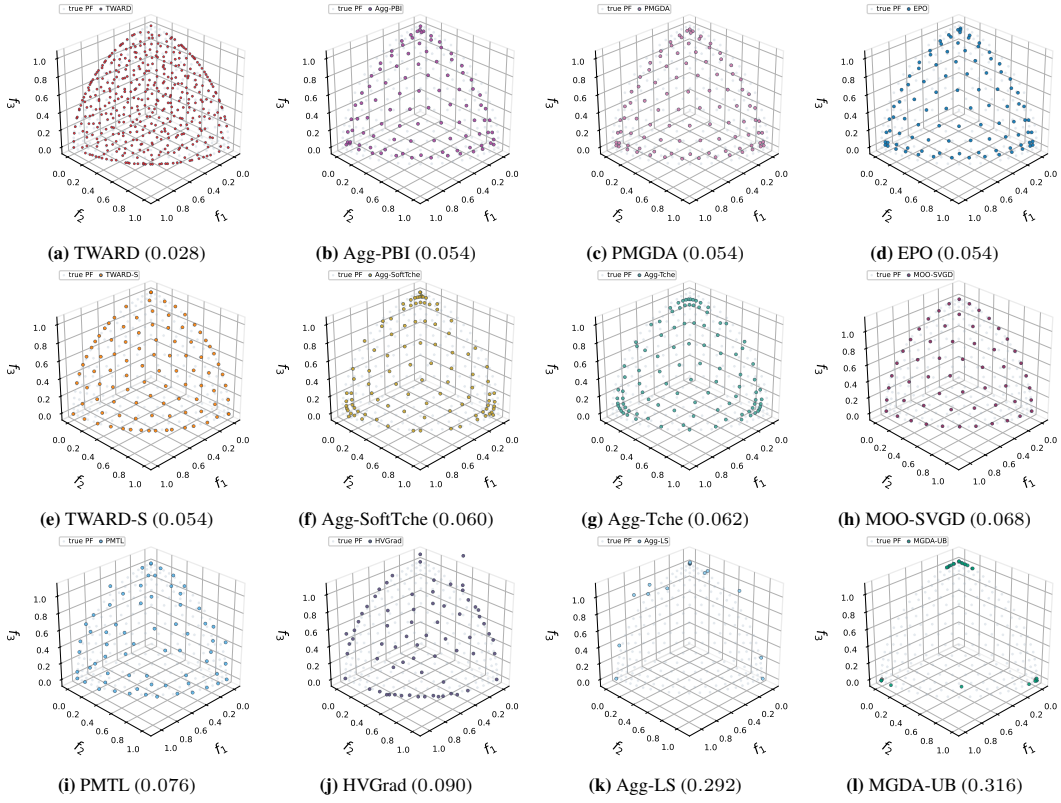


(a) HVGrad (0.005) **(b)** TWARD (0.006) **(c)** TWARD-S (0.011) **(d)** Agg-Tche (0.030)

(e) Agg-SoftTche (0.030) **(f)** EPO (0.033) **(g)** Agg-PBI (0.054) **(h)** Agg-LS (0.054)

(i) PMGDA (0.077) **(j)** MOO-SVGD (0.078) **(k)** PMTL (0.121) **(l)** MGDA-UB (0.155)

Figure 49: ZDT3: every method (sorted by IGD).

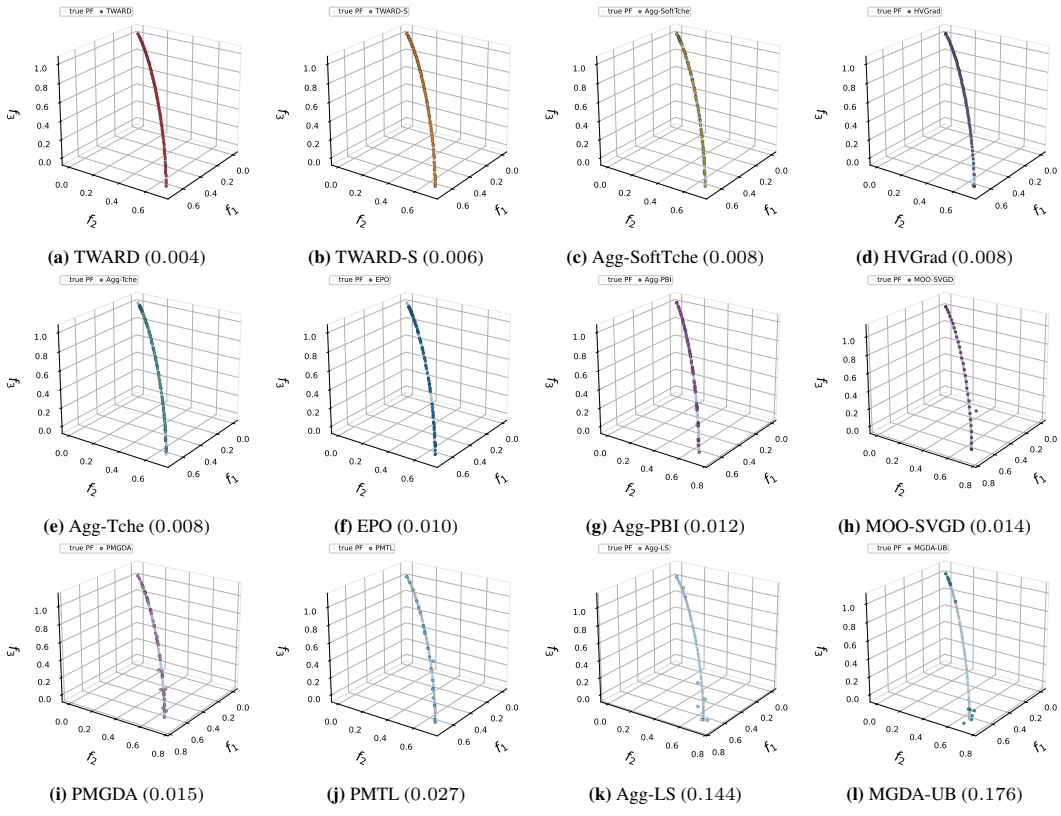


(a) TWARD (0.028) **(b)** Agg-PBI (0.054) **(c)** PMGDA (0.054) **(d)** EPO (0.054)

(e) TWARD-S (0.054) **(f)** Agg-SoftTche (0.060) **(g)** Agg-Tche (0.062) **(h)** MOO-SVGD (0.068)

(i) PMTL (0.076) **(j)** HVGrad (0.090) **(k)** Agg-LS (0.292) **(l)** MGDA-UB (0.316)

Figure 50: DTLZ2: every method (sorted by IGD).

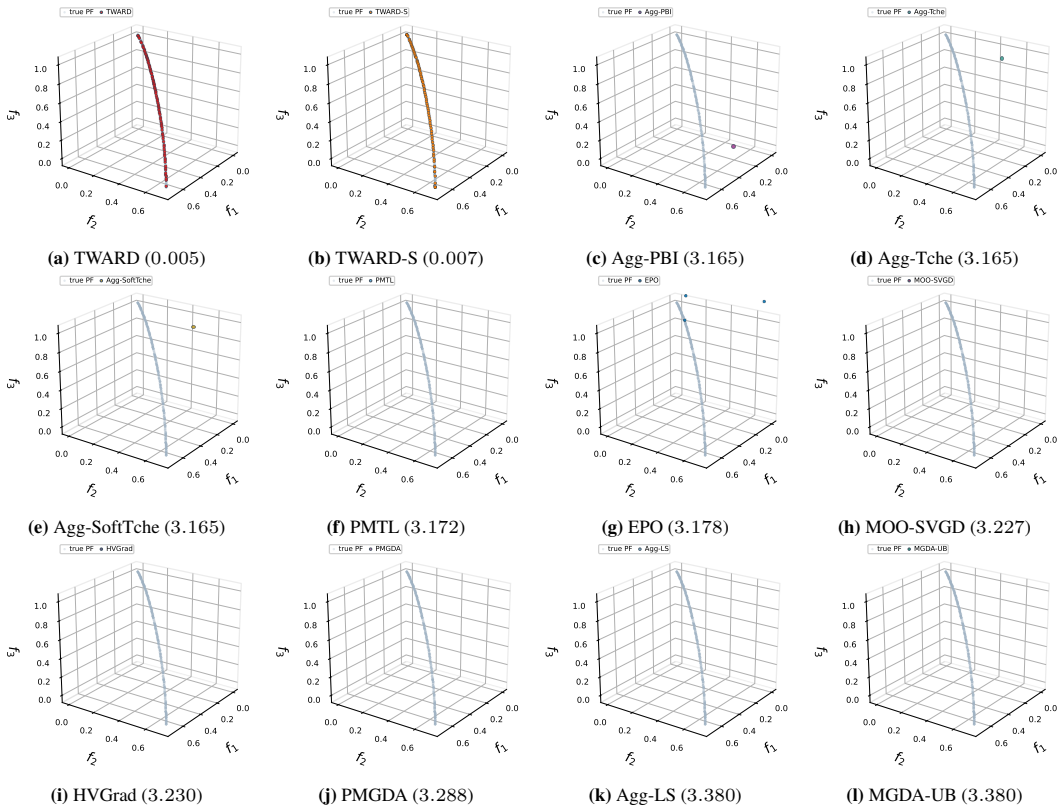


(a) TWARD (0.004) **(b)** TWARD-S (0.006) **(c)** Agg-SoftTche (0.008) **(d)** HVGrad (0.008)

(e) Agg-Tche (0.008) **(f)** EPO (0.010) **(g)** Agg-PBI (0.012) **(h)** MOO-SVGD (0.014)

(i) PMGDA (0.015) **(j)** PMTL (0.027) **(k)** Agg-LS (0.144) **(l)** MGDA-UB (0.176)

Figure 51: DTLZ5: every method (sorted by IGD).

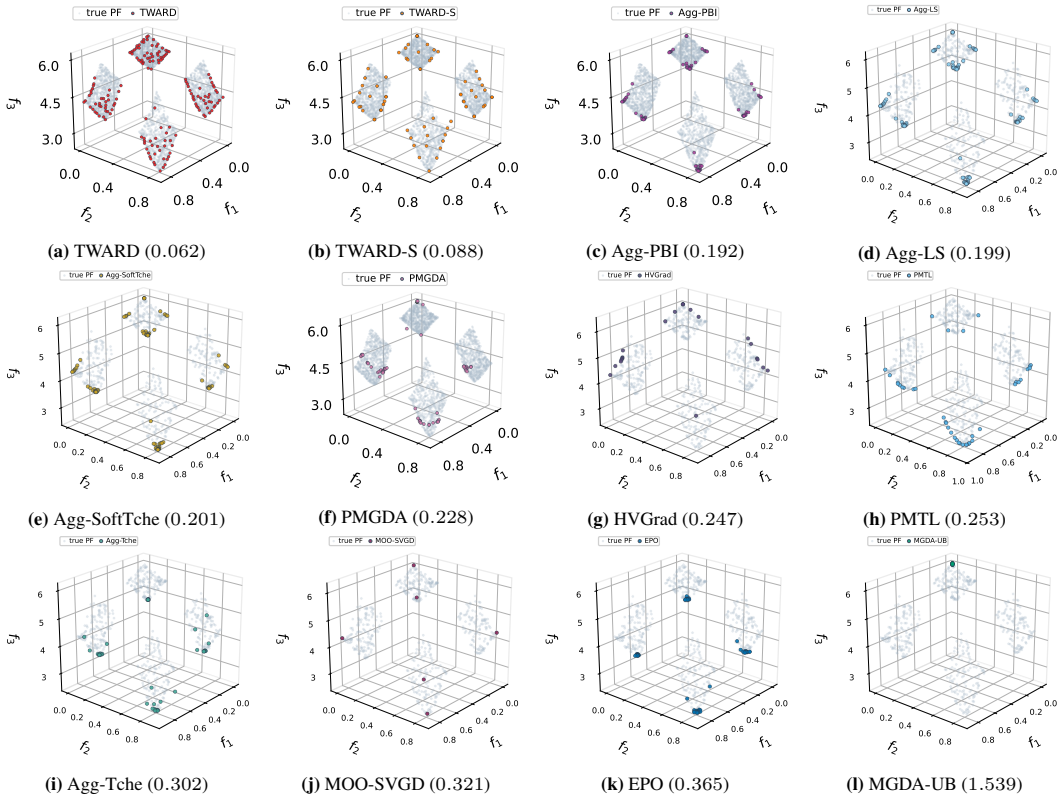


(a) TWARD (0.005) **(b)** TWARD-S (0.007) **(c)** Agg-PBI (3.165) **(d)** Agg-Tche (3.165)

(e) Agg-SoftTche (3.165) **(f)** PMTL (3.172) **(g)** EPO (3.178) **(h)** MOO-SVGD (3.227)

(i) HVGrad (3.230) **(j)** PMGDA (3.288) **(k)** Agg-LS (3.380) **(l)** MGDA-UB (3.380)

Figure 52: DTLZ6: every method (sorted by IGD).

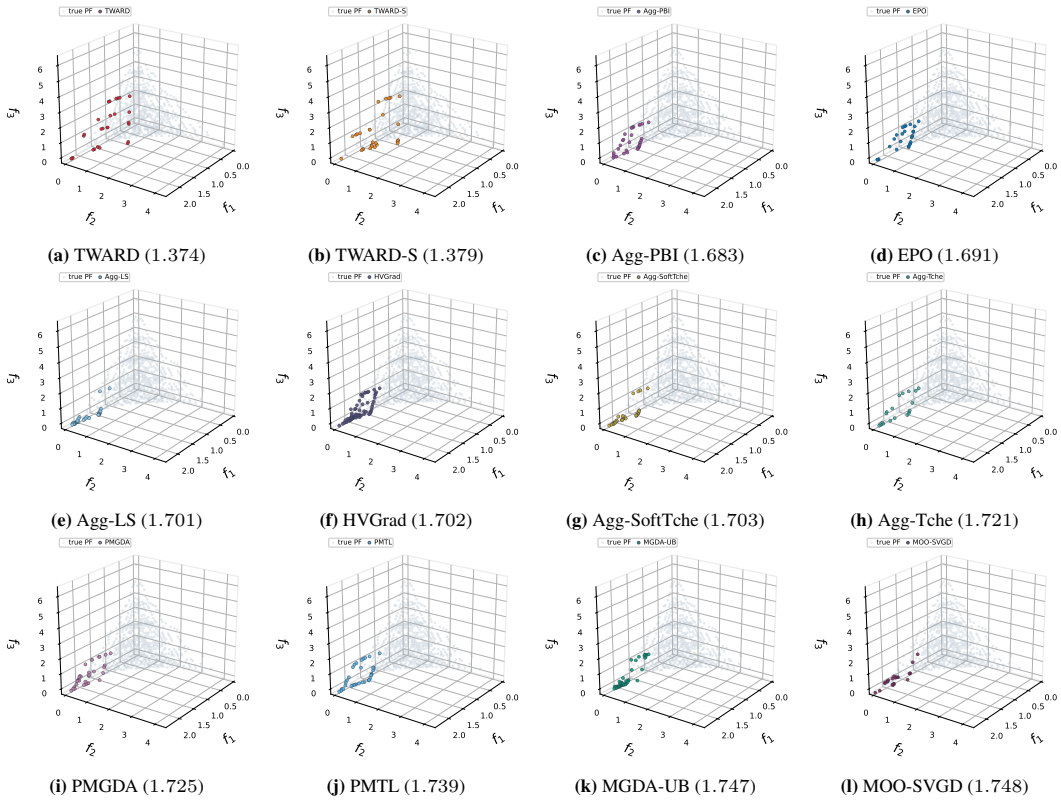


(a) TWARD (0.062) **(b)** TWARD-S (0.088) **(c)** Agg-PBI (0.192) **(d)** Agg-LS (0.199)

(e) Agg-SoftTche (0.201) **(f)** PMGDA (0.228) **(g)** HVGrad (0.247) **(h)** PMTL (0.253)

(i) Agg-Tche (0.302) **(j)** MOO-SVGD (0.321) **(k)** EPO (0.365) **(l)** MGDA-UB (1.539)

Figure 53: DTLZ7: every method (sorted by IGD).

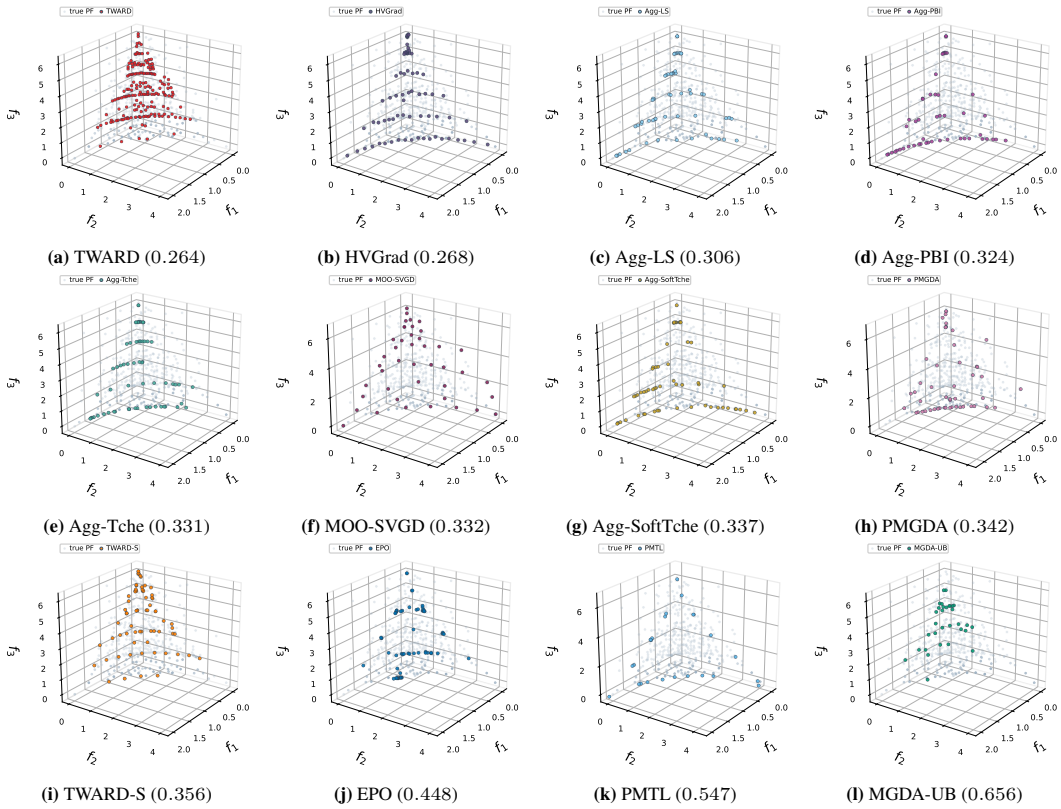


(a) TWARD (1.374) **(b)** TWARD-S (1.379) **(c)** Agg-PBI (1.683) **(d)** EPO (1.691)

(e) Agg-LS (1.701) **(f)** HVGrad (1.702) **(g)** Agg-SoftTche (1.703) **(h)** Agg-Tche (1.721)

(i) PMGDA (1.725) **(j)** PMTL (1.739) **(k)** MGDA-UB (1.747) **(l)** MOO-SVGD (1.748)

Figure 54: WFG1: every method (sorted by IGD).

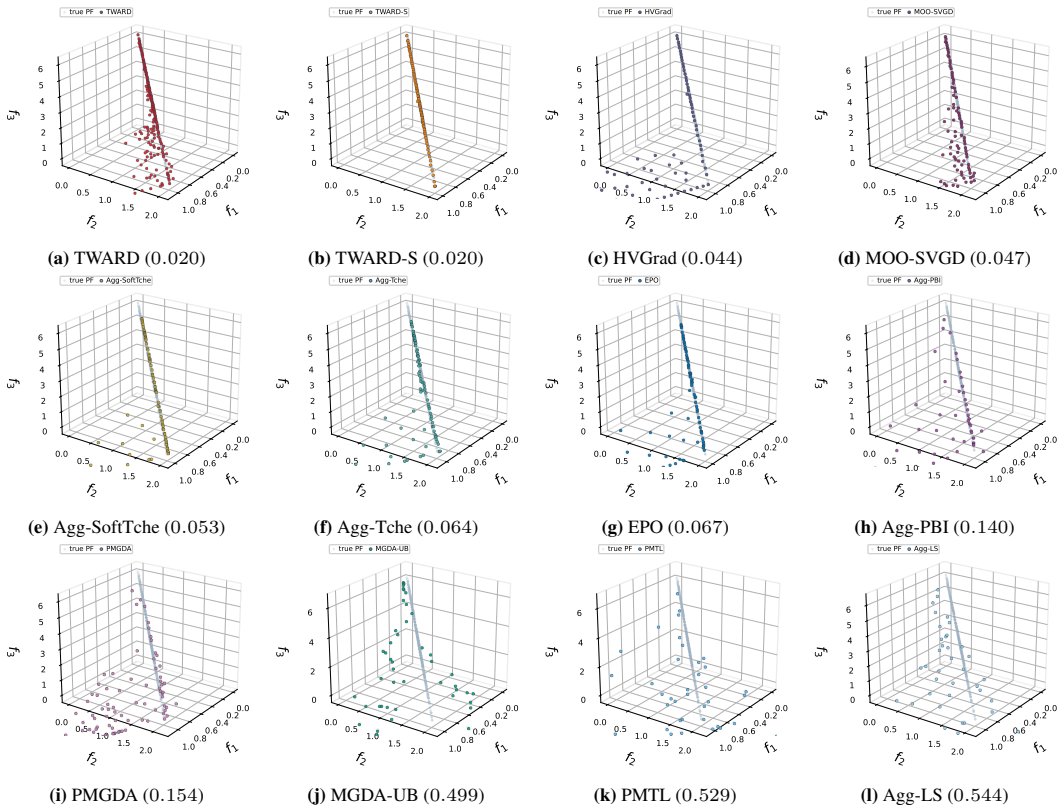


(a) TWARD (0.264) **(b)** HVGrad (0.268) **(c)** Agg-LS (0.306) **(d)** Agg-PBI (0.324)

(e) Agg-Tche (0.331) **(f)** MOO-SVGD (0.332) **(g)** Agg-SoftTche (0.337) **(h)** PMGDA (0.342)

(i) TWARD-S (0.356) **(j)** EPO (0.448) **(k)** PMTL (0.547) **(l)** MGDA-UB (0.656)

Figure 55: WFG2: every method (sorted by IGD).

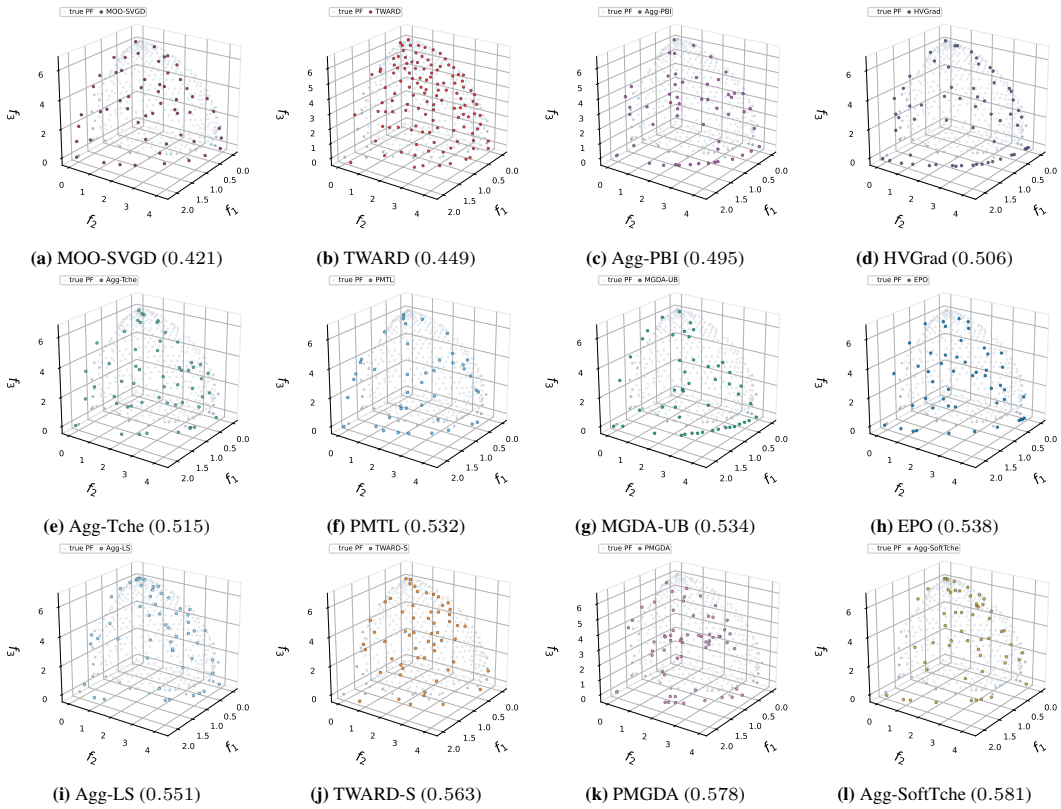


(a) TWARD (0.020) **(b)** TWARD-S (0.020) **(c)** HVGrad (0.044) **(d)** MOO-SVGD (0.047)

(e) Agg-SoftTche (0.053) **(f)** Agg-Tche (0.064) **(g)** EPO (0.067) **(h)** Agg-PBI (0.140)

(i) PMGDA (0.154) **(j)** MGDA-UB (0.499) **(k)** PMTL (0.529) **(l)** Agg-LS (0.544)

Figure 56: WFG3: every method (sorted by IGD).

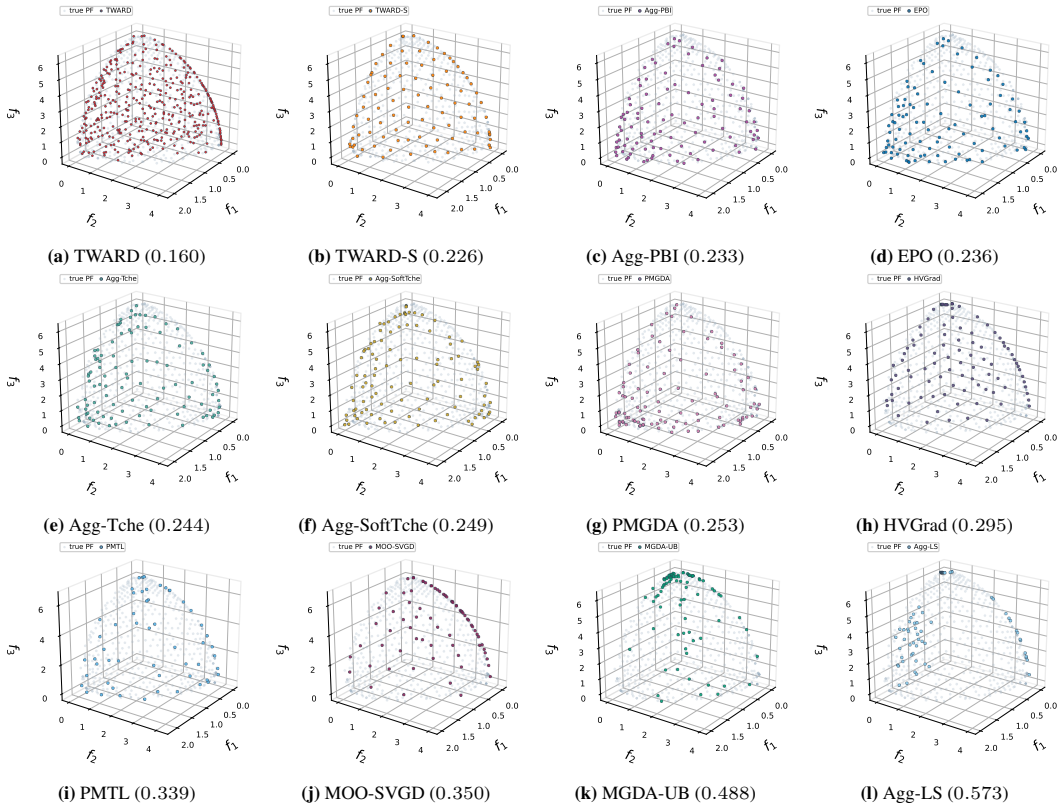


(a) MOO-SVGD (0.421) **(b)** TWARD (0.449) **(c)** Agg-PBI (0.495) **(d)** HVGrad (0.506)

(e) Agg-Tche (0.515) **(f)** PMTL (0.532) **(g)** MGDA-UB (0.534) **(h)** EPO (0.538)

(i) Agg-LS (0.551) **(j)** TWARD-S (0.563) **(k)** PMGDA (0.578) **(l)** Agg-SoftTche (0.581)

Figure 57: WFG4: every method (sorted by IGD).

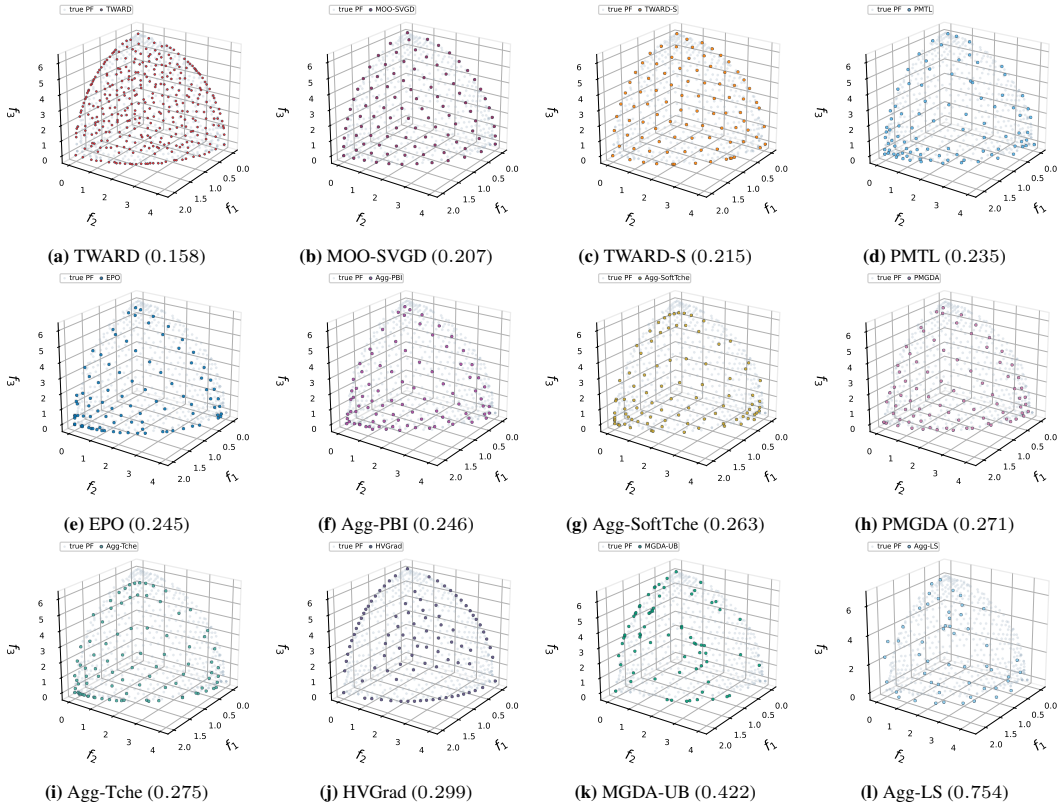


(a) TWARD (0.160) **(b)** TWARD-S (0.226) **(c)** Agg-PBI (0.233) **(d)** EPO (0.236)

(e) Agg-Tche (0.244) **(f)** Agg-SoftTche (0.249) **(g)** PMGDA (0.253) **(h)** HVGrad (0.295)

(i) PMTL (0.339) **(j)** MOO-SVGD (0.350) **(k)** MGDA-UB (0.488) **(l)** Agg-LS (0.573)

Figure 58: WFG5: every method (sorted by IGD).

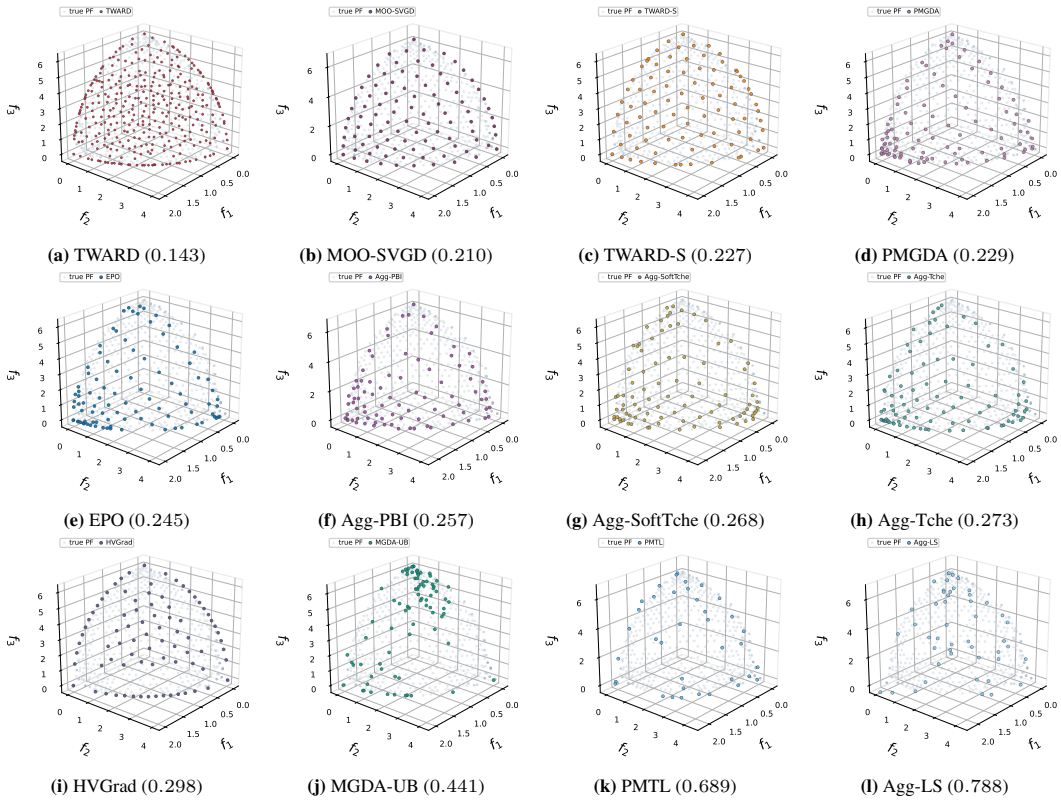


(a) TWARD (0.158) **(b)** MOO-SVGD (0.207) **(c)** TWARD-S (0.215) **(d)** PMTL (0.235)

(e) EPO (0.245) **(f)** Agg-PBI (0.246) **(g)** Agg-SoftTche (0.263) **(h)** PMGDA (0.271)

(i) Agg-Tche (0.275) **(j)** HVGrad (0.299) **(k)** MGDA-UB (0.422) **(l)** Agg-LS (0.754)

Figure 59: WFG6: every method (sorted by IGD).

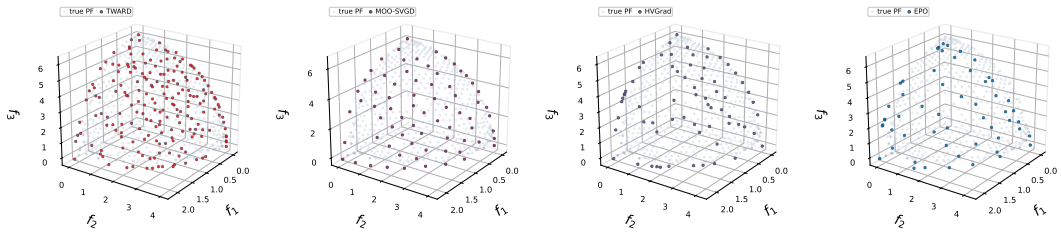


(a) TWARD (0.143) **(b)** MOO-SVGD (0.210) **(c)** TWARD-S (0.227) **(d)** PMGDA (0.229)

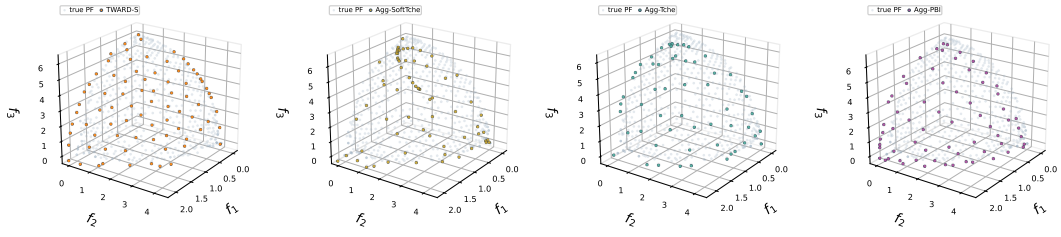
(e) EPO (0.245) **(f)** Agg-PBI (0.257) **(g)** Agg-SoftTche (0.268) **(h)** Agg-Tche (0.273)

(i) HVGrad (0.298) **(j)** MGDA-UB (0.441) **(k)** PMTL (0.689) **(l)** Agg-LS (0.788)

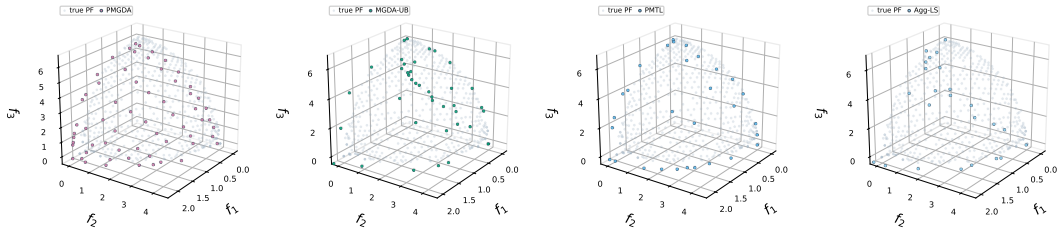
Figure 60: WFG7: every method (sorted by IGD).



(a) TWARD (0.300) **(b)** MOO-SVGD (0.375) **(c)** HVGrad (0.394) **(d)** EPO (0.418)

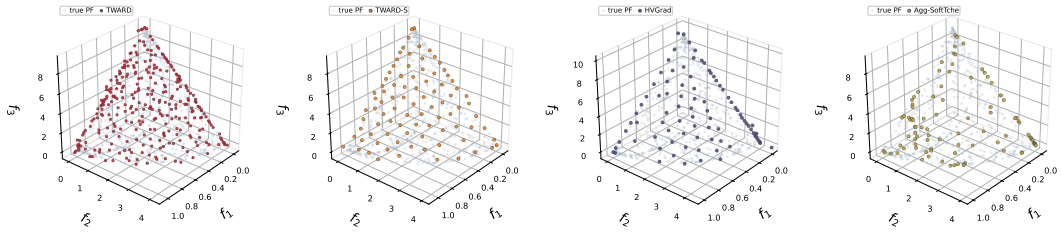


(e) TWARD-S (0.427) **(f)** Agg-SoftTche (0.428) **(g)** Agg-Tche (0.431) **(h)** Agg-PBI (0.435)

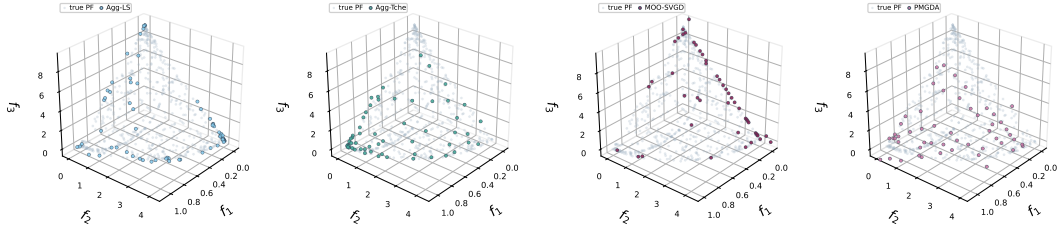


(i) PMGDA (0.437) **(j)** MGDA-UB (0.587) **(k)** PMTL (0.781) **(l)** Agg-LS (0.806)

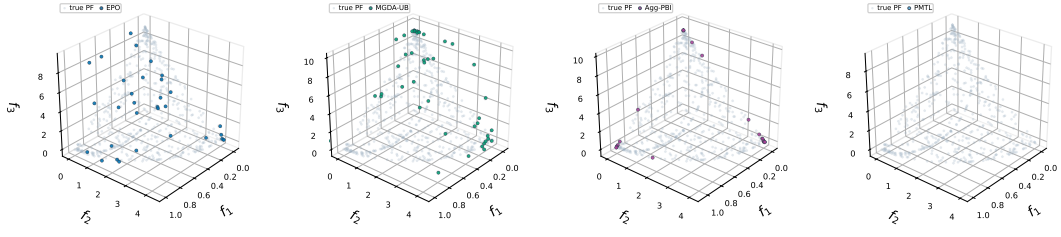
Figure 61: WFG8: every method (sorted by IGD).



(a) TWARD (0.137) **(b)** TWARD-S (0.218) **(c)** HVGrad (0.243) **(d)** Agg-SoftTche (0.258)

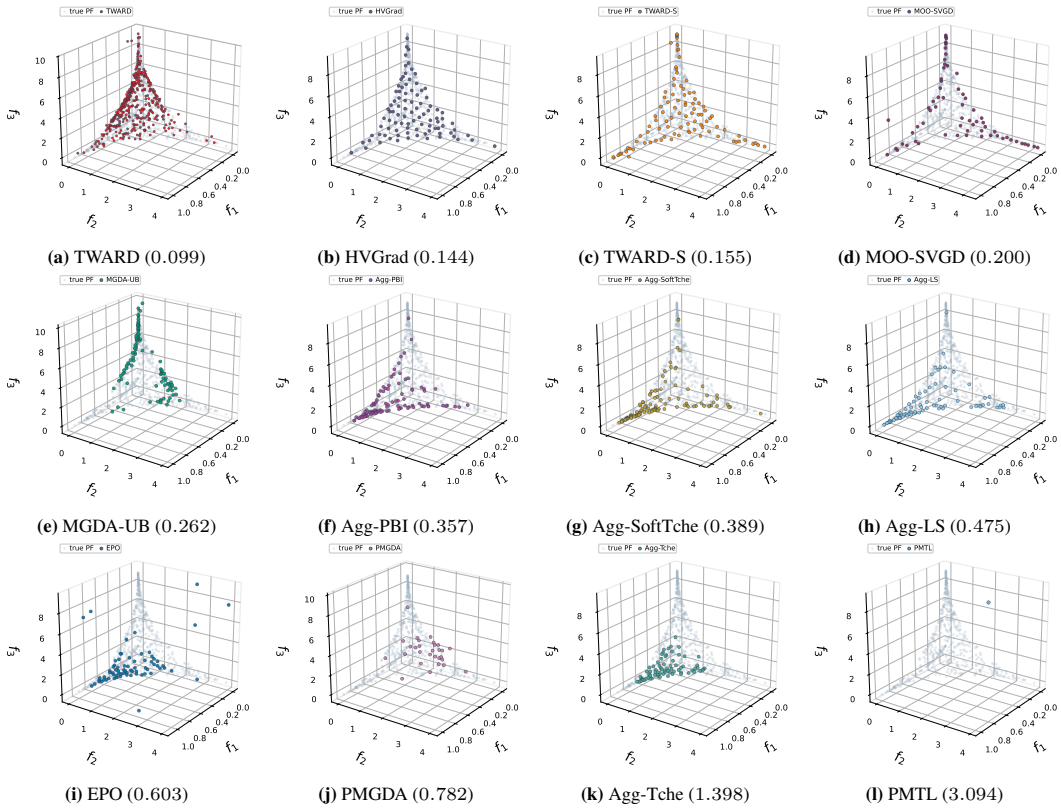


(e) Agg-LS (0.441) **(f)** Agg-Tche (0.477) **(g)** MOO-SVGD (0.494) **(h)** PMGDA (0.526)



(i) EPO (0.598) **(j)** MGDA-UB (0.780) **(k)** Agg-PBI (0.895) **(l)** PMTL (2.231)

Figure 62: ZCAT1: every method (sorted by IGD).

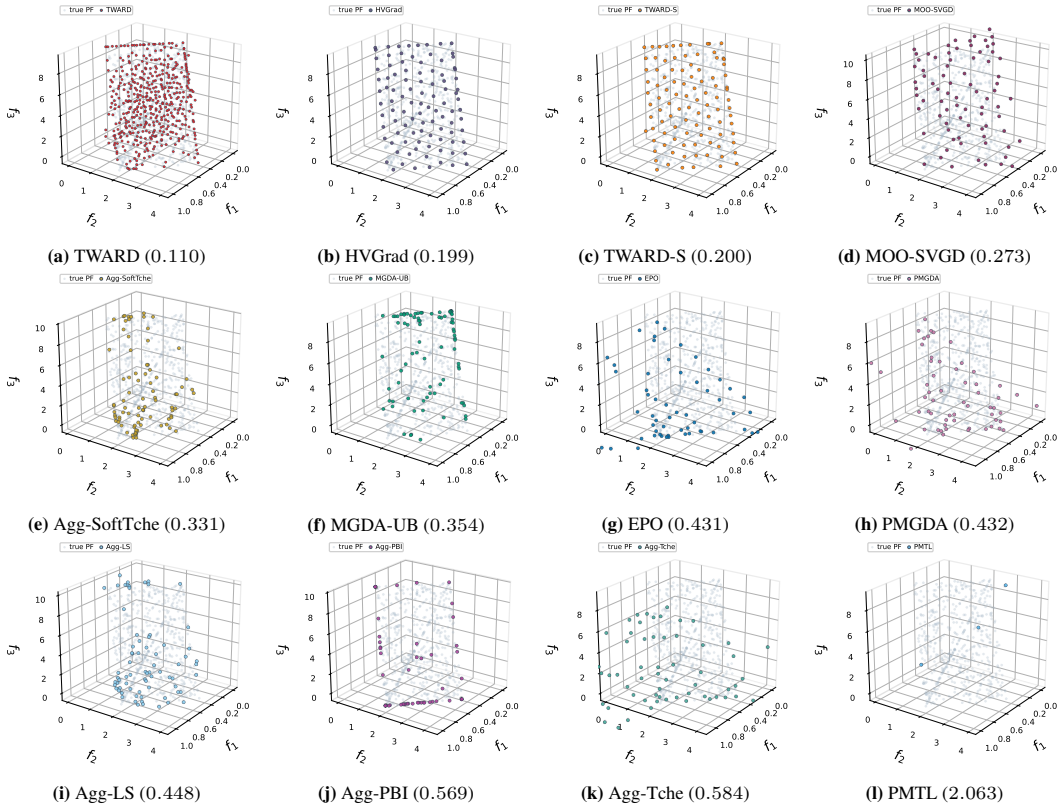


(a) TWARD (0.099) **(b)** HVGrad (0.144) **(c)** TWARD-S (0.155) **(d)** MOO-SVGD (0.200)

(e) MGDA-UB (0.262) **(f)** Agg-PBI (0.357) **(g)** Agg-SoftTche (0.389) **(h)** Agg-LS (0.475)

(i) EPO (0.603) **(j)** PMGDA (0.782) **(k)** Agg-Tche (1.398) **(l)** PMTL (3.094)

Figure 63: ZCAT2: every method (sorted by IGD).

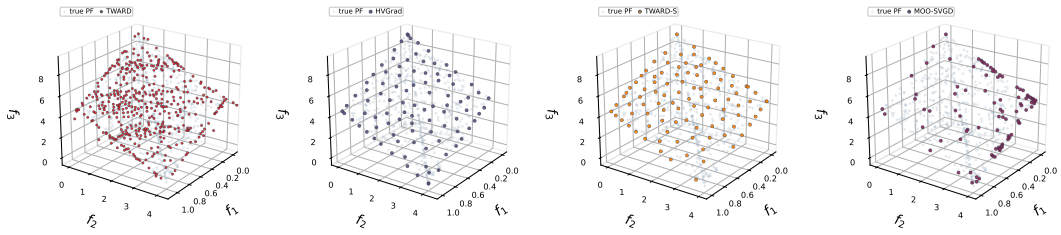


(a) TWARD (0.110) **(b)** HVGrad (0.199) **(c)** TWARD-S (0.200) **(d)** MOO-SVGD (0.273)

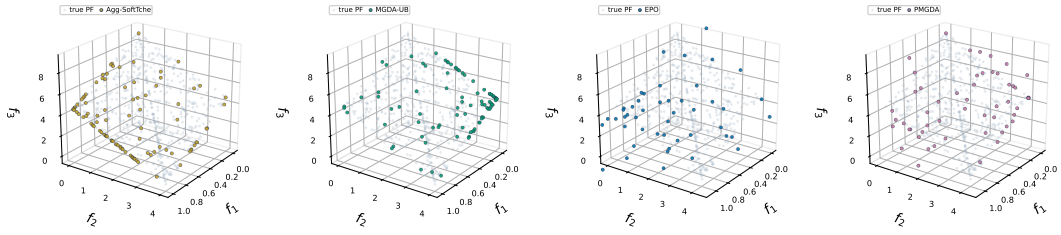
(e) Agg-SoftTche (0.331) **(f)** MGDA-UB (0.354) **(g)** EPO (0.431) **(h)** PMGDA (0.432)

(i) Agg-LS (0.448) **(j)** Agg-PBI (0.569) **(k)** Agg-Tche (0.584) **(l)** PMTL (2.063)

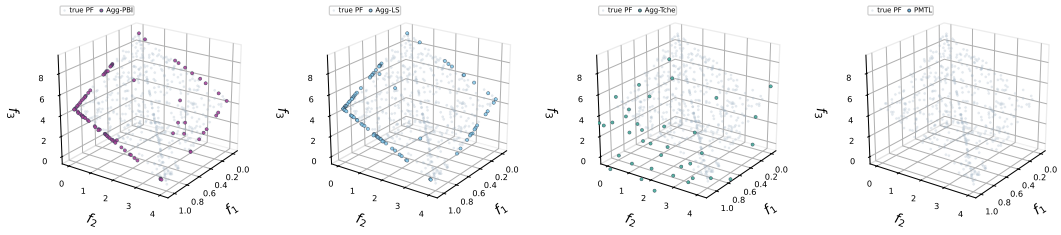
Figure 64: ZCAT3: every method (sorted by IGD).



(a) TWARD (0.120) **(b)** HVGrad (0.207) **(c)** TWARD-S (0.209) **(d)** MOO-SVGD (0.321)

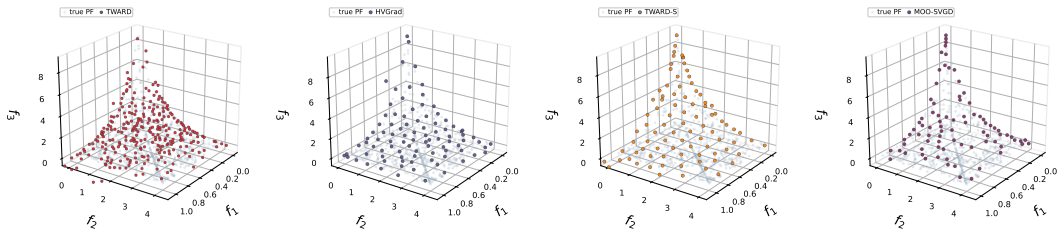


(e) Agg-SoftTche (0.338) **(f)** MGDA-UB (0.360) **(g)** EPO (0.443) **(h)** PMGDA (0.462)

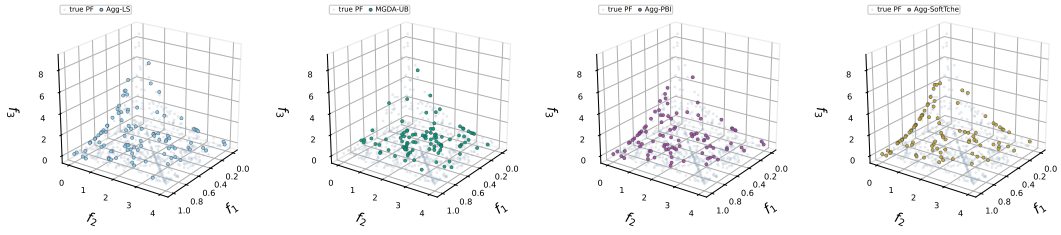


(i) Agg-PBI (0.496) **(j)** Agg-LS (0.571) **(k)** Agg-Tche (0.676) **(l)** PMTL (3.499)

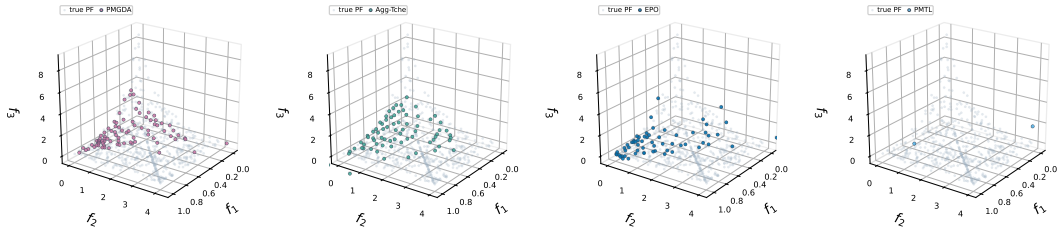
Figure 65: ZCAT4: every method (sorted by IGD).



(a) TWARD (0.108) **(b)** HVGrad (0.149) **(c)** TWARD-S (0.166) **(d)** MOO-SVGD (0.174)

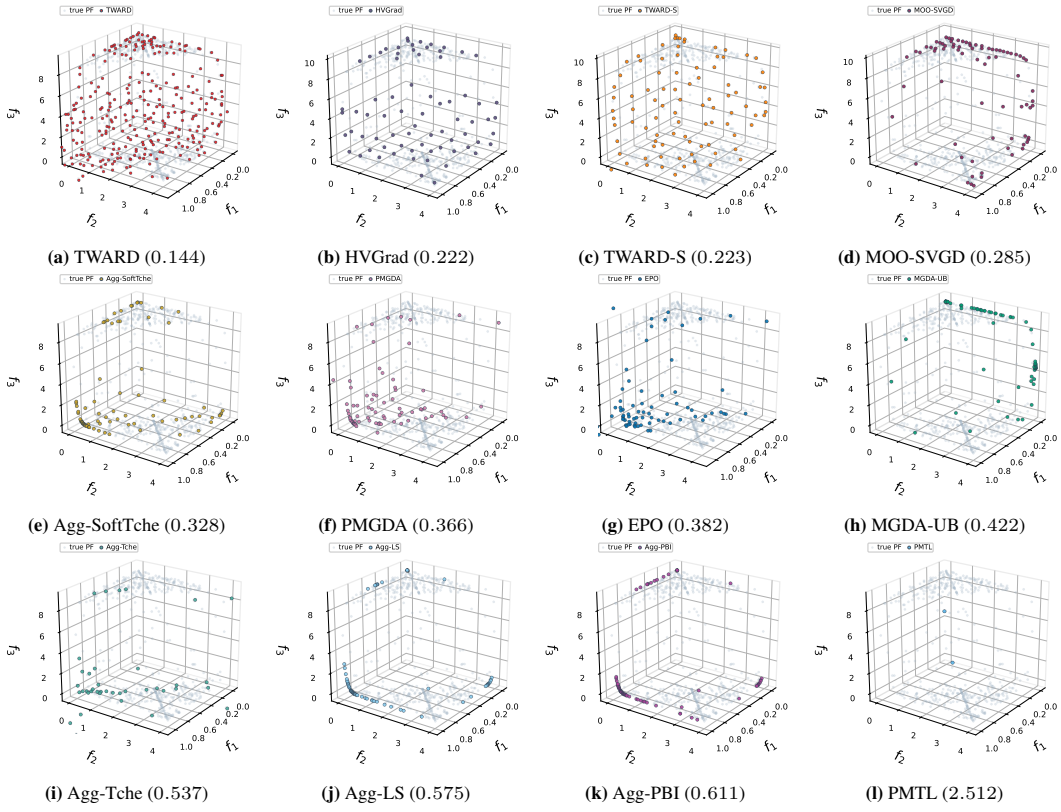


(e) Agg-LS (0.200) **(f)** MGDA-UB (0.211) **(g)** Agg-PBI (0.231) **(h)** Agg-SoftTche (0.250)



(i) PMGDA (0.354) **(j)** Agg-Tche (0.380) **(k)** EPO (0.421) **(l)** PMTL (2.499)

Figure 66: ZCAT5: every method (sorted by IGD).

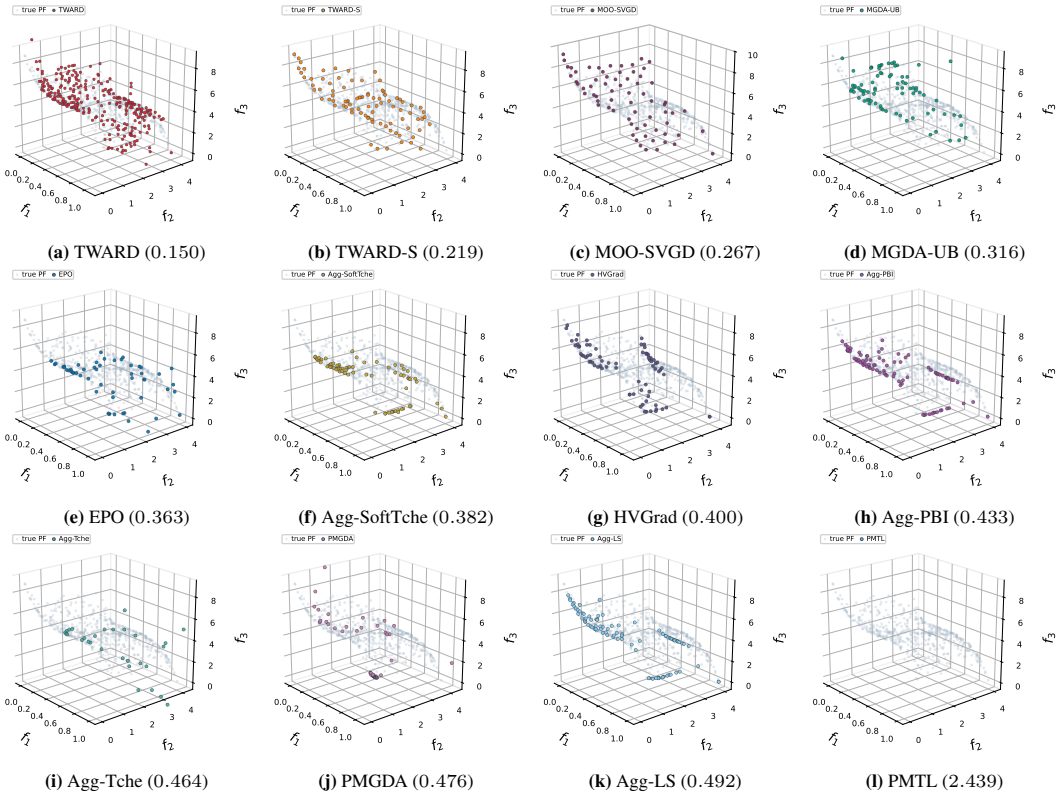


(a) TWARD (0.144) **(b)** HVGrad (0.222) **(c)** TWARD-S (0.223) **(d)** MOO-SVGD (0.285)

(e) Agg-SoftTche (0.328) **(f)** PMGDA (0.366) **(g)** EPO (0.382) **(h)** MGDA-UB (0.422)

(i) Agg-Tche (0.537) **(j)** Agg-LS (0.575) **(k)** Agg-PBI (0.611) **(l)** PMTL (2.512)

Figure 67: ZCAT6: every method (sorted by IGD).

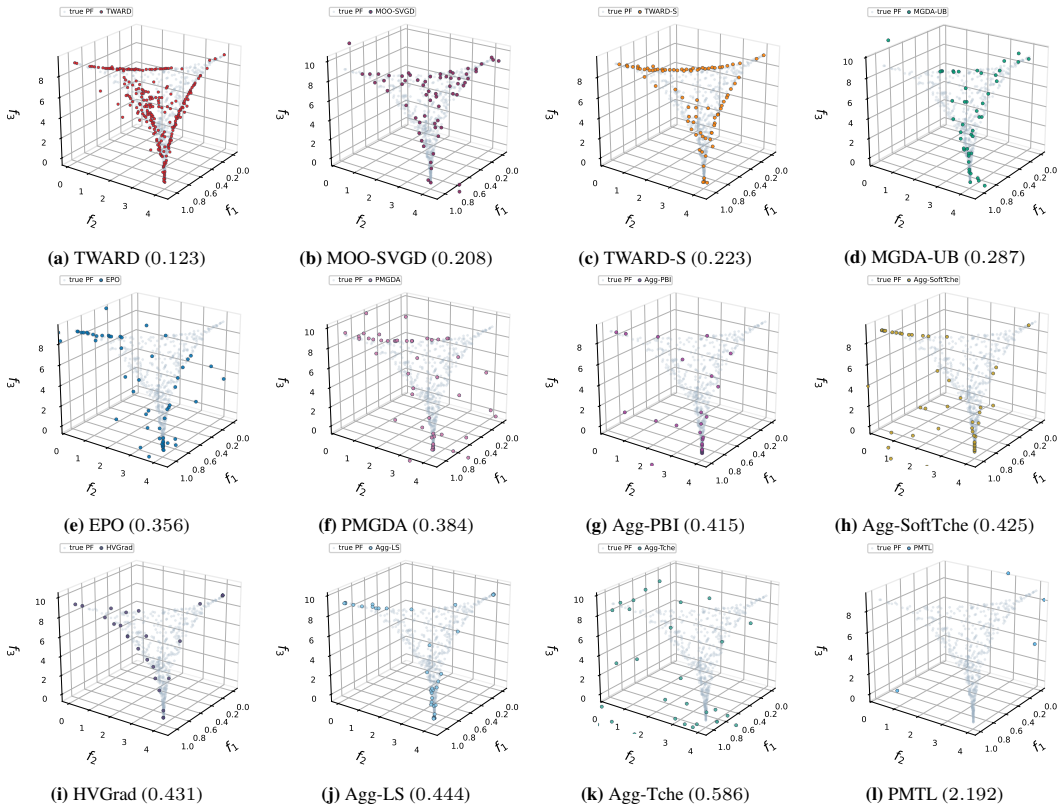


(a) TWARD (0.150) **(b)** TWARD-S (0.219) **(c)** MOO-SVGD (0.267) **(d)** MGDA-UB (0.316)

(e) EPO (0.363) **(f)** Agg-SoftTche (0.382) **(g)** HVGrad (0.400) **(h)** Agg-PBI (0.433)

(i) Agg-Tche (0.464) **(j)** PMGDA (0.476) **(k)** Agg-LS (0.492) **(l)** PMTL (2.439)

Figure 68: ZCAT7: every method (sorted by IGD).

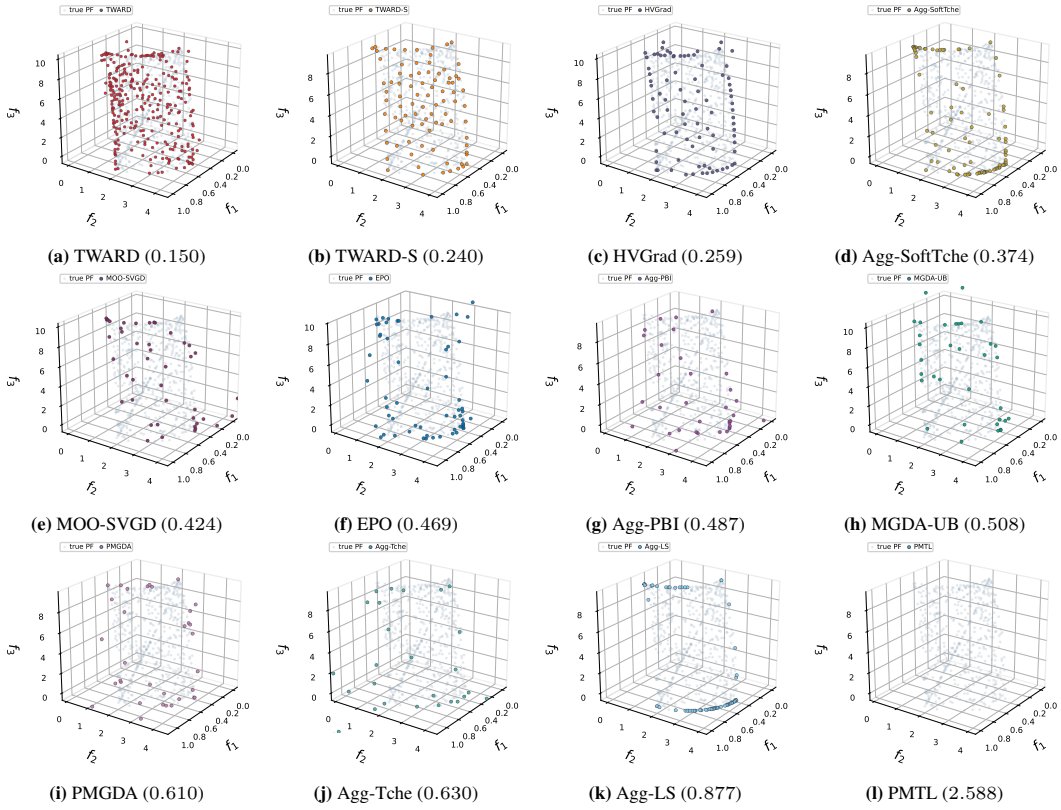


(a) TWARD (0.123) **(b)** MOO-SVGD (0.208) **(c)** TWARD-S (0.223) **(d)** MGDA-UB (0.287)

(e) EPO (0.356) **(f)** PMGDA (0.384) **(g)** Agg-PBI (0.415) **(h)** Agg-SoftTche (0.425)

(i) HVGrad (0.431) **(j)** Agg-LS (0.444) **(k)** Agg-Tche (0.586) **(l)** PMTL (2.192)

Figure 69: ZCAT8: every method (sorted by IGD).

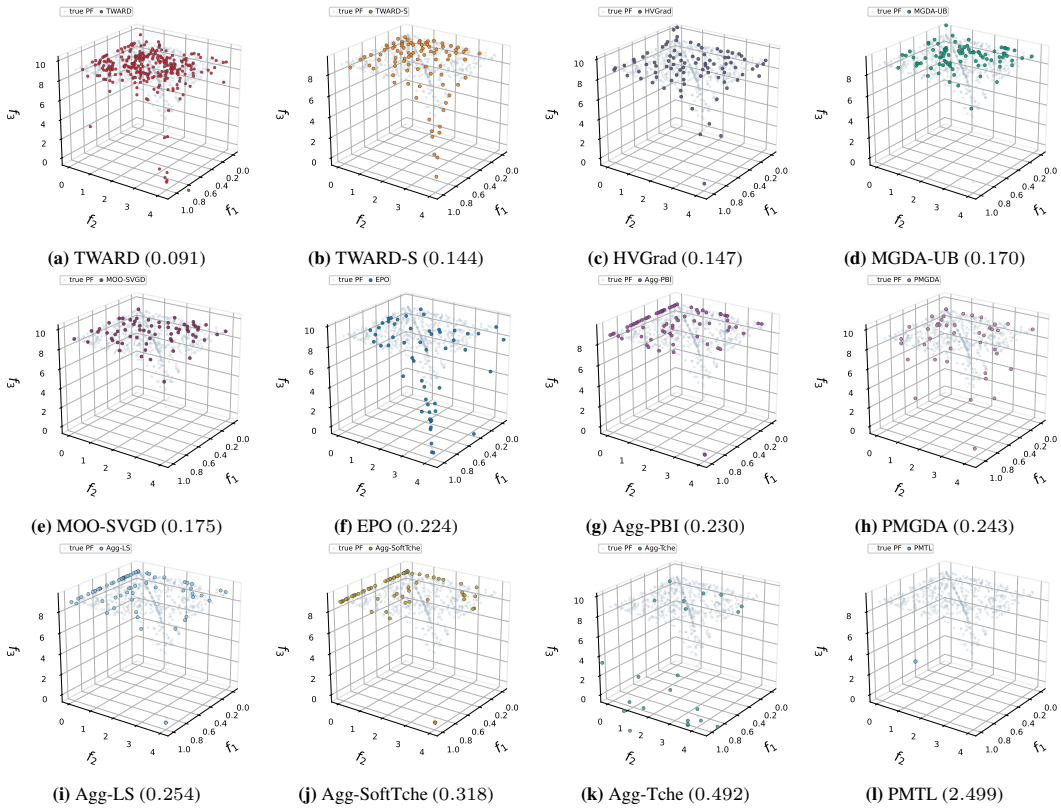


(a) TWARD (0.150) **(b)** TWARD-S (0.240) **(c)** HVGrad (0.259) **(d)** Agg-SoftTche (0.374)

(e) MOO-SVGD (0.424) **(f)** EPO (0.469) **(g)** Agg-PBI (0.487) **(h)** MGDA-UB (0.508)

(i) PMGDA (0.610) **(j)** Agg-Tche (0.630) **(k)** Agg-LS (0.877) **(l)** PMTL (2.588)

Figure 70: ZCAT9: every method (sorted by IGD).

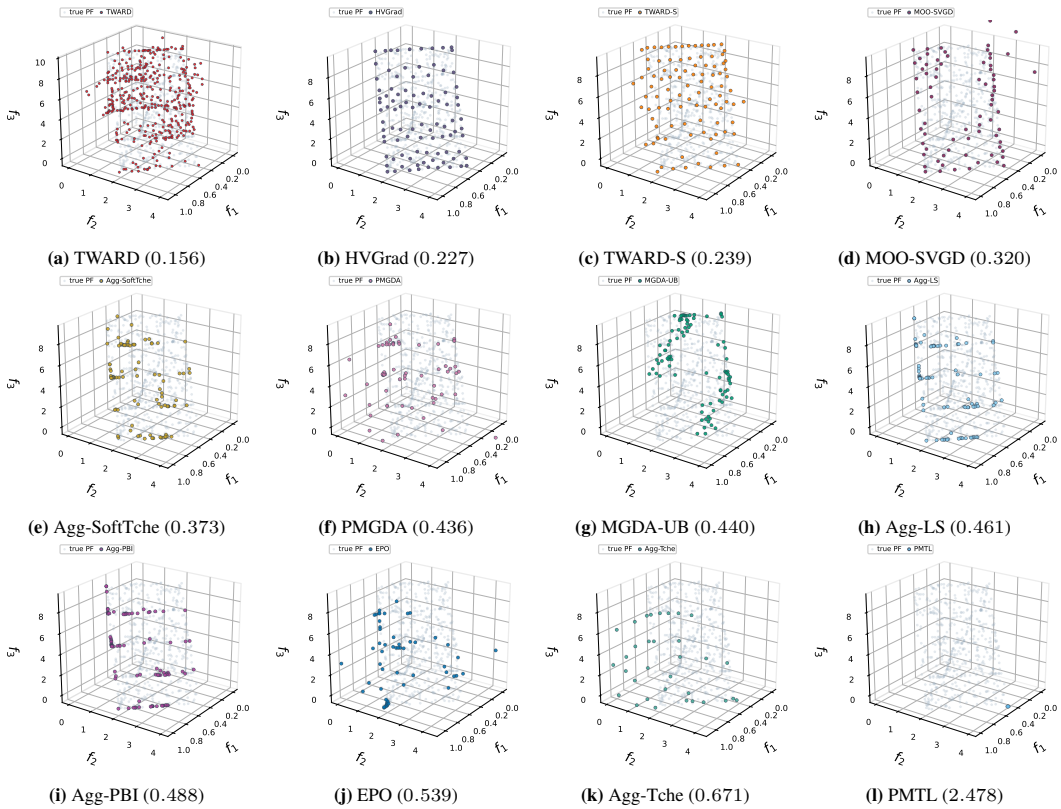


(a) TWARD (0.091) **(b)** TWARD-S (0.144) **(c)** HVGrad (0.147) **(d)** MGDA-UB (0.170)

(e) MOO-SVGD (0.175) **(f)** EPO (0.224) **(g)** Agg-PBI (0.230) **(h)** PMGDA (0.243)

(i) Agg-LS (0.254) **(j)** Agg-SoftTche (0.318) **(k)** Agg-Tche (0.492) **(l)** PMTL (2.499)

Figure 71: ZCAT10: every method (sorted by IGD).

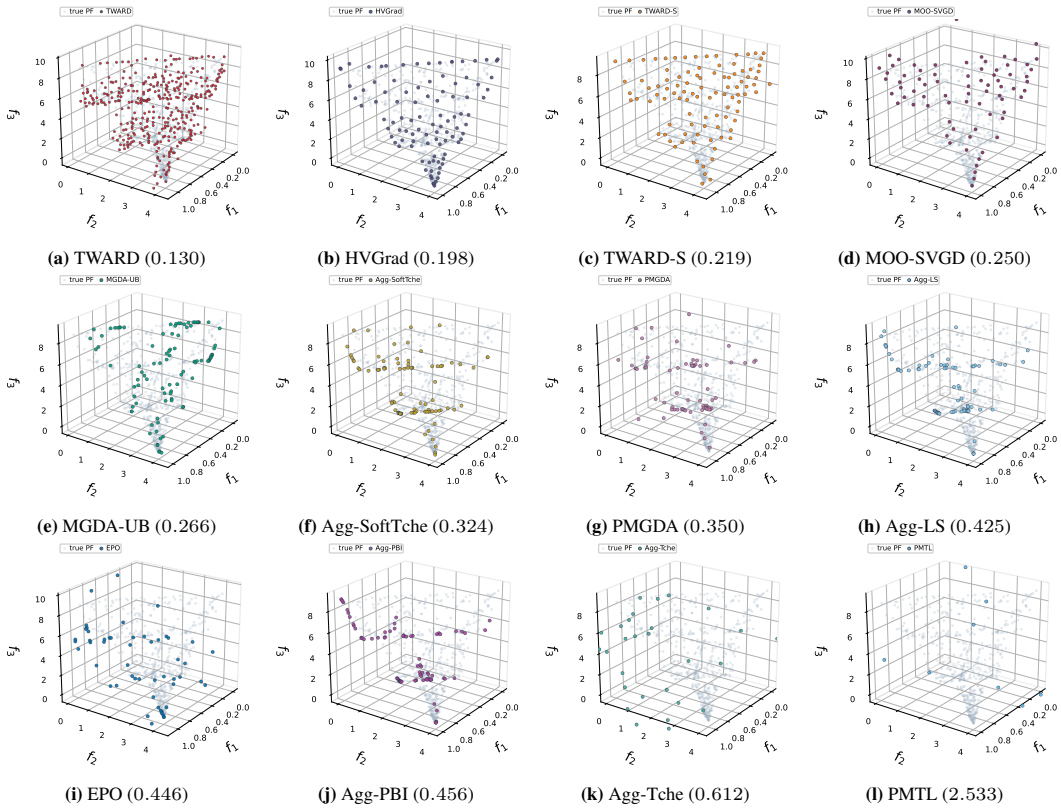


(a) TWARD (0.156) **(b)** HVGrad (0.227) **(c)** TWARD-S (0.239) **(d)** MOO-SVGD (0.320)

(e) Agg-SoftTche (0.373) **(f)** PMGDA (0.436) **(g)** MGDA-UB (0.440) **(h)** Agg-LS (0.461)

(i) Agg-PBI (0.488) **(j)** EPO (0.539) **(k)** Agg-Tche (0.671) **(l)** PMTL (2.478)

Figure 72: ZCAT11: every method (sorted by IGD).

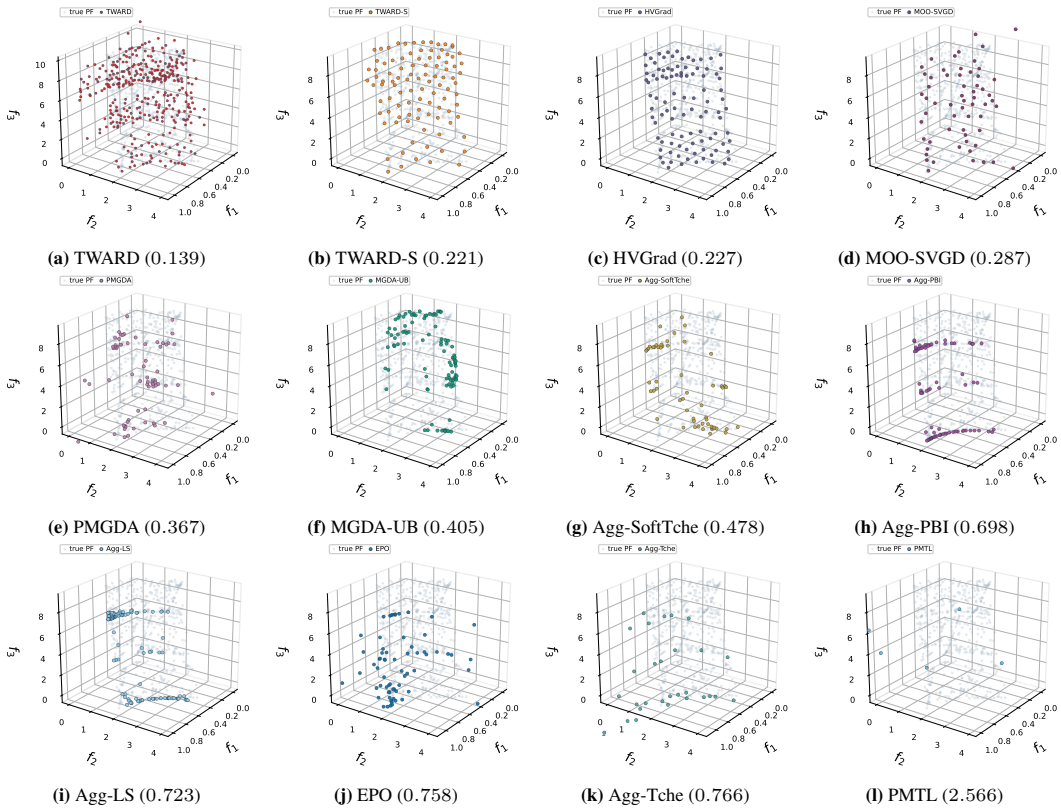


(a) TWARD (0.130) **(b)** HVGrad (0.198) **(c)** TWARD-S (0.219) **(d)** MOO-SVGD (0.250)

(e) MGDA-UB (0.266) **(f)** Agg-SoftTche (0.324) **(g)** PMGDA (0.350) **(h)** Agg-LS (0.425)

(i) EPO (0.446) **(j)** Agg-PBI (0.456) **(k)** Agg-Tche (0.612) **(l)** PMTL (2.533)

Figure 73: ZCAT12: every method (sorted by IGD).

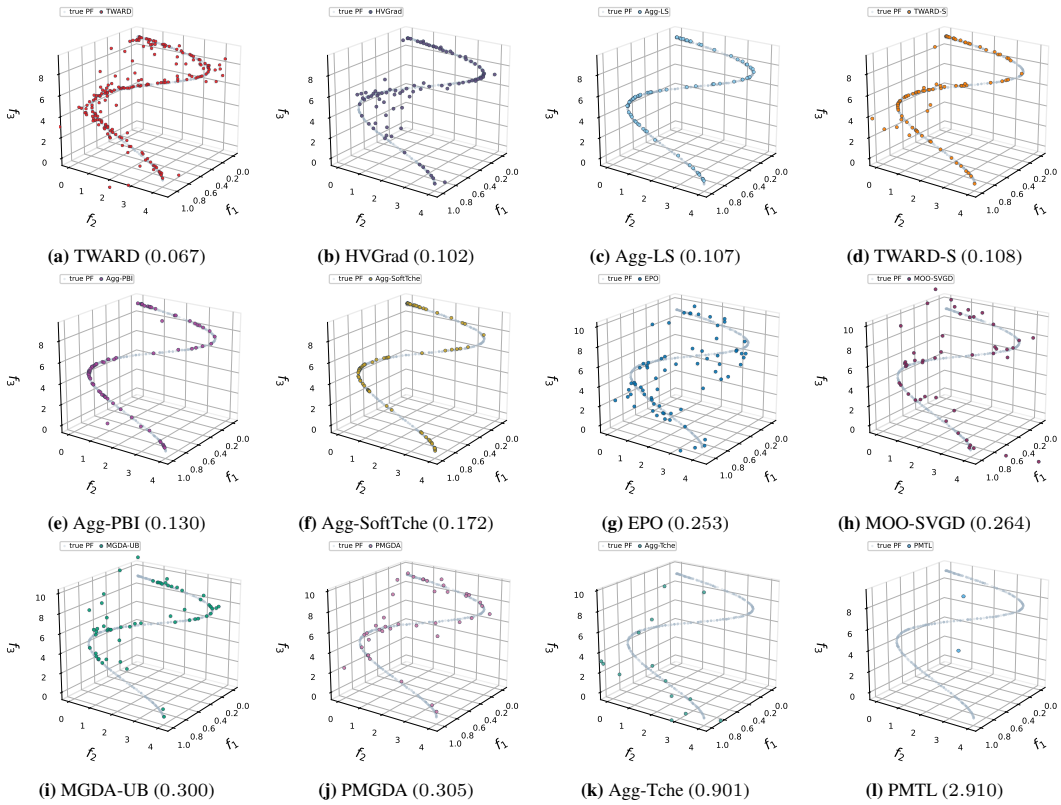


(a) TWARD (0.139) **(b)** TWARD-S (0.221) **(c)** HVGrad (0.227) **(d)** MOO-SVGD (0.287)

(e) PMGDA (0.367) **(f)** MGDA-UB (0.405) **(g)** Agg-SoftTche (0.478) **(h)** Agg-PBI (0.698)

(i) Agg-LS (0.723) **(j)** EPO (0.758) **(k)** Agg-Tche (0.766) **(l)** PMTL (2.566)

Figure 74: ZCAT13: every method (sorted by IGD).

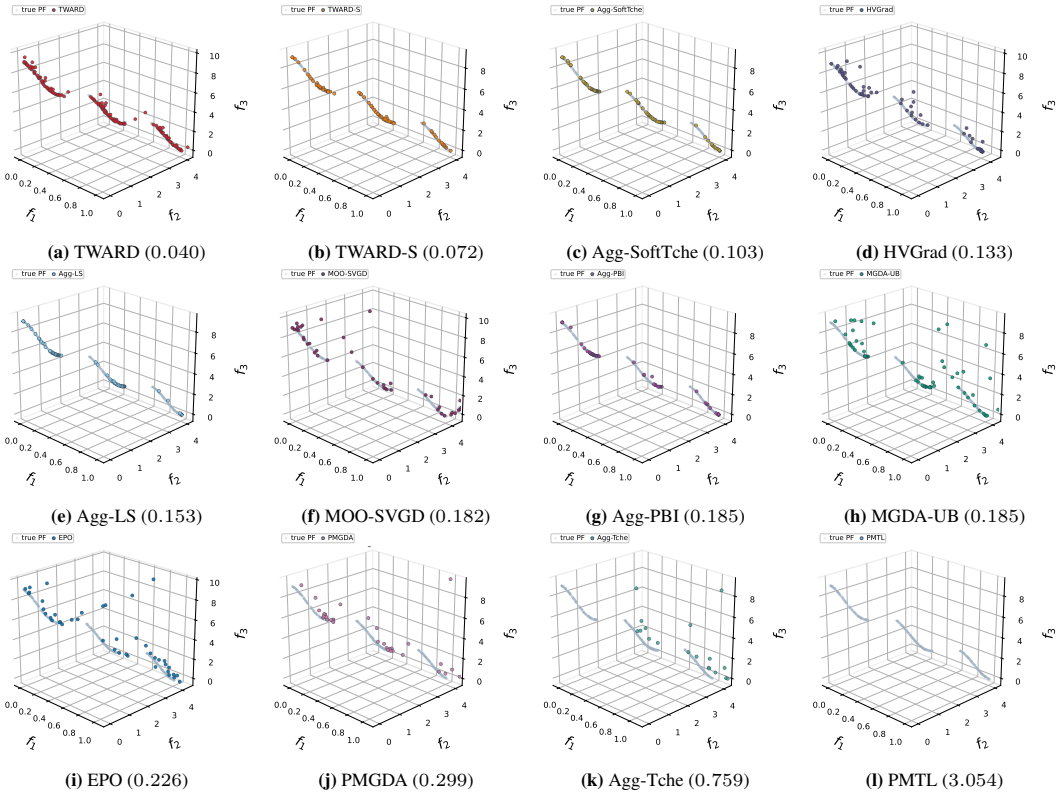


(a) TWARD (0.067) **(b)** HVGrad (0.102) **(c)** Agg-LS (0.107) **(d)** TWARD-S (0.108)

(e) Agg-PBI (0.130) **(f)** Agg-SoftTche (0.172) **(g)** EPO (0.253) **(h)** MOO-SVGD (0.264)

(i) MGDA-UB (0.300) **(j)** PMGDA (0.305) **(k)** Agg-Tche (0.901) **(l)** PMTL (2.910)

Figure 75: ZCAT14: every method (sorted by IGD).

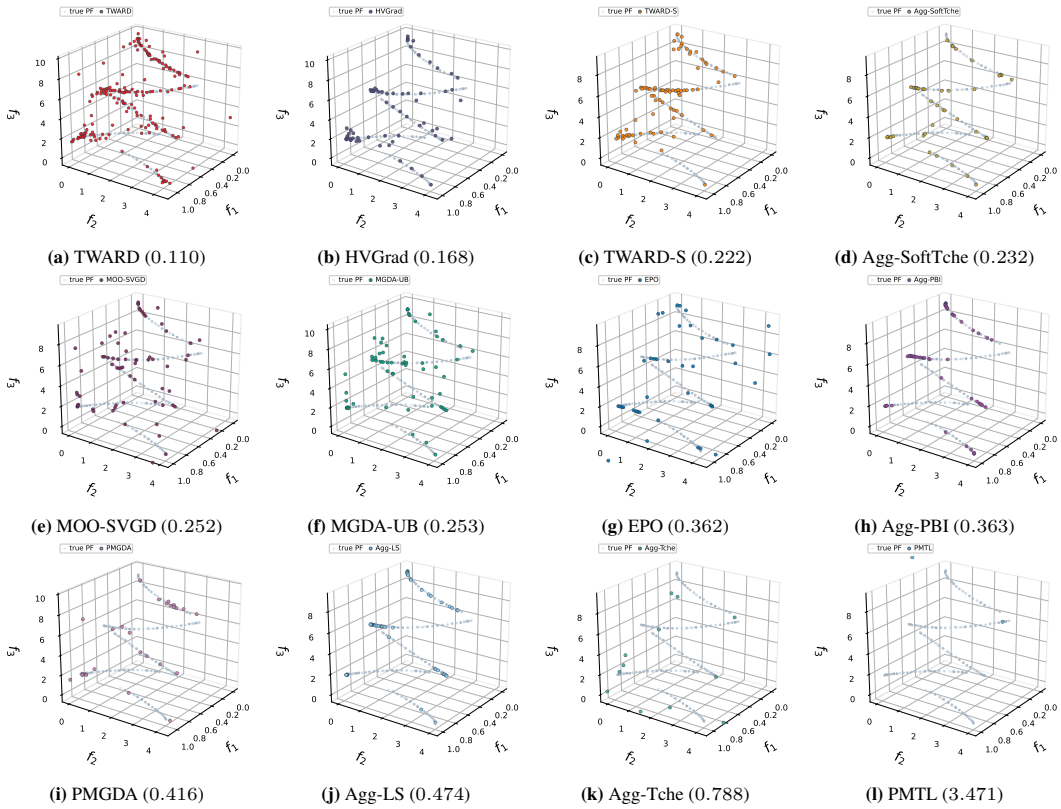


(a) TWARD (0.040) **(b)** TWARD-S (0.072) **(c)** Agg-SoftTche (0.103) **(d)** HVGrad (0.133)

(e) Agg-LS (0.153) **(f)** MOO-SVGD (0.182) **(g)** Agg-PBI (0.185) **(h)** MGDA-UB (0.185)

(i) EPO (0.226) **(j)** PMGDA (0.299) **(k)** Agg-Tche (0.759) **(l)** PMTL (3.054)

Figure 76: ZCAT15: every method (sorted by IGD).

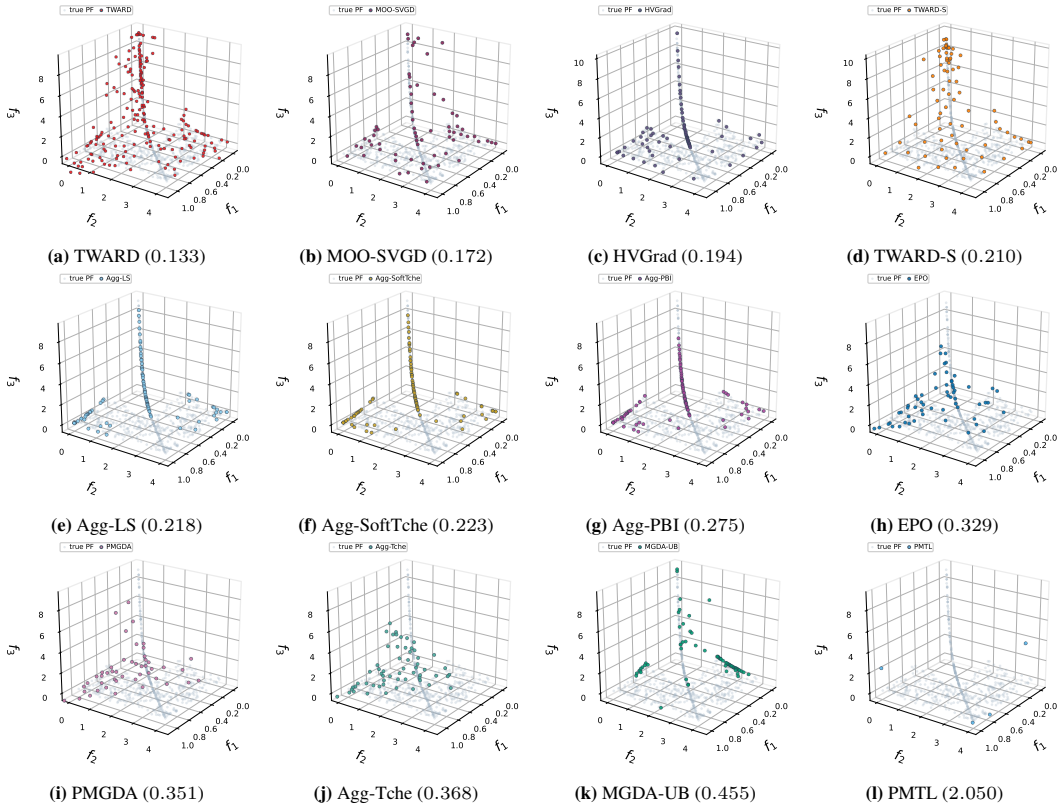


(a) TWARD (0.110) **(b)** HVGrad (0.168) **(c)** TWARD-S (0.222) **(d)** Agg-SoftTche (0.232)

(e) MOO-SVGD (0.252) **(f)** MGDA-UB (0.253) **(g)** EPO (0.362) **(h)** Agg-PBI (0.363)

(i) PMGDA (0.416) **(j)** Agg-LS (0.474) **(k)** Agg-Tche (0.788) **(l)** PMTL (3.471)

Figure 77: ZCAT16: every method (sorted by IGD).

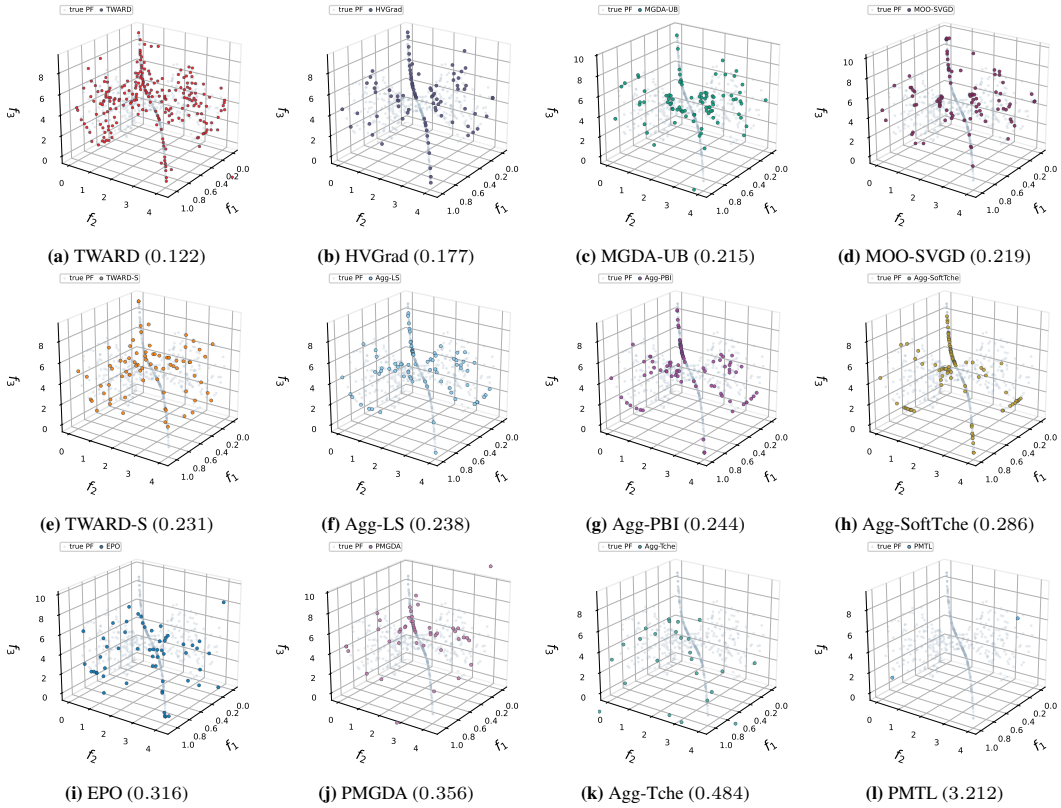


(a) TWARD (0.133) **(b)** MOO-SVGD (0.172) **(c)** HVGrad (0.194) **(d)** TWARD-S (0.210)

(e) Agg-LS (0.218) **(f)** Agg-SoftTche (0.223) **(g)** Agg-PBI (0.275) **(h)** EPO (0.329)

(i) PMGDA (0.351) **(j)** Agg-Tche (0.368) **(k)** MGDA-UB (0.455) **(l)** PMTL (2.050)

Figure 78: ZCAT17: every method (sorted by IGD).



(a) TWARD (0.122) **(b)** HVGrad (0.177) **(c)** MGDA-UB (0.215) **(d)** MOO-SVGD (0.219)

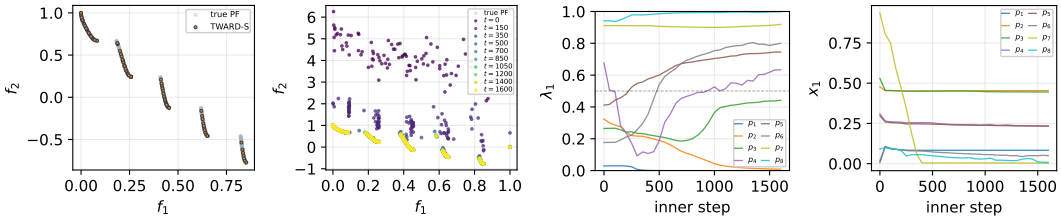
(e) TWARD-S (0.231) **(f)** Agg-LS (0.238) **(g)** Agg-PBI (0.244) **(h)** Agg-SoftTche (0.286)

(i) EPO (0.316) **(j)** PMGDA (0.356) **(k)** Agg-Tche (0.484) **(l)** PMTL (3.212)

Figure 79: ZCAT18: every method (sorted by IGD).

D.8 TWARD-S TRAJECTORIES ON ZDT3

With repulsion on and no outer evolution, TWARD-S already covers all five ZDT3 segments. Figure 82 shows the final front, F-space descent snapshots, and the separated λ_1 / x_1 time scales.

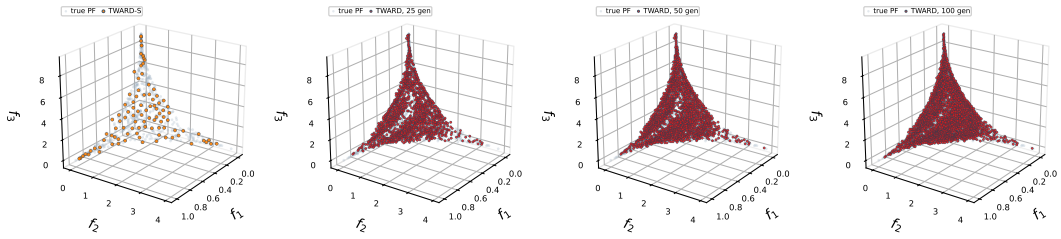


(a) Final F (0.009) **(b)** Descent **(c)** λ_1 **(d)** x_1

Figure 82: ZDT3 with TWARD-S only (no outer evolution).

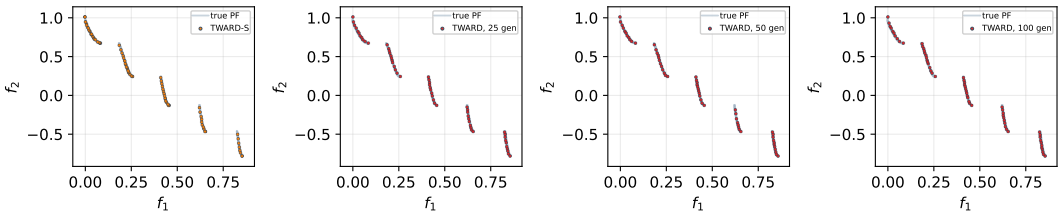
D.9 More generations beyond the matched budget

The main text shows DTLZ7 and WFG7 with TWARD-S then TWARD at $G \in \{25, 50, 100\}$ (Figures 4b–4a); the tables remain at the matched $G{=}4$ budget. Figures 83–84 repeat the same protocol on ZCAT2 and ZDT3 ($T_{\text{in}} = 400$). ZCAT2 keeps improving with G; on ZDT3 the solver is already near the floor, so further generations barely move IGD.



(a) TWARD-S (0.142) **(b)** 25 gen (0.047) **(c)** 50 gen (0.040) **(d)** 100 gen (0.035)

Figure 83: ZCAT2: TWARD-S then TWARD at 25, 50, 100 generations (IGD in parentheses).



(a) TWARD-S (0.008) **(b)** 25 gen (0.008) **(c)** 50 gen (0.009) **(d)** 100 gen (0.008)

Figure 84: ZDT3: TWARD-S then TWARD at 25, 50, 100 generations (IGD in parentheses).

D.10 More steps at fixed cardinality

Table 7 holds $N{=}100$ for every method and grows only the steps per solution, $T \in \{100, 400, 1600\}$, so cost rises $16\times$ while cardinality stays fixed. On ZDT3 several baselines improve with steps, and HVGrad passes TWARD-S. On DTLZ7 every baseline is flat or worse except MOO-SVGD, which improves from a level far above the rest.

Table 7: IGD under a step ladder at $N{=}100$, seed 0. Baselines use step size 0.01. †TWARD-S in an earlier configuration. The table checks the trend in T, not the tuned ranking of the main tables.

	DTLZ7			ZDT3		
Method	$T{=}100$	400	1600	100	400	1600
TWARD-S	0.143	0.210	0.126	0.021	0.027	0.028
HVGrad	0.433	0.476	0.611	0.028	0.005	0.017
MOO-SVGD	1.334	0.963	0.869	0.364	0.121	0.050
Agg-LS	0.247	0.207	0.201	0.058	0.062	0.062
Agg-Tche	0.328	0.284	0.353	0.255	0.178	0.170
Agg-SoftTche	0.216	0.229	0.188	0.124	0.081	0.075
Agg-PBI	0.438	0.444	0.444	0.063	0.072	0.072
EPO	0.342	0.375	0.353	0.206	0.099	0.121
PMGDA	0.369	0.375	0.359	0.255	0.267	0.275
PMTL	0.283	0.257	0.288	0.198	0.201	0.195
MGDA-UB	1.527	1.539	1.539	0.283	1.051	1.051

D.11 FULL RESULT TABLES

Tables 1–2 are in the main text. Tables 8–10 repeat the comparison over twelve methods. Every cell is mean±std over seeds 1–5 at N=100, T=1600.

Problem	TWARD-S		TWARD		HVGrad		MOO-SVGD		soft-Tche		EPO	
	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV
ZDT3	0.010 ±0.000	1.094 ±0.000	0.010 ±0.001	1.093 ±0.000	**0.005** ±0.000	**1.096** ±0.000	0.088 ±0.013	1.031 ±0.033	0.033 ±0.003	1.084 ±0.002	0.046 ±0.005	1.060 ±0.010
DTLZ2	0.055 ±0.001	0.751 ±0.001	**0.029** ±0.001	**0.779** ±0.000	0.089 ±0.004	0.711 ±0.003	0.065 ±0.002	0.735 ±0.002	0.060 ±0.001	0.715 ±0.002	0.055 ±0.000	0.682 ±0.001
DTLZ5	**0.007** ±0.000	0.133 ±0.000	0.007 ±0.001	**0.133** ±0.000	0.233 ±0.174	0.091 ±0.031	0.016 ±0.000	0.128 ±0.000	0.009 ±0.001	0.132 ±0.000	0.010 ±0.001	0.124 ±0.001
DTLZ6	**0.008** ±0.001	**0.133** ±0.000	0.022 ±0.011	0.125 ±0.006	3.234 ±0.003	0.000 ±0.000	3.228 ±0.001	0.000 ±0.000	3.165 ±0.000	0.000 ±0.000	3.174 ±0.003	0.000 ±0.000
DTLZ7	0.092 ±0.004	1.402 ±0.004	**0.090** ±0.014	**1.416** ±0.009	0.296 ±0.034	1.010 ±0.079	0.281 ±0.028	0.995 ±0.017	0.235 ±0.019	1.248 ±0.024	0.360 ±0.035	0.871 ±0.027

Problem	PBI		PMGDA		MGDA-UB		Tche		LS		PMTL	
	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV
ZDT3	0.061 ±0.010	1.060 ±0.011	0.085 ±0.023	1.007 ±0.027	0.153 ±0.009	0.834 ±0.036	0.036 ±0.004	1.074 ±0.002	0.052 ±0.005	1.076 ±0.002	0.282 ±0.036	0.819 ±0.033
DTLZ2	0.054 ±0.000	0.690 ±0.008	0.054 ±0.000	0.683 ±0.004	0.345 ±0.032	0.402 ±0.027	0.062 ±0.000	0.697 ±0.002	0.248 ±0.022	0.438 ±0.020	0.073 ±0.006	0.704 ±0.021
DTLZ5	0.013 ±0.000	0.131 ±0.000	0.018 ±0.002	0.128 ±0.001	0.135 ±0.040	0.095 ±0.008	0.009 ±0.000	0.125 ±0.000	0.137 ±0.010	0.081 ±0.001	0.039 ±0.005	0.119 ±0.002
DTLZ6	3.166 ±0.001	0.000 ±0.000	3.288 ±0.000	0.000 ±0.000	3.380 ±0.000	0.000 ±0.000	3.165 ±0.000	0.000 ±0.000	3.380 ±0.000	0.000 ±0.000	3.177 ±0.005	0.000 ±0.000
DTLZ7	0.223 ±0.014	1.266 ±0.016	0.227 ±0.014	1.171 ±0.024	1.539 ±0.000	0.303 ±0.000	0.350 ±0.036	0.913 ±0.058	0.239 ±0.016	1.253 ±0.022	0.251 ±0.027	0.978 ±0.075

Table 8: ZDT3/DTLZ, all methods (two blocks). Shading as in Table 1.

Problem	TWARD-S		TWARD		HVGrad		MOO-SVGD		soft-Tche		EPO	
	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV
WFG1	**1.514** ±0.233	**17.265** ±4.082	1.525 ±0.163	17.083 ±2.960	1.704 ±0.003	10.167 ±0.080	2.086 ±0.414	5.357 ±4.374	1.709 ±0.004	9.969 ±0.087	1.693 ±0.002	10.625 ±0.064
WFG2	0.467 ±0.082	54.982 ±3.365	**0.287** ±0.034	59.011 ±0.265	0.304 ±0.009	**59.494** ±0.057	0.329 ±0.018	58.539 ±0.208	0.373 ±0.044	58.174 ±0.390	0.486 ±0.072	55.860 ±0.472
WFG3	**0.022** ±0.001	**6.646** ±0.004	0.035 ±0.006	6.571 ±0.020	0.043 ±0.003	6.534 ±0.014	0.092 ±0.026	6.237 ±0.163	0.052 ±0.001	6.202 ±0.002	0.066 ±0.005	5.985 ±0.056
WFG4	0.640 ±0.092	21.245 ±0.536	0.482 ±0.026	24.529 ±0.417	0.492 ±0.021	23.648 ±0.703	**0.459** ±0.022	**25.418** ±0.733	0.580 ±0.026	22.305 ±0.819	0.597 ±0.034	21.460 ±0.768
WFG5	0.236 ±0.012	30.240 ±0.200	**0.215** ±0.012	**30.578** ±0.466	0.310 ±0.017	29.017 ±0.023	0.418 ±0.056	26.685 ±0.746	0.245 ±0.005	29.012 ±0.411	0.240 ±0.006	28.552 ±0.350
WFG6	0.220 ±0.004	35.744 ±0.063	**0.168** ±0.001	34.907 ±0.079	0.293 ±0.009	**36.094** ±0.057	0.202 ±0.003	35.764 ±0.020	0.265 ±0.002	32.949 ±0.277	0.247 ±0.004	32.207 ±0.131
WFG7	0.232 ±0.003	35.908 ±0.054	**0.160** ±0.006	**37.000** ±0.057	0.317 ±0.008	36.064 ±0.050	0.211 ±0.002	35.717 ±0.080	0.260 ±0.004	32.726 ±0.305	0.249 ±0.007	32.187 ±0.178
WFG8	0.438 ±0.012	27.863 ±0.450	**0.313** ±0.008	**29.959** ±0.156	0.425 ±0.024	29.066 ±0.526	0.439 ±0.039	26.273 ±0.784	0.462 ±0.013	24.472 ±0.487	0.426 ±0.017	25.321 ±0.303

Problem	PBI		PMGDA		MGDA-UB		Tche		LS		PMTL	
	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV
WFG1	1.687 ±0.003	10.607 ±0.119	1.751 ±0.028	8.674 ±1.204	1.790 ±0.027	8.818 ±0.385	1.728 ±0.005	9.720 ±0.130	1.708 ±0.006	10.008 ±0.167	1.742 ±0.002	9.419 ±0.050
WFG2	0.400 ±0.023	56.655 ±0.252	0.384 ±0.036	54.699 ±0.511	0.724 ±0.081	51.285 ±0.867	0.355 ±0.017	57.931 ±0.140	0.360 ±0.024	58.218 ±0.122	0.581 ±0.010	53.038 ±0.681
WFG3	0.148 ±0.003	5.175 ±0.040	0.157 ±0.012	5.366 ±0.137	0.534 ±0.040	3.285 ±0.237	0.071 ±0.002	5.990 ±0.015	0.559 ±0.033	2.607 ±0.341	0.602 ±0.027	2.476 ±0.166
WFG4	0.591 ±0.051	22.164 ±1.483	0.493 ±0.028	23.891 ±0.566	0.660 ±0.064	20.526 ±1.287	0.579 ±0.013	21.131 ±0.956	0.613 ±0.036	21.721 ±0.725	0.563 ±0.019	22.624 ±0.847
WFG5	0.234 ±0.004	29.045 ±0.243	0.255 ±0.004	28.434 ±0.361	0.538 ±0.110	26.265 ±1.550	0.257 ±0.007	28.680 ±0.309	0.616 ±0.069	24.110 ±0.421	0.359 ±0.010	27.457 ±0.345
WFG6	0.249 ±0.002	30.432 ±0.082	0.278 ±0.008	29.028 ±0.081	0.455 ±0.047	28.241 ±0.913	0.274 ±0.001	32.130 ±0.075	0.817 ±0.043	13.392 ±0.784	0.245 ±0.003	34.209 ±0.121
WFG7	0.282 ±0.017	31.396 ±0.251	0.233 ±0.005	32.584 ±0.421	0.541 ±0.089	29.041 ±1.346	0.268 ±0.006	32.437 ±0.149	0.787 ±0.024	15.477 ±0.898	0.763 ±0.057	18.535 ±0.660
WFG8	0.449 ±0.018	23.835 ±0.977	0.471 ±0.022	23.211 ±0.421	0.640 ±0.056	22.873 ±0.655	0.436 ±0.017	25.281 ±0.504	0.841 ±0.078	16.682 ±1.717	0.805 ±0.041	14.045 ±0.491

Table 9: WFG, all methods (two blocks). Shading as in Table 1.

Problem	TWARD-S		TWARD		HVGrad		MOO-SVGD		soft-Tche		EPO	
	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV
ZCAT1	0.243 ±0.013	36.003 ±0.212	**0.178** ±0.010	**36.991** ±0.099	0.239 ±0.012	36.767 ±0.122	0.581 ±0.052	31.968 ±0.673	0.280 ±0.010	35.600 ±0.176	0.687 ±0.071	24.539 ±0.875
ZCAT2	0.180 ±0.006	45.653 ±0.064	**0.086** ±0.003	**45.863** ±0.039	0.162 ±0.012	45.801 ±0.018	0.234 ±0.038	45.271 ±0.079	0.397 ±0.049	45.075 ±0.089	1.058 ±0.224	41.394 ±0.391
ZCAT3	0.204 ±0.004	23.657 ±0.025	**0.130** ±0.002	**24.183** ±0.029	0.200 ±0.003	23.722 ±0.079	0.284 ±0.025	22.196 ±0.266	0.349 ±0.024	18.846 ±0.535	0.448 ±0.002	18.527 ±0.154
ZCAT4	0.252 ±0.008	24.469 ±0.256	**0.149** ±0.004	**25.904** ±0.062	0.213 ±0.010	25.444 ±0.085	0.348 ±0.028	23.665 ±0.340	0.319 ±0.017	23.647 ±0.322	0.431 ±0.012	20.336 ±0.440
ZCAT5	0.169 ±0.003	42.891 ±0.045	**0.110** ±0.005	43.173 ±0.035	0.154 ±0.007	**43.208** ±0.045	0.196 ±0.017	43.063 ±0.081	0.257 ±0.020	42.146 ±0.230	0.495 ±0.050	37.774 ±0.538
ZCAT6	0.235 ±0.014	26.759 ±0.091	**0.160** ±0.011	**27.747** ±0.049	0.247 ±0.028	27.166 ±0.084	0.312 ±0.026	23.025 ±1.112	0.356 ±0.028	25.318 ±0.186	0.368 ±0.003	23.681 ±0.485
ZCAT7	0.231 ±0.008	25.044 ±0.260	**0.148** ±0.005	25.898 ±0.068	0.378 ±0.017	**25.911** ±0.039	0.322 ±0.010	24.655 ±0.279	0.386 ±0.029	22.413 ±0.101	0.372 ±0.017	19.727 ±0.304
ZCAT8	0.235 ±0.017	2.980 ±0.090	**0.143** ±0.013	**3.298** ±0.060	0.410 ±0.032	2.441 ±0.035	0.240 ±0.030	2.996 ±0.074	0.383 ±0.023	2.460 ±0.024	0.383 ±0.016	2.196 ±0.123
ZCAT9	0.284 ±0.015	13.144 ±0.150	**0.209** ±0.008	13.187 ±0.081	0.280 ±0.008	**13.260** ±0.070	0.429 ±0.062	11.335 ±0.459	0.416 ±0.023	12.353 ±0.191	0.501 ±0.041	10.364 ±0.596
ZCAT10	0.119 ±0.003	**6.041** ±0.042	**0.093** ±0.002	6.007 ±0.032	0.143 ±0.005	5.994 ±0.033	0.229 ±0.020	5.440 ±0.191	0.323 ±0.029	5.520 ±0.076	0.237 ±0.006	5.291 ±0.249
ZCAT11	0.264 ±0.014	23.436 ±0.150	**0.186** ±0.006	**24.055** ±0.106	0.236 ±0.016	24.051 ±0.313	0.329 ±0.035	22.767 ±0.573	0.431 ±0.023	21.898 ±0.384	0.532 ±0.030	19.277 ±0.988
ZCAT12	0.236 ±0.005	11.125 ±0.091	**0.140** ±0.006	**11.724** ±0.044	0.213 ±0.006	11.253 ±0.111	0.272 ±0.006	10.414 ±0.151	0.388 ±0.023	9.727 ±0.132	0.451 ±0.038	7.779 ±0.389
ZCAT13	0.353 ±0.053	29.336 ±0.797	**0.157** ±0.005	31.366 ±0.072	0.230 ±0.010	**31.427** ±0.042	0.333 ±0.035	29.625 ±0.479	0.511 ±0.081	26.466 ±0.653	0.834 ±0.036	25.544 ±0.734
ZCAT14	0.162 ±0.021	17.406 ±0.078	0.103 ±0.004	17.332 ±0.070	**0.102** ±0.015	**17.636** ±0.050	0.288 ±0.037	16.432 ±0.268	0.126 ±0.009	17.422 ±0.092	0.258 ±0.022	16.158 ±0.196
ZCAT15	0.066 ±0.007	**21.521** ±0.102	**0.062** ±0.006	21.449 ±0.047	0.111 ±0.020	21.348 ±0.286	0.237 ±0.048	20.382 ±0.247	0.143 ±0.022	21.116 ±0.110	0.263 ±0.037	19.449 ±0.397
ZCAT16	0.201 ±0.032	20.305 ±0.109	**0.130** ±0.015	**20.583** ±0.022	0.238 ±0.080	20.144 ±0.410	0.317 ±0.055	19.935 ±0.144	0.316 ±0.034	19.466 ±0.283	0.413 ±0.026	18.135 ±0.190
ZCAT17	0.242 ±0.023	42.506 ±0.081	0.229 ±0.142	**43.215** ±0.339	**0.203** ±0.019	42.806 ±0.201	0.213 ±0.021	42.410 ±0.420	0.259 ±0.019	42.805 ±0.125	0.382 ±0.036	40.867 ±0.458
ZCAT18	0.268 ±0.032	23.938 ±0.780	**0.126** ±0.005	26.223 ±0.045	0.172 ±0.009	**26.238** ±0.094	0.239 ±0.015	25.568 ±0.215	0.288 ±0.026	24.884 ±0.184	0.345 ±0.045	21.712 ±1.025
ZCAT19	0.197 ±0.015	23.054 ±0.132	**0.127** ±0.006	**24.004** ±0.097	0.180 ±0.027	23.667 ±0.208	0.353 ±0.039	21.824 ±0.195	0.204 ±0.018	23.090 ±0.165	0.316 ±0.035	19.618 ±0.630
ZCAT20	0.326 ±0.132	23.440 ±0.128	**0.161** ±0.008	24.400 ±0.181	0.191 ±0.010	**24.495** ±0.161	0.227 ±0.036	24.473 ±0.299	0.456 ±0.086	23.873 ±0.177	0.403 ±0.024	18.741 ±0.821

Problem	PBI		PMGDA		MGDA-UB		Tche		LS		PMTL	
	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV
ZCAT1	1.000 ±0.033	25.135 ±0.470	0.487 ±0.083	29.539 ±0.273	0.679 ±0.091	30.423 ±0.981	0.646 ±0.045	29.182 ±0.636	0.485 ±0.035	33.134 ±0.486	2.410 ±0.065	0.000 ±0.000
ZCAT2	0.383 ±0.059	45.097 ±0.114	0.606 ±0.067	40.877 ±0.651	0.290 ±0.046	44.538 ±0.344	1.700 ±0.099	41.725 ±0.198	0.727 ±0.274	44.505 ±0.195	3.139 ±0.154	0.000 ±0.000
ZCAT3	0.637 ±0.054	19.003 ±0.516	0.453 ±0.021	17.187 ±0.153	0.351 ±0.020	21.371 ±0.483	0.619 ±0.018	17.793 ±0.384	0.511 ±0.049	17.626 ±0.596	2.443 ±0.038	0.000 ±0.000
ZCAT4	0.535 ±0.042	21.549 ±0.285	0.440 ±0.032	19.148 ±0.670	0.373 ±0.031	23.475 ±0.309	0.740 ±0.076	10.786 ±1.194	0.552 ±0.057	21.565 ±0.602	3.532 ±0.239	0.000 ±0.000
ZCAT5	0.243 ±0.018	41.625 ±0.525	0.414 ±0.041	38.096 ±0.503	0.262 ±0.026	41.129 ±0.922	0.509 ±0.038	37.974 ±0.831	0.216 ±0.022	41.928 ±0.530	2.582 ±0.145	0.000 ±0.000
ZCAT6	0.727 ±0.172	16.497 ±2.456	0.383 ±0.016	24.022 ±0.240	0.459 ±0.084	19.620 ±2.659	0.489 ±0.062	20.228 ±0.543	0.811 ±0.160	13.545 ±2.531	2.776 ±0.098	0.000 ±0.000
ZCAT7	0.482 ±0.034	23.685 ±0.283	0.707 ±0.042	20.127 ±0.490	0.332 ±0.037	24.169 ±0.362	0.505 ±0.026	17.782 ±0.293	0.509 ±0.058	24.449 ±0.086	2.557 ±0.150	0.000 ±0.000
ZCAT8	0.432 ±0.010	2.014 ±0.044	0.376 ±0.042	2.235 ±0.082	0.291 ±0.023	2.365 ±0.111	0.625 ±0.031	1.165 ±0.117	0.633 ±0.129	1.820 ±0.193	2.595 ±0.113	0.000 ±0.000
ZCAT9	0.604 ±0.049	9.142 ±0.516	0.626 ±0.067	9.339 ±0.517	0.465 ±0.060	11.022 ±0.317	0.781 ±0.078	7.227 ±0.221	0.913 ±0.205	11.052 ±0.539	3.173 ±0.369	0.000 ±0.000
ZCAT10	0.274 ±0.046	5.562 ±0.042	0.288 ±0.036	5.447 ±0.105	0.237 ±0.021	5.318 ±0.218	0.666 ±0.100	3.260 ±0.323	0.277 ±0.034	5.563 ±0.053	2.622 ±0.142	0.000 ±0.000
ZCAT11	0.480 ±0.028	22.037 ±0.346	0.445 ±0.037	19.944 ±0.603	0.456 ±0.008	20.478 ±0.157	0.734 ±0.072	15.392 ±0.546	0.449 ±0.029	22.208 ±0.281	2.551 ±0.101	0.002 ±0.005
ZCAT12	0.503 ±0.036	9.588 ±0.107	0.388 ±0.027	9.113 ±0.105	0.295 ±0.024	10.136 ±0.301	0.640 ±0.066	6.080 ±0.629	0.453 ±0.040	9.699 ±0.125	2.507 ±0.104	0.000 ±0.000
ZCAT13	0.710 ±0.019	28.961 ±0.159	0.398 ±0.021	27.966 ±0.351	0.490 ±0.045	27.157 ±0.814	0.853 ±0.082	21.605 ±0.636	0.725 ±0.026	29.387 ±0.127	2.588 ±0.185	0.000 ±0.000
ZCAT14	0.135 ±0.017	17.237 ±0.138	0.357 ±0.031	14.989 ±0.286	0.261 ±0.030	16.590 ±0.181	0.797 ±0.103	11.210 ±0.643	0.132 ±0.033	17.320 ±0.125	3.079 ±0.190	0.000 ±0.000
ZCAT15	0.252 ±0.054	20.527 ±0.200	0.279 ±0.029	19.010 ±0.228	0.185 ±0.032	21.044 ±0.136	0.711 ±0.091	11.877 ±0.989	0.194 ±0.032	20.993 ±0.080	3.231 ±0.248	0.000 ±0.000
ZCAT16	0.400 ±0.071	19.707 ±0.186	0.506 ±0.059	16.971 ±0.715	0.329 ±0.056	19.884 ±0.132	1.000 ±0.084	11.151 ±1.888	0.488 ±0.044	19.498 ±0.107	3.236 ±0.285	0.000 ±0.000
ZCAT17	0.315 ±0.037	42.434 ±0.142	0.395 ±0.047	38.913 ±1.242	0.461 ±0.030	42.300 ±0.203	0.508 ±0.066	36.575 ±0.633	0.247 ±0.012	42.879 ±0.188	2.045 ±0.187	0.010 ±0.021
ZCAT18	0.270 ±0.018	24.873 ±0.311	0.345 ±0.031	22.586 ±0.338	0.227 ±0.016	25.022 ±0.266	0.553 ±0.060	17.080 ±0.267	0.273 ±0.019	24.690 ±0.420	3.075 ±0.105	0.000 ±0.000
ZCAT19	0.242 ±0.023	22.766 ±0.312	0.341 ±0.048	20.994 ±0.223	0.312 ±0.034	22.094 ±0.361	0.550 ±0.053	14.623 ±0.477	0.205 ±0.014	23.031 ±0.170	2.430 ±0.344	0.217 ±0.434
ZCAT20	0.637 ±0.156	22.330 ±0.489	0.640 ±0.183	19.662 ±0.536	0.238 ±0.018	23.579 ±0.188	0.424 ±0.047	17.448 ±0.621	0.655 ±0.112	23.199 ±0.333	2.364 ±0.570	0.316 ±0.631

Table 10: ZCAT, all methods (two blocks). Shading as in Table 2.

D.12 UCI ADULT ACCURACY–FAIRNESS FRONTS

Figure 85 shows the test fronts behind Table 3 (Section 5.5). Axes are demographic-parity gap, cross-entropy and equalized-odds gap (all minimised). TWARD-S / TWARD use the κ-restore default of baselines use 48 preference rays and 200 Adam steps.

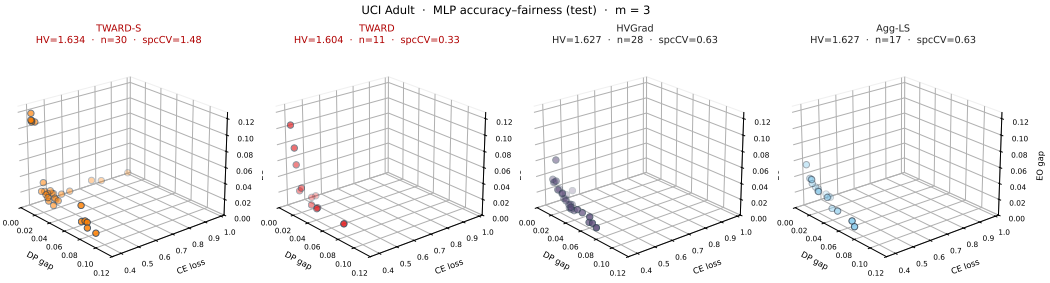


Figure 85: UCI Adult, $m=3$: test accuracy–fairness fronts