

AI-Driven Representation of Pareto Fronts for Continuous Multi-Objective Optimization Problems

Janusz Granat

Institute of Control and Computation Engineering, Warsaw University of
Technology, Warsaw, Poland
Nowowiejska 15/19, Warsaw, Poland

Multi-objective problems with continuous decision variables arise in engineering design, resource allocation, and other domains. Their solution is not a single optimum but so-called Pareto solutions: the set of trade-off points where no objective can be improved without worsening another. This remains a long-standing computational challenge, especially when objective evaluations are expensive. This paper surveys how artificial intelligence, including neural hypernetworks, Gaussian-process surrogates, and diffusion models, has recently been used to build Pareto front representations (PFRs). We propose a novel method that combines three ideas—implicit neural representations, uncertainty-guided active learning over an expensive black-box objective, and diffusion-based mode discovery—into a single training pipeline. Its end deliverable is a compact network that returns an estimated Pareto-optimal solution and its objective values in one forward pass for any continuous trade-off preference, while budgeting expensive true evaluations and aiming to repair blind spots on disconnected fronts. In a small-scale study, the training objective represents a non-convex front on which a weighted-sum baseline collapses to a single point, the aspiration interface returns points on the front, and the full closed loop reaches the front under a 220-evaluation budget; the acquisition and repair steps are not yet shown to improve on random selection, and a disconnected front can still lose a segment.

1 Introduction

A continuous multi-objective optimization problem asks for the set of decision vectors for which no objective can be improved without worsening another – the Pareto set, whose image in objective space is the Pareto front. When the decision space is continuous, this front is in general an uncountable, possibly disconnected and non-convex manifold, and no single scalar number describes it; a decision maker needs either the whole manifold or a fast way to query it at a chosen trade-off.

Several methods exist for finding Pareto-optimal solutions. We can distinguish two main approaches:

- *Interactive methods*, in which a decision maker iteratively specifies preferences (e.g. aspiration and reservation levels, weights, or a reference point) and the algorithm returns a single Pareto-optimal solution matching that specification; the decision maker then revises the preferences based on

the returned solution, and the process repeats until a satisfactory compromise is found. This tradition goes back to Wierzbicki’s (Wierzbicki, 1980, 1982) achievement scalarizing function (ASF) and is embodied in interactive systems such as ISAAP (Granat & Makowski, 2000).

- *Generation of a Pareto front representation (PFR) by a set of points*, in which the algorithm computes, up front and without further decision-maker involvement, a finite or continuous set of points approximating the whole efficient set, which the decision maker can then browse freely. This includes classical population-based multi-objective evolutionary algorithms (e.g. NSGA-II (Deb K. et. al (2002)), MOEA/D (Zhang Q. & Li H. (2007))), their surrogate-assisted variants, learned hypernetwork-based representations (Pareto set learning), and more recent diffusion-guided generators.

Method proposed in this paper, called PFR_INR, belongs to the second family – it produces a Pareto front representation – but because that representation is a queryable function rather than a fixed, discrete point cloud, it also supports fast, repeated interactive querying, narrowing the practical gap between the two approaches.

Discrete point cloud (classical MOEA): an evolutionary algorithm terminates with a finite list of concrete Pareto optimal solutions. That is all information that is available. If the decision maker states a preference that does not coincide exactly with one of these solutions, there is no ready answer. In such a case, the nearest already computed point must be picked (an approximation), or the entire evolutionary algorithm should be re-run for the new preferences. The set is “frozen” at the moment the computation ends.

Queryable function – PFR_INR: instead of a list of points, you have a trained neural network – a function which, for any preferences (not only the ones seen during training), immediately returns the corresponding solution with no re-running of the optimization.

Figure 1 shows two representations of the same Pareto front.

2 Mathematical problem formulation

The nonlinear multi-objective optimization problem is formulated as follows:

$$\max_{\mathbf{x} \in X} \mathbf{q}(\mathbf{x}) = (q_1(\mathbf{x}), q_2(\mathbf{x}), \dots, q_k(\mathbf{x}))^\top \quad (\text{MOP})$$

where X is the *feasible set* in decision space:

$$X = \{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \leq 0 \ \forall j, \ h_l(\mathbf{x}) = 0 \ \forall l, \ \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U\},$$

and the Q is a *feasible objective set*:

$$Q = \{\mathbf{q}(\mathbf{x}) \mid \mathbf{x} \in X\} \subset \mathbb{R}^k.$$

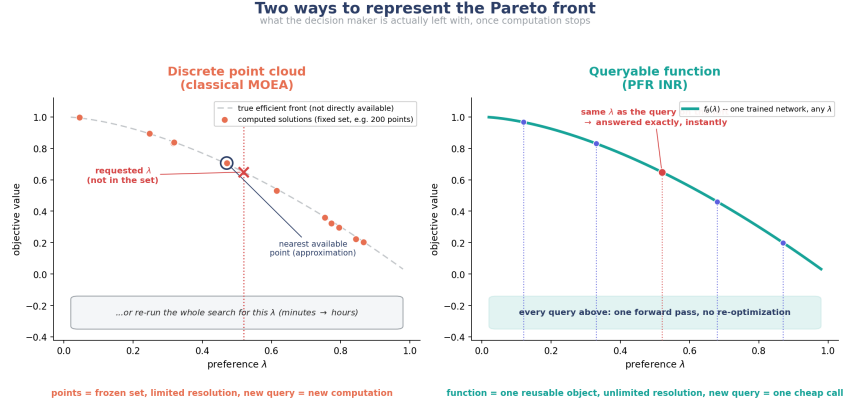


Figure 1: Two representations of the same Pareto front. *Left*: a classical evolutionary approach *Right*: PFR.INR represents the front as a trained function

Definition 1 (Pareto dominance). A point $\mathbf{x} \in X$ dominates $\mathbf{x}' \in X$, written $\mathbf{x} \succ \mathbf{x}'$, if

$$q_i(\mathbf{x}) \geq q_i(\mathbf{x}') \quad \forall i \in \{1, \dots, k\} \quad \text{and} \quad \exists r \in \{1, \dots, k\} : q_r(\mathbf{x}) > q_r(\mathbf{x}').$$

Equivalently, $\mathbf{q}(\mathbf{x}) \geq \mathbf{q}(\mathbf{x}')$.

Definition 2 (Pareto optimality). A point $\mathbf{x}^* \in X$ is

1. Pareto optimal (efficient) if there is no $\mathbf{x} \in X$ with $\mathbf{q}(\mathbf{x}) \geq \mathbf{q}(\mathbf{x}^*)$, i.e. $(\mathbf{q}(\mathbf{x}^*) + \mathbb{R}_+^k) \cap Q = \{\mathbf{q}(\mathbf{x}^*)\}$;
2. weakly Pareto optimal if there is no $\mathbf{x} \in X$ with $\mathbf{q}(\mathbf{x}) > \mathbf{q}(\mathbf{x}^*)$;
3. properly Pareto optimal (Geoffrion) if it is Pareto optimal and there exists $M > 0$ such that for every i and every $\mathbf{x} \in X$ with $q_i(\mathbf{x}) > q_i(\mathbf{x}^*)$ there is j with $q_j(\mathbf{x}) < q_j(\mathbf{x}^*)$ and

$$\frac{q_i(\mathbf{x}) - q_i(\mathbf{x}^*)}{q_j(\mathbf{x}^*) - q_j(\mathbf{x})} \leq M.$$

Definition 3 (Pareto set and Pareto front).

$$X_P = \{\mathbf{x}^* \in X : \nexists \mathbf{x} \in X \text{ with } \mathbf{q}(\mathbf{x}) \geq \mathbf{q}(\mathbf{x}^*)\}, \quad Q_P = \mathbf{q}(X_P) = \{\mathbf{q}(\mathbf{x}) : \mathbf{x} \in X_P\}.$$

X_P is the Pareto (efficient) set; Q_P is the Pareto front (nondominated set).

Definition 4 (Discrete representation). A discrete representation of the Pareto front is a finite set

$$R_P = \{\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(N)}\} \subseteq Q_P, \quad \mathbf{q}^{(s)} = \mathbf{q}(\mathbf{x}^{(s)}), \quad \mathbf{x}^{(s)} \in X_P,$$

with the corresponding decision-space representation $R_X = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\} \subseteq X_P$. If only $R \subseteq Q$ is required and the points of R_P are mutually nondominated, R_P is called an approximation of Q_P .

Typically R_P is generated by solving a scalarized problem on a finite parameter grid $\{\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(N)}\} \subset \Theta$:

$$\mathbf{x}^{(s)} \in \arg \max_{\mathbf{x} \in X} s(\mathbf{q}(\mathbf{x}); \boldsymbol{\theta}^{(s)}), \quad s = 1, \dots, N.$$

Definition 5 (Characteristic points of the Pareto front). *The ideal point $\mathbf{q}^I$, utopia point $\mathbf{q}^U$, and nadir point $\mathbf{q}^N$ are*

$$q_i^I = \max_{\mathbf{x} \in X} q_i(\mathbf{x}), \quad q_i^U = q_i^I + \varepsilon_i \ (\varepsilon_i > 0), \quad q_i^N = \min_{\mathbf{x} \in X_P} q_i(\mathbf{x}), \quad i = 1, \dots, k.$$

Hence $Q_P \subseteq \{\mathbf{q} : \mathbf{q}^N \leq \mathbf{y} \leq \mathbf{y}^I\}$.

3 Quality measures of a representation

Let $d(\cdot, \cdot)$ be a metric on $\mathbb{R}^k$ (e.g. the ℓ_∞ or ℓ_2 distance, usually applied to objectives normalized by $\tilde{y}_i = (y_i - y_i^N)/(y_i^I - y_i^N) \in [0, 1]$).

Definition 6 (Cardinality, coverage, uniformity – Sayin S. (2000)).

$$\text{Cardinality: } |R_P| = N, \quad (1)$$

$$\text{Coverage error: } \gamma(R_P) = \sup_{\mathbf{q} \in Q_P} \min_{\mathbf{q}' \in R_P} d(\mathbf{q}, \mathbf{q}'), \quad (2)$$

$$\text{Uniformity level: } \delta(R_P) = \min_{\substack{\mathbf{q}, \mathbf{q}' \in R \\ \mathbf{q} \neq \mathbf{q}'}} d(\mathbf{q}, \mathbf{q}'). \quad (3)$$

A good representation has small $\varepsilon(R_P)$, large $\delta(R_P)$, and moderate $|R_P|$.

Hypervolume (HV). Let $A = \{a_1, \dots, a_m\} \subset \mathbb{R}^k$ be a finite approximation set in objective space (maximization assumed, k objectives), and let $r \in \mathbb{R}^k$ be a reference point dominated by every point of A (i.e. $a_i \leq r$ componentwise for all $a_i \in A$, and r strictly worse than A in every objective). The hypervolume indicator is the Lebesgue measure of the region of objective space that is simultaneously dominated by at least one point of A and dominates r :

$$\text{HV}(A, r) = \Lambda \left(\bigcup_{a \in A} [a, r] \right) = \Lambda(\{x \in \mathbb{R}^m : \exists a \in A, a \leq x \leq r\}), \quad (4)$$

where $\Lambda(\cdot)$ denotes Lebesgue measure (area for $m=2$, volume for $m=3, \dots$) and $[a, r] = \{x \in \mathbb{R}^k : a_i \leq x_i \leq r_i \ \forall i\}$ is the axis-aligned box spanned by a and r . Larger HV is better; HV rewards both proximity to the true front and spread across it, but – as noted in the experiments below – it conflates the two into a single number, so two very differently-shaped approximation sets can achieve similar HV.

Inverted Generational Distance (IGD). Let $Y_N = \{y_1, \dots, y_k\} \subset \mathbb{R}^k$ be a finite, densely sampled reference set drawn from the true Pareto front, and let A be the achieved approximation set as above. For each reference point y_i , define its gap to A as the distance to its nearest neighbor in A :

$$g_i = \min_{a \in A} \|y_i - a\|_2. \quad (5)$$

The inverted generational distance is the mean of these gaps over the reference set:

$$\text{IGD}(A, Q_P) = \frac{1}{|Q_P|} \sum_{i=1}^{|Q_P|} g_i = \frac{1}{|Q_P|} \sum_{y \in Q_P} \min_{a \in A} \|y - a\|_2. \quad (6)$$

Smaller IGD is better. IGD is the average-case counterpart of coverage error $\varepsilon(A) = \max_i g_i$: coverage error is the *worst*-covered reference point, IGD is the *typical* one. Unlike HV, IGD requires a reference set rather than a single reference point, but needs no dominance/reference-point assumption and is not biased toward sets that merely spread widely.

4 Related Work

In this paper we are not focusing on overview of various method for multicriteria analysis. We will focus only on generation of a Pareto front representation (PFR).

Classical multi-objective evolutionary algorithms (MOEAs), such as NSGA-II (Deb et al., 2002) and MOEA/D (Zhang & Li, 2007), approximate this manifold with a finite, evolving population of points. This is simple and robust but couples resolution to population size, offers no query capability between the points actually evaluated, and becomes expensive when each objective evaluation is costly (e.g. a simulation or a physical experiment). Surrogate-assisted evolutionary algorithms (Liang, J. et al., 2025 ; Jin, 2011; Jin et al., 2019) address the cost problem by guiding the search with a cheap Gaussian-process or neural surrogate, as in MOEA/D-EGO (Zhang et al., 2010), but the output is still a discrete point cloud, and the surrogate and the search remain loosely coupled.

A different line of work, Pareto set learning (PSL), trains a hypernetwork that maps a preference vector λ on the objective simplex directly to a decision vector, turning the front into a single queryable network instead of a population (Navon et al., 2021; Ruchte & Grabocka, 2021; Hoang et al., 2023; see also Ma et al., 2020, for continuous Pareto exploration). This is attractive but almost always assumes cheap, differentiable objectives; extensions that add a surrogate for expensive objectives, such as Pareto set learning with Gaussian-process surrogates and a Tchebycheff scalarization (Lin et al., 2022) and its parametric extension (Cheng et al., 2025), have not, to this paper’s knowledge, been stress-tested on genuinely disconnected fronts. A third, more recent line uses diffusion models to generate front-covering candidates (Yao et al., 2024;

Hotegni & Peitz, 2025), which can reach non-convex and disconnected regions but is expensive to query at inference time, since generation requires an iterative denoising process (Ho et al., 2020) rather than one forward pass.

A fourth, older tradition, reference-point methods with aspiration and reservation levels, solves a different part of the problem: how a decision maker specifies what they want in the first place, rather than an abstract weight vector. Wierzbicki’s achievement scalarizing function (ASF) formalizes this by mapping a stated aspiration point and reservation point to a single Pareto-optimal compromise solution, and interactive systems built on it, such as ISAAP (Granat & Makowski, 2000), let a decision maker drag a target and immediately see the resulting trade-off. This procedure, however, re-solves an optimization problem every time the reference point moves.

No method surveyed here combines a cheap, single-forward-pass queryable representation with an active-evaluation loop for expensive objectives, a repair mechanism for disconnected fronts, and a decision-maker-friendly aspiration/reservation interface. This paper proposes PFR INR to close that gap and reports a small-scale computational study of its components and of the assembled closed loop. Its contributions are: the method itself; a set of component tests on the ZDT problems with exactly known fronts, including an aspiration–reservation interface; a test of the closed loop under an evaluation budget against two ablations of itself; and a comparison of the representation with exact, certified efficient sets of four linear multi-objective programs. Section 2 states the problem, Section 3 the quality measures, Section 4 the related work, Section 5 the method, Section 6 the experiments on benchmark problems and Section 7 the linear benchmarks; Section 8 summarizes the evidence and its limits.

Table 1 summarizes the families discussed above and the family the proposed method draws from most heavily. Two gaps stand out. First, methods built for a compact queryable representation (PSL/hypernetworks) mostly assume cheap objectives, while methods built for expensive objectives (SAEAs) give up the queryable representation; none pairs the two with a mechanism that actively detects and repairs collapsed or disconnected regions of the learned front. Second, the reference-point/ASF tradition already has a decision-maker-friendly way to specify a target – an aspiration level to aim for and a reservation level that must still be acceptable – but none of the neural PSL methods surveyed condition their representation on this kind of input; they condition on abstract preference weights instead.

The same reference-point tradition has also produced software that generates and certifies *discrete* representations of a linear model’s efficient set directly: pyMCMA (Makowski et al., 2024) generates a uniformly distributed representation via repeated ASF optimization.

The closest precedents of PFR.INR learn one network that maps a preference to a Pareto-optimal solution. Navon et al. (2021) introduced Pareto HyperNetworks: a hypernetwork receives a preference vector and returns the weights of a target network, and it is trained on sampled preference rays with a linear-scalarization loss or with the Exact Pareto Optimal (EPO) solver of Mahapatra and Rajan (2020), so that a single training run yields a model for

Table 1: Families of AI-driven approaches to representing continuous Pareto fronts

Approach family	Representative work	Core idea	Main limitation
Classical MOEAs	NSGA-II (Deb et al., 2002); MOEA/D (Zhang & Li, 2007)	Finite evolving population of points	Discrete, fixed-size cloud; no query between points
Surrogate-assisted MOEAs	Jin (2011); Jin et al. (2019); Liang et al. (2025)	Population guided by GP/NN surrogates	Still a point cloud; surrogate and search loosely coupled
Pareto set learning (hypernetworks)	Navon et al. (2021); Hoang et al. (2023); Ruchte & Grabocka (2021)	Hypernetwork $\lambda \rightarrow$ decision weights	Assumes cheap/differentiable objectives
Generative / diffusion-guided	Yao et al. (2024); Hotegni & Peitz (2025)	Guided denoising toward the front	Expensive at inference; iterative sampling
Reference-point / ASF (classical MCDM)	Wierzbicki (1980, 1982); Granat & Makowski (2000)	Aspiration/reservation $\rightarrow$ achievement scalarizing function	No learned representation; re-solved per query
Proposed: PFR_INR	—	Single implicit neural manifold, one forward pass per query	Tests in this paper

every preference. The method assumes objectives that are cheap to evaluate and differentiable, and it has no notion of an evaluation budget. Lin et al. (2022) address the expensive case with a Pareto set model that maps a preference vector to a solution and is trained on Gaussian-process surrogates of the objectives with a Tchebycheff scalarization; in each round, candidates sampled from the learned set are selected for true evaluation in batches by hypervolume improvement. Cheng, Xue and Zhang (2025), referred to here as PPSL-MOBO, extend this recipe to parametric problems whose objectives depend on external parameters: a hypernetwork with low-rank adaptation maps both a preference vector and the task parameters to a solution, a Gaussian-process surrogate over the augmented space supplies lower-confidence-bound acquisition signals, and training uses the smooth Tchebycheff scalarization. PFR_INR shares with these methods the idea of a single preference-conditioned network trained on a surrogate under an evaluation budget. It differs in three respects: the network is a Fourier-feature implicit neural representation conditioned on a scalar preference rather than a hypernetwork; the learned front is periodically repaired with a diffusion model trained on the accumulated preference–decision pairs; and the same network can be conditioned on aspiration and reservation levels, which gives a decision maker a direct way to specify a preference.

5 Proposed Method: PFR_INR

5.1 Architecture and Training Objective

PFR_INR is an implicit neural representation (INR) with two heads sharing a trunk: a decision head $x^*(\lambda)$ and an objective head $q^*(\lambda)$, both conditioned on a preference vector λ on the objective simplex via Fourier positional features. It is trained with a Smooth Tchebycheff (STCH) scalarization loss (a differentiable relaxation of the weighted-Tchebycheff scalarization, which – unlike a weighted sum – can reach non-convex regions of the front), a consistency term tying the objective head’s prediction to the true objective evaluated at the decision head’s output, a Sobolev smoothness penalty that discourages the manifold from oscillating between neighbouring preferences, and a differentiable hypervolume-diversity loss that discourages the sampled front from clustering, giving the continuous representation the same diversity pressure that crowding distance gives a discrete population.

5.2 Active-Evaluation Loop and Diffusion-Based Repair

Because true objective evaluations are assumed expensive, PFR_INR spends them deliberately. At each round, a batch of preference vectors is scored by a surrogate ensemble’s epistemic uncertainty combined with the expected hypervolume improvement of evaluating it, in the style of Bayesian optimization; a small batch is then evaluated on the true objective and used to update the surrogate ensemble and briefly fine-tune the INR.

Periodically, a lightweight conditional diffusion model is trained on all preference–decision pairs (λ, x) accumulated in $\mathcal{D}$, that is, on points whose objective values were obtained from true evaluations; the surrogate ensemble is not used to generate training data. Sampled at randomly drawn preferences, the model proposes candidate decision vectors across the decision space. Candidates whose surrogate-predicted objective vector is dominated by the current non-dominated set are discarded, the survivors are ranked by predicted hypervolume gain, and the best few are evaluated on the true objective. Those that turn out to be non-dominated are added as supervised anchor points, and the INR is fine-tuned to match them. The surrogate therefore only filters and ranks the proposals, and every anchor is confirmed by a true evaluation. The diffusion model is discarded at deployment time – only the cheap INR is shipped, keeping query cost at a single forward pass.

The diffusion model is discarded at deployment time – only the cheap INR is shipped, keeping query cost at a single forward pass. Algorithm 1 states this outer loop explicitly; x_θ^* , q_ϕ^* are the INR’s two heads (parameters θ), $\hat{F}_\phi$ is the surrogate ensemble, p_ψ is the diffusion repair model (retrained on $\mathcal{D}$ each repair round), and $\mathcal{D}$ is the accumulated evaluation history.

Algorithm 1 PFR-INR: training and deployment

Require: initial evaluations X_0 ; per-round budget b ; rounds R ; repair period k

- 1: $\mathcal{D} \leftarrow \{(x, F(x)) : x \in X_0\}$; initialize θ, ϕ on $\mathcal{D}$
- 2: **for** $r = 1, \dots, R$ **do**
- 3: sample preferences Λ_r ; train θ on STCH + consistency + smoothness + diversity losses (using $\hat{F}_\phi$)
- 4: score each $\lambda \in \Lambda_r$ by ensemble variance + $\kappa \cdot$ expected hypervolume improvement; take top- b as batch B_r
- 5: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_\theta^*(\lambda), F(x_\theta^*(\lambda))) : \lambda \in B_r\}$; refit $\hat{F}_\phi$ on $\mathcal{D}$
- 6: **if** $r \bmod k = 0$ **then**
- 7: train (or update) diffusion model p_ψ on $\mathcal{D}$
- 8: sample diffusion candidates from p_ψ ; dominance-filter against $\hat{F}_\phi$, then against F
- 9: add hypervolume-improving survivors as anchors; briefly fine-tune θ on anchors only
- 10: **end if**
- 11: **end for**
- 12: **return** θ ▷ deployed as a single-forward-pass Pareto front representation

Figure 2 illustrates this pipeline: the active-evaluation loop (TrainStep, SelectBatch, true evaluation) runs every round, the diffusion repair branch runs only periodically, and both the surrogate ensemble and the diffusion model are discarded once θ is deployed.

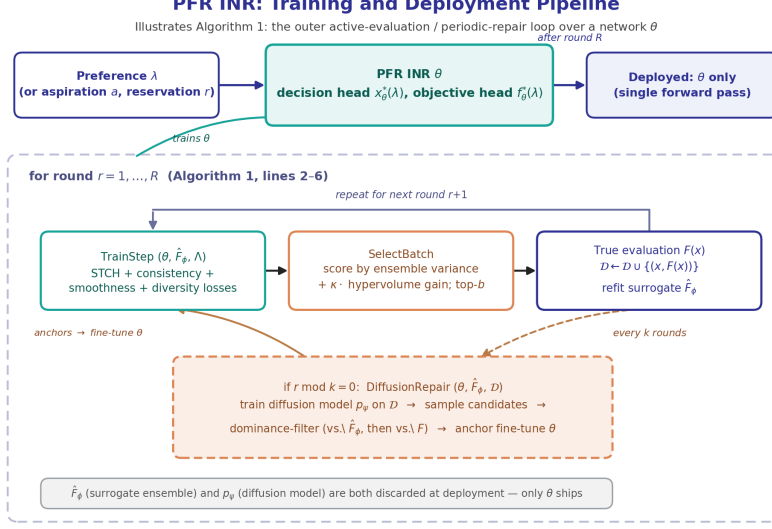


Figure 2: The PFR INR training and deployment pipeline (Algorithm 1): the active-evaluation loop runs every round, the diffusion-repair branch only periodically (every k rounds), and only the trained network θ is shipped at deployment.

5.3 Aspiration-Reservation Conditioning

As an additional interface alongside λ -conditioning, the same architecture can instead be conditioned on an aspiration point a and a reservation point r (a $2m$ -dimensional input for m objectives), trained with a piecewise-linear achievement function per objective i , with normalized deviation $d_i = (q_i - a_i)/(r_i - a_i)$:

$$g_i(d_i) = \alpha \cdot d_i + (1 - \alpha) \cdot \max(d_i, 0) + (\gamma - 1) \cdot \max(d_i - 1, 0), \quad (7)$$

which has a small slope α (e.g. 0.15) for values better than the aspiration, slope 1 in the priority zone between aspiration and reservation, and a steep slope γ (e.g. 8) beyond the reservation (Ogryczak, 2008; Ogryczak & Kozłowski, 2011, Luque et al., 2012). The per-objective terms are combined with a smooth max plus a small augmentation term to guarantee properly (not merely weakly) Pareto-optimal solutions:

$$L_{\text{ASF}}(a, r) = \text{smooth-max}_i g_i(d_i) + \rho \cdot \text{mean}_i g_i(d_i). \quad (8)$$

Because the reservation point is explicitly infeasible-tolerant by construction, an overly ambitious aspiration does not make the interface fail; the network instead returns the nearest properly Pareto-optimal compromise. This reproduces, as a single forward pass, the classical interactive procedure in which a decision maker drags a target and immediately inspects the resulting compromise, without resolving an optimization problem at every step.

6 Computational Experiments on benchmark problems

The proposed method was tested on several benchmark problems. A summary of this results is presented in this section.

Benchmark problems. All problems are bi-objective minimization problems over $x \in [0, 1]^n$ with $n = 6$ decision variables (the original definitions use $n = 30$; $n = 6$ is kept here so that the small network can reliably learn to drive the “distractor” coordinates $x_2, \dots, x_n$ to zero). They share the distance function

$$g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, \quad (9)$$

and the first objective $q_1(x) = x_1$. The Zitzler–Deb–Thiele (ZDT) problems (Zitzler, Deb & Thiele, 2000) then differ only in f_2 :

$$\text{ZDT1: } q_2 = g \left(1 - \sqrt{q_1/g} \right), \quad (10)$$

$$\text{ZDT2: } q_2 = g \left(1 - (q_1/g)^2 \right), \quad (11)$$

$$\text{ZDT3: } q_2 = g \left(1 - \sqrt{q_1/g} - \frac{q_1}{g} \sin(10\pi q_1) \right). \quad (12)$$

For every ZDT problem the Pareto-optimal set is $x_2 = \dots = x_n = 0$ (so $g = 1$), which gives the true fronts $q_2 = 1 - \sqrt{q_1}$ (ZDT1, convex), $q_2 = 1 - q_1^2$ (ZDT2, non-convex) and, for ZDT3, $q_2 = 1 - \sqrt{q_1} - q_1 \sin(10\pi q_1)$ restricted to its non-dominated parts, which form five disconnected segments.

In summary, the ZDT1–ZDT3 benchmark problems, respectively have a convex, non-convex, and disconnected Pareto front.

Experiment setup. Six implementations of PFR INR were run, all using the same small network: a Fourier-feature implicit neural representation that takes a scalar preference parameter $t \in [0, 1]$ (with $\lambda = (t, 1-t)$), seven fixed sine/cosine features, two 32-unit tanh hidden layers, a six-output sigmoid decision head and a two-output objective head. They differ only in what the network is trained against and in how true evaluations are spent (Table 2). *Weighted-sum baseline* and *PFR INR-lite* are trained directly on the true objectives, with plain weighted-sum scalarization and with the Smooth Tchebycheff objective plus the consistency, smoothness and diversity losses, respectively; PFR INR-lite tests only the training step of Algorithm 1. The *aspiration/reservation variant* has a four-dimensional input and the piecewise-linear achievement function of Section 5.3. The remaining three implementations run the closed loop on a surrogate under an evaluation budget: *Full* is Algorithm 1 as a whole, *active-only* is Full without the repair step, and *random* is Full with the batch of preferences chosen uniformly at random and without repair.

Table 2: Implementations compared in Experiments 1–5. “Surrogate” means the five-model ensemble $\hat{F}_\phi$; “true” means the exact objective. Evaluation counts are true objective calls per run.

Implementation	Network trained against	Preference batch selection	Diffusion repair	True evals	Used in
Weighted-sum baseline	true, plain weighted sum	none (all queried)	no	unrestricted	Exp. 1
PFR INR-lite	true, STCH + 3 losses	none (all queried)	no (proxy in Exp. 2)	unrestricted	Exp. 1, 2
Aspiration/reservation variant	true, achievement function	none (all queried)	no	unrestricted	Exp. 4
<i>Full</i>	surrogate, STCH + 3 losses	variance + κ · HV gain	yes ($k=2$)	≈ 205 (cap 220)	Exp. 5
<i>Active-only</i>	surrogate, STCH + 3 losses	variance + κ · HV gain	no	≈ 200	Exp. 5
<i>Random</i>	surrogate, STCH + 3 losses	uniform random	no	≈ 200	Exp. 5

Closed-loop settings. The settings below are shared by *Full*, *active-only* and *random*; the variants differ only in two switches: the scoring of the preference pool (used by *Full* and *active-only*, replaced by uniform random selection in *random*) and the diffusion repair step (used only by *Full*). The surrogate $\hat{F}_\phi$ is a bootstrap ensemble of five ridge regressions (ridge parameter 0.1) on $[1, x, x^2, 64 \text{ random Fourier features}]$, refit on all true evaluations after every round. The loop starts from $N_0=50$ uniformly random points and runs $R=5$ rounds. In each round the network takes 250 Adam steps (learning rate 0.02) on 48 preference points, minimizing the Smooth Tchebycheff scalarization of the surrogate output (smoothing $\mu=0.05$) plus the consistency (weight 0.5), smoothness (0.0005) and diversity (0.01) terms. A pool of 300 random preference vectors is then scored by ensemble variance plus $\kappa=2$ times the expected hypervolume improvement of the surrogate-predicted objective vector, and the top $b=30$ are evaluated exactly. Every $k=2$ rounds (rounds 2 and 4) a 10-step conditional denoising diffusion model with a two-layer MLP noise predictor is trained on all accumulated preference–decision pairs; it proposes 100 candidates, those not dominated by the current non-dominated set under the surrogate are ranked by hypervolume gain, up to 10 of the surviving candidates and those that are non-dominated become anchors on which the network is fine-tuned for 50 Adam steps (learning rate 0.01). Because only surviving candidates are confirmed, *Full* spends about 205 true evaluations, against about 200 for *active-only* and *random*; all are below the 220-evaluation cap.

Experiment plan. Table 3 states which implementation was run on which test problem, with what budget and number of seeds, and how it was judged. All problems have $n = 6$ decision variables and two objectives. Experiments 1–4

form a first phase that checks each component in isolation, with a cheap, exactly known objective and one trained network per method; Experiment 5 runs the closed loop under an evaluation budget and compares it with two ablations of itself.

Table 3: Experiment plan. HV: hypervolume; IGD: inverted generational distance; dist.: distance of returned solutions to the true front. Budgets are true objective evaluations.

Exp.	Problem	Implementations	Budget, seeds	Metrics	Question
1	ZDT1, ZDT2, ZDT3	weighted-sum baseline, PFR.INR-lite	unrestricted, 1 seed	HV, IGD	does the training objective handle non-convex fronts?
2	ZDT3	PFR.INR-lite with and without proxy repair	unrestricted, 1 seed	HV, IGD	does a proxy repair pass improve a disconnected front?
3	ZDT1	uncertainty-guided vs. random evaluation order	40 evals, random order averaged over 5 repeats	HV	does uncertainty- guided selection beat random order (stand-alone)?
4	ZDT2	aspiration and reservation variant	unrestricted, 1 seed	HV, IGD, dist.	does it return near-optimal points for aspiration queries ?
5	ZDT2	<i>Full</i> , active-only, random	≈ 205 of 220, 5 seeds	HV, IGD	does the closed loop reach the front within budget, and do acquisition and repair change it?

Experiment results. Table 4 collects the numerical results of Experiments 1–4. Table 5 collects the numerical results of Experiment 5. Percentages below are the hypervolume of a method divided by the hypervolume of the exact front given in the caption.

Conclusions of Experiments 1–5. Experiments 1–4 were run on a small prototype, with one trained network per method and problem (seed 0; the two orders in the active-evaluation test are averaged over repeated runs), a cheap, exactly known objective and no evaluation budget, so they establish what the

Table 4: Hypervolume and IGD from the small-scale prototype (higher HV is better; lower IGD is better). HV is measured against a problem-specific reference point and is comparable only within a problem; the HV of the exact front is 1.105 (ZDT1), 0.773 (ZDT2), 1.301 (ZDT3) and 0.656 for the Experiment 5 row (reference point (1.15, 1.15)). The Experiment 5 row is the mean $\pm$ std over 5 seeds; all other rows are a single trained network (seed 0).

Problem	Method	HV	IGD
ZDT1	baseline (weighted-sum)	1.1012	0.0031
ZDT1	PFR_INR-lite	1.0974	0.0064
ZDT2	baseline (weighted-sum)	0.2400	0.6095
ZDT2	PFR_INR-lite	0.7621	0.0070
ZDT3	baseline (weighted-sum)	1.0816	0.1349
ZDT3	PFR_INR-lite + repair pass	1.2313	0.0388
ZDT1 (active-eval, N=40)	random order / uncertainty-guided	0.382 / 0.769	—
ZDT2 (ASF-conditioned)	ASF-PFR_INR (aspiration path)	0.6961	0.0284
ZDT2 (budget=220, closed loop)	Full PFR_INR (Exp. 5)	0.6477 $\pm$ 0.0027	0.0052 $\pm$ 0.0022

Table 5: Experiment 5: the full closed loop and its two ablations, ZDT2, 5 seeds (mean $\pm$ std). HV is measured against the reference point (1.15, 1.15); the exact front has HV 0.6557. The last row is PFR INR-lite trained on the true objective without an evaluation budget (one network).

Variant	True evals	IGD	HV
<i>Full</i> (active + repair)	≈ 205	0.0052 ± 0.0022	0.6477 ± 0.0027
Active-only	≈ 200	0.0048 ± 0.0011	0.6475 ± 0.0020
Random	≈ 200	0.0048 ± 0.0020	0.6478 ± 0.0035
PFR INR-lite, unrestricted	unrestricted	0.0070	0.6446

components can do rather than how large their advantage is. Within that scope they support three conclusions.

First, the training objective decides whether a non-convex front can be represented at all. On ZDT2 the weighted-sum baseline collapses to a single corner solution and attains 31% of the exact hypervolume (IGD 0.6095), while PFR_INR-lite attains 99% (IGD 0.0070), an 87-fold difference in IGD. On the convex ZDT1 both reach at least 99% and the baseline is the more accurate (IGD 0.0031 against 0.0064), as expected, since a weighted sum can reach every point of a convex front. The baseline also lacks the consistency, smoothness and diversity terms of PFR_INR-lite, so this is a statement about the whole training objective, not about the scalarization alone.

Second, a preference-conditioned network can represent a front with several parts, but it can still lose a whole segment. On ZDT3 PFR_INR-lite with the repair pass is better than the baseline on both metrics (IGD 0.0388 against 0.1349; hypervolume 1.2313 against 1.0816, i.e. 95% against 83% of the exact value). The table does not isolate the effect of the repair pass itself, and the achieved fronts show that the fifth segment is still missing; a hypervolume of 95% would not reveal this loss, which is why a measure of how many segments of the true front are reached is needed in addition to hypervolume. Hypervolume and IGD agree on every problem in this phase: the table contains no case in which one metric favours the method that the other ranks last.

Third, the two remaining components work as stand-alone mechanisms. Selecting 40 evaluations by uncertainty and dominance reached about twice the hypervolume of a random order (0.769 against 0.382), and the aspiration-reservation network traced the ZDT2 front with an IGD of 0.0284, accurate but about four times that of the λ -conditioned network (0.0070); its four discrete decision-maker queries, one of them deliberately infeasible, all returned points on the front.

Experiment 5 is the first test of the components together. With about 205 of 220 evaluations the full closed loop reaches 99% of the exact hypervolume on ZDT2 (0.6477 ± 0.0027 of 0.656) with an IGD of 0.0052 ± 0.0022 over five seeds, comparable to PFR_INR-lite trained without any evaluation budget (IGD 0.0070). The closed loop therefore runs end to end and reaches a good front. The three variants are indistinguishable in Table 5: their means differ by far less than the spread between seeds. A separate run of the same three variants with 20 seeds confirms this: IGD 0.0062 ± 0.0033 (full), 0.0060 ± 0.0026 (active-only) and 0.0053 ± 0.0021 (random), hypervolume 0.6462, 0.6449 and 0.6467, with paired Wilcoxon p -values of 0.55–0.78 for the full variant against each of the others on both metrics. This experiment therefore gives no evidence that the acquisition rule or the repair step improves on random selection.

Two limits apply to the whole phase. The proxy repair pass of Experiment 2 and the stand-alone selection of Experiment 3 are not the diffusion repair and the ensemble acquisition of Algorithm 1, so Experiments 1–4 do not test those mechanisms as the closed loop uses them; and Experiment 5 uses a single non-convex problem, on which even random selection does well. Whether acquisition and repair help when the front is disconnected, and how the loop compares with

published methods under the same evaluation budget, are open questions that this paper leaves for future work.

Visualization of experiments. All figures come from a single trained network per method (seed 0), except the evaluation-efficiency figure, which shows means over repeated runs; they illustrate behaviour and are not statistical comparisons. In the figures, the axes f_1 and f_2 denote the objectives q_1 and q_2 .

Experiments 1 and 2 (Figure 4). The panels show the true front (grey) and the fronts of the weighted-sum baseline (orange), PFR_INR-lite (blue) and, on ZDT3 only, PFR_INR-lite after the repair pass (green, dashed). On ZDT2 the baseline is not visible because all its non-dominated points coincide in the corner solution $(f_1, f_2) = (0, 1)$, which is why its IGD of 0.6095 is so large; PFR_INR-lite lies on the front from end to end except for a small deviation near $f_1 \approx 0.95$. On ZDT1 both curves follow the front. On ZDT3 neither network reaches the fifth segment, and the curves leave the front between segments along straight connecting lines, which are not solutions. The dashed curve lies visibly closer to the second, third and fourth segments than the unrepared curve, but it does not recover the fifth. The short vertical stroke at the left edge of the blue curves is an under-trained preference direction that lies outside the plotted range; it is included in the reported metrics.

Experiment 3 – evaluation-efficiency (Figure 3). Each curve is the hypervolume of the set of points evaluated so far, on ZDT1 with a budget of 40 true evaluations. A random order is ahead for roughly the first 13 evaluations; the uncertainty-guided order then overtakes it, rises in steps as it reaches new regions of the front, and ends at 0.769 against 0.382 for the random order. The curves are means over repeated runs and no spread is shown, so the figure demonstrates the effect but not its significance.

Experiment 4 (Figure 5). In panel (a) the points returned while the aspiration point slides along the dashed line lie on the true ZDT2 front over most of its length, from about $f_1 = 0.2$ to $f_1 = 0.93$; the two ends of the front are not reached, which is consistent with an IGD of 0.0284 rather than a value near that of the λ -conditioned network (0.0070). In panel (b) each open marker is a decision-maker query and the filled marker of the same shape is the returned solution; all four lie on the front. The infeasible query, whose aspiration point lies far below the front, is answered with a balanced point near the middle of the front rather than with a failure.

Experiment 5 (closed-loop Figure 6). The full loop, the active-only variant and random acquisition follow the ZDT2 front closely; the active-only curve is hidden beneath the full one. The one visible difference is a bump in the random-acquisition curve between $f_1 \approx 0.43$ and 0.54, which belongs to the single seed shown; the five-seed means of the Experiment 5 table do not separate the three variants. The narrow vertical stroke at the left edge is again an under-trained preference direction that is cropped from view and counted in the metrics.

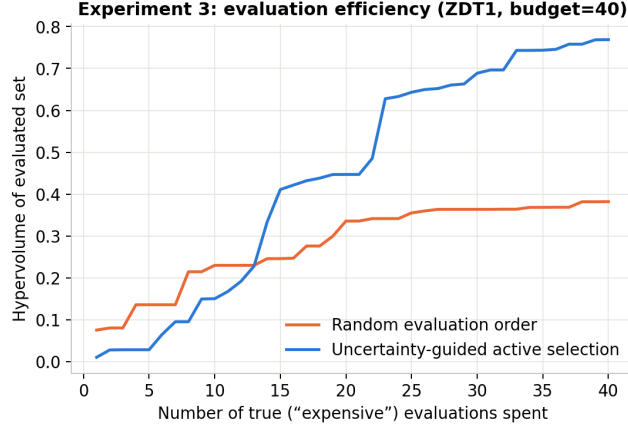


Figure 3: Experiment 3: hypervolume of the evaluated set against the number of true evaluations on ZDT1 (budget 40), for a random evaluation order and for uncertainty-guided selection; means over repeated runs.

Conclusions drawn from the visualisations. The figures add four things that the table cannot show. First, they show *where* a method fails, not only how much: the weighted-sum baseline fails by returning one point on ZDT2, and every network fails on ZDT3 by omitting the fifth segment, a loss that the hypervolume of the repaired network (95% of the exact value) conceals. Second, the repair pass acts locally. It moves points toward the segments the network already covers, but in this experiment it does not recover a segment that the network does not reach, so it acts as a refinement step rather than a remedy for a lost segment. Third, the active-evaluation advantage is not uniform: random order is better while the evaluated set is small, and the uncertainty-guided order appears to pay off only after it has found the non-dominated region, so the benefit may depend on the budget. Fourth, the aspiration interface covers the interior of the ZDT2 front and returns a reasonable compromise for an infeasible request, but the figure does not show that the two ends of the front can be reached by this interface, and the closed-loop figure gives no visual reason to prefer the full loop over its ablations on this problem. These observations rest on one network per method and, for Experiments 1–4, on small prototypes; they motivate larger, seed-paired comparisons on disconnected fronts.

The aspiration-reservation variant was trained on ZDT2 and tested two ways (Figure 5). Sliding a single aspiration point along a path with a fixed reservation margin reproduced the true front closely (IGD 0.0284, HV 0.6961 against 0.773 for the exact front). Four discrete decision-maker queries, including one deliberately infeasible aspiration (“ambitious on both objectives”), all returned points within 0.0001–0.0002 (Euclidean, normalized units) of the true front, and the infeasible query resolved to a properly Pareto-optimal compromise rather than failing.

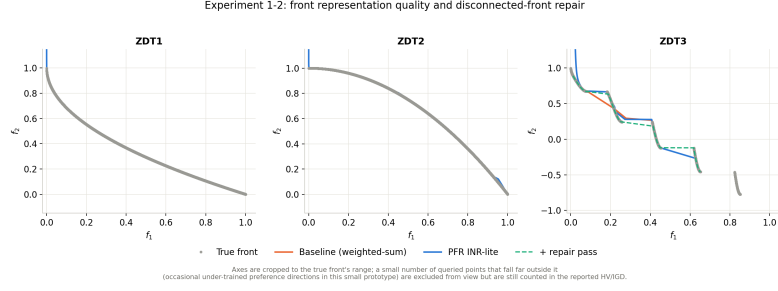


Figure 4: Achieved fronts vs. the true front, baseline and PFR_INR-lite, on ZDT1–ZDT3; ZDT3 additionally shows the front after the repair pass.

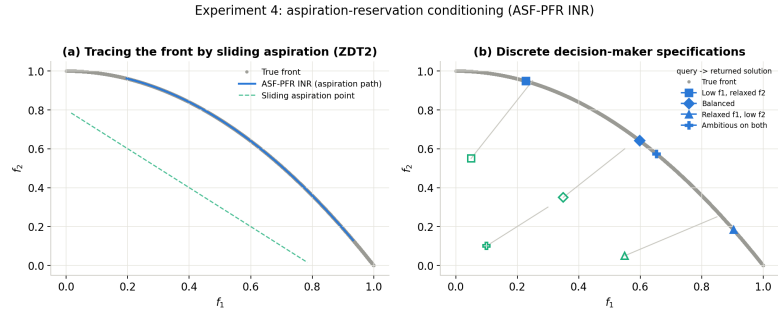
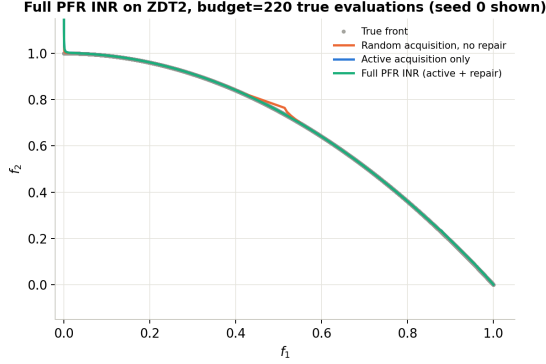


Figure 5: Aspiration-reservation conditioning on ZDT2: (a) front traced by a sliding aspiration point; (b) four discrete decision-maker queries and their returned solutions.



Cropped to the true front's range; occasional under-trained points that fall far outside it are excluded from view but still counted in the reported IGD/HV.

Figure 6: Achieved fronts on ZDT2 under a 220-evaluation budget, seed 0: full PFR.INR, active acquisition only, and random acquisition without repair, against the true front. The active-only curve lies beneath the full curve. Axes are cropped to the true front’s range; occasional under-trained points outside it are excluded from view but counted in the reported metrics.

7 Benchmark I–IV: Exact Envelope Discovery on Four Linear MOLPs

A stricter, complementary question than the hypervolume and IGD of Experiments 1–5 is whether the entire efficient set of a model can be found and proven complete outright. For linear multi-objective linear programs (MOLPs) the efficient set is a union of finitely many flat facets. PFR.INR-lite can be measured without sampling error on the reference side. Table 6 lists the four benchmarks.

Table 6: The four benchmark MOLPs

#	Model	Dim.	$ V_E $	Facets	Nadir z^{nad}
I	Benson & Sayin (1997)	$3 \times 3 \times 6$	6	4	$(0, 2, 0)$
II	Alves & Costa (2009)	$3 \times 10 \times 5$	12	6	$(-235.00, 93.16, 121.52)$
III	Unit simplex	$3 \times 3 \times 1$	3	1	$(0, 0, 0)$
IV	Two-plane simplex	$3 \times 3 \times 2$	4	2	$(0, 0, 0)$

Benchmark I reconstructs Benson and Sayin’s (1997) model. Benchmark II (Alves M.J., Costa J.P. (2009)) is *problem3.10.5*, a larger model from the paper’s appendix (10 variables, 5 constraints). Benchmarks III–IV are geometric

controls (a unit simplex; a two-plane polytope whose two facets meet in a ridge, confirming the region stays convex). The difference from Experiments 1–5 is the reference: there, hypervolume and IGD were computed against an approximate or sampled front; here the reference front itself is exact and formally certified complete, so coverage gaps are measured without error on the reference side.

7.1 PFR_INR-lite Run on Benchmarks I–IV

A PFR_INR-lite (same network family for Experiments 1–5) was trained from scratch on all four benchmarks and checked against the exact envelope. For Benchmarks I–IV the full LP (C, A, b) is known, so the network maps a sampled preference weight to $x \geq 0$ (softplus) with a soft penalty for violating $Ax \leq b$, scored via STCH normalized by $q^I - q^N$; every reported point was then projected by a small quadratic program onto the exact feasible set, so all results below are genuinely feasible, not merely approximately so.

Coverage is measured by the same per-point distance IGD averages: for the exact reference cloud $R = \{r_i\}$ sampled on the true facets and the achieved set A , both normalized by $s = q^I - q^N$, the *gap* at r_i is $g_i = \min_{a \in A} \|r_i/s - a/s\|_2$. **Mean gap** = $\frac{1}{M} \sum_i g_i$ is exactly IGD (hence the matching columns below); **max gap** = $\max_i g_i$ is the single worst-covered point. **HV** is hypervolume against the exact nadir, same normalization (higher is better).

Table 7: PFR_INR-lite vs. the exact AR-ES envelope

#	IGD (norm.)	mean gap	max gap	HV	dominated
I	0.0222	0.0222	0.0749	0.3921	1.0%
II	0.0202	0.0202	0.0706	0.4280	42.2%
III	0.0210	0.0210	0.0668	0.1454	7.2%
IV	0.0183	0.0183	0.0555	0.2029	0.9%

All four IGD values (0.018–0.022) are about three times those of PFR_INR-lite on ZDT1 and ZDT2 (0.0064–0.0070); the problems, dimensions and normalizations differ, so the values are not directly comparable, but the three-objective faceted fronts are not represented more closely than the two-objective curves of Experiments 1 and 2.

Benchmark II’s high dominated-point fraction (42.2% vs. under 10% elsewhere) reflects it being the hardest instance: 10 decision variables, six facets, and sparse LP-vertex solutions a smooth network must approximate closely. Figure 7 plots only each benchmark’s Pareto-optimal (non-dominated) points – the dominated-but-feasible ones are left out, since on Benchmark II alone they would be 42.2% of the raw sample and visually clutter the panel without adding anything beyond the fraction already given above. The plotted points visibly hug every facet and ridge, including Benchmark II’s five- and four-vertex facets;

the omitted dominated points were separately checked and correctly fall inside the envelope rather than outside it.

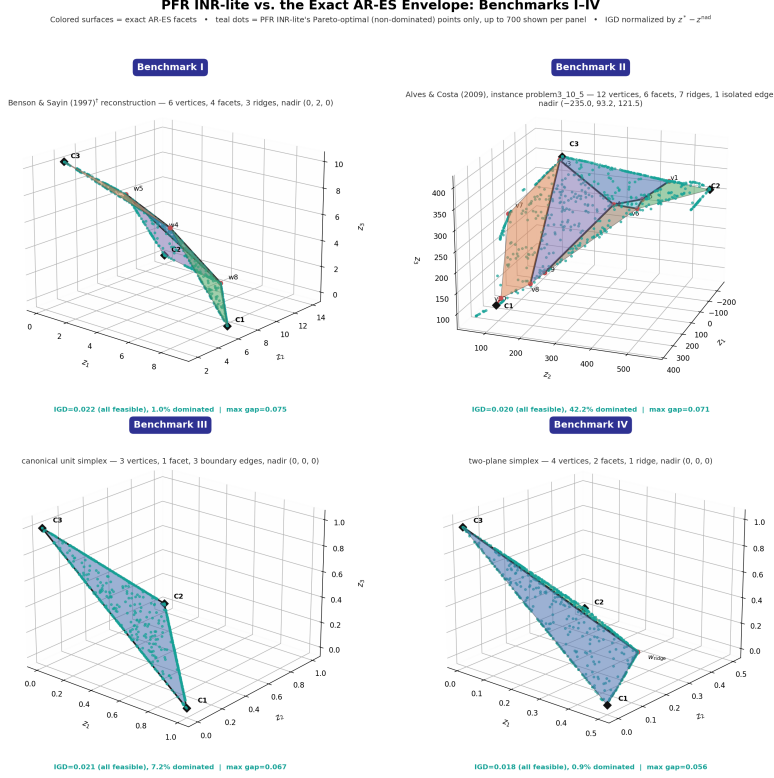


Figure 7: PFR_INR-lite's achieved front, Pareto-optimal (non-dominated) points only, over the exact AR-ES facets.

On Benchmark II, the mean gap to the isolated efficient edge C_1-v_{10} (0.027) is worse than to the regular facets (0.012–0.020), and anchor C_1 is the single hardest point to reach (0.025 away). Isolated edges and the anchors next to them are where coverage judged by nearest-neighbour distance to an external sample is least reliable: an independent check of AR-ES against SPFR (Structured Pareto Front Representation; Makowski et al., 2026) found a true gap of 27.30 at vertex (8, 2, 8) of Benchmark I where the naive radius was 12.50, and a gap of 14.60 around vertex v_7 and the isolated edge of Benchmark II where the naive maximum gap was 6.21. In this run the neural learner is also least accurate at such places; this is one trained network, so it is an observation and not a general result.

Restricting the sample to non-dominated points keeps 57.8% of Benchmark II's 2000 points (the rest are dominated points that lie inside the envelope), against 92.8–99.1% for Benchmarks I, III and IV; IGD on this subset is essen-

tially unchanged from the full sample (0.0203 against 0.0202).

In summary, based on the model formulation we can enumerate the efficient vertices and the exact nadir of all four linear models and so provide a formally certified reference front, and measured against it PFR INR-lite covers each efficient set closely, with a mean gap of 0.018–0.022 and a worst-case gap of at most 0.075 in normalized units. Coverage judged only by nearest-neighbour distance to an external sample can be badly optimistic on these models, and isolated edges and the anchors next to them are also the points that the neural learner reaches least accurately. The four benchmarks test the representation with the true objective and linear structure, not the closed loop under an evaluation budget, and they do not show that PFR INR-lite is better than other methods; their value is an exact standard against which coverage can be measured.

8 Summary

This paper proposed PFR_INR, a single queryable network that maps a preference, or an aspiration and reservation pair, to a solution of a continuous multi-objective problem, is trained on a surrogate under an evaluation budget, and is periodically repaired by a diffusion model. The evidence collected so far shows that the components behave as intended when tested on their own, and that the assembled loop runs and reaches a good front, but it does not yet show that the combination is better than simpler alternatives.

On the non-convex ZDT2 front the weighted-sum baseline collapses to a single point (IGD 0.6095), whereas the network trained with the Smooth Tchebycheff objective follows the whole front (IGD 0.0070); on the convex ZDT1 front the two behave alike. On the disconnected ZDT3 front the network improves on the baseline, but it still loses a whole segment, which a hypervolume of 95% of the exact value hides, and the simplified repair pass moves points closer to the segments already covered without recovering the missing one. Selecting 40 evaluations by uncertainty reached about twice the hypervolume of a random order on ZDT1 (0.769 against 0.382), and the aspiration–reservation network returned points on the front for every tested query, including a deliberately infeasible one. When the true objective is replaced by a surrogate and a budget of 220 evaluations, the full closed loop still reaches 99% of the exact hypervolume on ZDT2 (IGD 0.0052 ± 0.0022 over five seeds), but the variants with active acquisition only and with random acquisition are statistically indistinguishable from it. The experiments therefore show that the loop works end to end, not that its acquisition rule or its repair step improves on random selection, and the single non-convex problem used for this test is one on which even random selection does well.

Checked against the exact, formally certified efficient sets of the four linear benchmarks, PFR_INR-lite covers each set with a mean gap of 0.018 to 0.022 in normalized units and a worst-case gap of at most 0.075. These benchmarks use the true objective rather than a surrogate, so they test the representation and not the closed loop.

The first four experiments use one trained network per method, a cheap exactly known objective, and simplified stand-ins for the repair and selection mechanisms of Algorithm 1; no comparison with the published preference-conditioned and surrogate-based methods (Navon et al., 2021; Lin et al., 2022; Cheng et al., 2025) has been made (it is planned in future work); and no genuinely expensive objective has been tested. The natural next steps are a paired multi-seed comparison of the closed loop with random selection and with those methods on problems with disconnected fronts, where lost segments can be measured directly, a remedy for such lost segments, a test on a real expensive black-box problem.

9 References

- Alves M.J., Costa J.P. (2009) An Exact Method for Computing the Nadir Values in Multiple Objective Linear Programming. *European Journal of Operational Research*, **198**(2), 637-646.
- Benson H.P., Sayin S. (1997) Towards Finding Global Representations of the Efficient Set in Multiple Objective Mathematical Programming. *Naval Research Logistics*, **44**(1), 47-67.
- Cheng J., Xue B., Zhang Q. (2025) Parametric Pareto Set Learning for Expensive Multi-Objective Optimization (PPSL-MOBO). arXiv:2511.05815.
- Deb K., Pratap A., Agarwal S., Meyarivan T. (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**(2), 182-197.
- Granat J., Makowski M. (2000) ISAAP – Interactive Specification and Analysis of Aspiration-Based Preferences. *European Journal of Operational Research*, **122**(2), 469-485.
- Ho J., Jain A., Abbeel P. (2020) Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, **33**, 6840-6851.
- Hoang L.P., Le D.D., Tran A.T., Tran N.T. (2023) Improving Pareto front learning via multi-sample hypernetworks. *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**(7), 7875-7883.
- Hotegni S.S., Peitz S. (2025) SPREAD: Sampling-based Pareto front refinement via efficient adaptive diffusion. arXiv:2509.21058 (ICLR 2026).
- Jin Y. (2011) Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, **1**(2), 61-70.
- Jin Y., Wang H., Chugh T., Guo D., Miettinen K. (2019) Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, **23**(3), 442-458.
- Liang J., Lou Y., Yu M., et al. (2025) A survey of surrogate-assisted evolutionary algorithms for expensive optimization. *Journal of Membrane Computing*, **7**, 108-127. <https://doi.org/10.1007/s41965-024-00165-w>
- Lin X., Yang Z., Zhang X., Zhang Q. (2022) Pareto Set Learning for Expensive Multi-Objective Optimization. *Advances in Neural Information Processing Systems (NeurIPS)*, **35**; arXiv:2210.08495.

- Luque M., Miettinen K., Ruiz A.B., Ruiz F. (2012) A Two-Slope Achievement Scalarizing Function for Interactive Multiobjective Optimization. *Computers & Operations Research*.
- Ma P., Du Y., et al. (2020) Efficient Continuous Pareto Exploration in Multi-Task Learning. *ICML 2020*, PMLR v119.
- Mahapatra D., Rajan V. (2020) Multi-task learning with user preferences: Gradient descent with controlled ascent in Pareto optimization. *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, PMLR v119.
- Makowski M., Granat J., Shekhovtsov A., Nahorski Z., Zhao J. (2024) pyMCMA: Uniformly Distributed Pareto-Front Representation. *SoftwareX*, **27**, 101801.
- Makowski M., Granat J., Nahorski Z., Shekhovtsov A., Ren H., Zhao J. (2026) Structured Pareto Front Representation (SPFR). Zenodo preprint, DOI: 10.5281/zenodo.19704566.
- Navon A., Shamsian A., Chechik G., Fetaya E. (2021) Learning the Pareto Front with Hypernetworks. *International Conference on Learning Representations (ICLR 2021)*; arXiv:2010.04104.
- Ogryczak W. (2008) Reference Point Method with Importance Weighted Partial Achievements. *Journal of Telecommunications and Information Technology*, 2008(4).
- Ogryczak W., Kozłowski B. (2011) Reference Point Method with Importance Weighted Ordered Partial Achievements. *TOP*, **19**(2), 380-401.
- Ruchte M., Grabocka J. (2021) Scalable Pareto Front Approximation for Deep Multi-Objective Learning. *IEEE International Conference on Data Mining (ICDM)*; arXiv:2103.13392.
- Sayin S. (2000) Measuring the Quality of Discrete Representations of Efficient Sets in Multiple Objective Mathematical Programming. *Mathematical Programming*, **87**(3), 543-560.
- Wierzbicki A.P. (1980) The Use of Reference Objectives in Multiobjective Optimization. In: Fandel & Gal (Eds.), *Multiple Criteria Decision Making Theory and Applications*, Lecture Notes in Economics and Mathematical Systems vol. 177, Springer, pp. 468-486.
- Wierzbicki A.P. (1982) A Mathematical Basis for Satisficing Decision Making. *Mathematical Modelling*, **3**(5), 391-405.
- Yao Y., Pan Y., Li J., Tsang I.W., Yao X. (2024) PROUD: PaRetO-guided diffusion model for multi-objective generation. *Machine Learning*, **113**, 6511-6538. <https://doi.org/10.1007/s10994-024-06575-2>
- Zhang Q., Li H. (2007) MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, **11**(6), 712-731.
- Zhang Q., Liu W., Tsang E., Virginas B. (2010) Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, **14**(3), 456-474.
- Zitzler E., Deb K., Thiele L. (2000) Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, **8**(2), 173-195.