ARCHIVE-GUIDED INFERENCE REALLOCATION FOR CONDITIONAL GENERATIVE MODELS IN OFFLINE MULTI-OBJECTIVE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Conditional generative models provide a promising approach to offline multi-objective optimization, but existing inference procedures typically fix all conditioning targets before sampling begins. This prevents them from using information revealed by partially generated solutions. We introduce Archive-Guided Reallocation (ARC), a closed-loop inference controller for pretrained conditional diffusion and flow models. ARC audits intermediate generations, records their estimated trade-offs in an evolving archive, and redirects selected trajectories toward new target objective vectors while preserving their current states and the total number of backbone evaluations. Our experiments separate two effects-reallocating the inference budget improves Pareto-frontier quality, while selecting new targets from the archive improves convergence. On Off-MOO-Bench, closed-loop reallocation improves mean full-set hypervolume by 4.49% over a compute-matched open-loop control and improves all 15 engineering tasks over its backbone. Archive-guided fronts dominate random-target fronts more often (C-metric 0.36 vs. 0.16) at comparable objective-space occupancy. We further show that the benefit depends on how much of sampling remains steerable after a trajectory is redirected, consistent with the weaker, statistically non-significant effects on a flow-matching backbone whose direct guidance is active during the final 20% of integration. ARC requires no retraining, trajectory restarts, true-objective queries during generation, or additional backbone evaluations.

1 INTRODUCTION

Multi-objective optimization (MOO) seeks diverse Pareto-optimal designs representing competing objectives, such as activity versus toxicity in drug design or performance versus cost in engineering (Miettinen, 1999; Deb, 2001). New evaluations may require expensive experiments or simulations (Angermueller et al., 2020). Whereas online MOO queries objectives during search (Deb et al., 2002; Daulton et al., 2020; 2021), offline MOO uses only a fixed evaluated dataset (Trabucco et al., 2022; Xue et al., 2024). It must recover useful trade-offs without exploiting unreliable predictions (Trabucco et al., 2021; Yu et al., 2021; Qi et al., 2022).

Conditional diffusion and flow models generate designs from desired objective vectors (Shrestha et al., 2026; Yuan et al., 2025). Their conditioning targets are typically fixed before sampling, preventing adaptation when partial trajectories reveal unreachable, redundant, or already covered regions. Closing this loop requires two decisions: *which trajectories to redirect* reallocates remaining computation, while *where to send them* determines which objective-space regions that computation explores.

This distinction motivates two questions: (1) does reallocating ongoing trajectories improve the Pareto frontier when the backbone, initial noise, sampling schedule, and network-evaluation budget are held fixed, and (2) does selecting new targets from previously reached regions further improve the frontier, or does it instead improve the convergence of the returned set toward the best-known trade-offs? We answer these with *Archive-Guided Reallocation* (ARC), a closed-loop inference controller for pretrained conditional diffusion and flow models. ARC periodically evaluates partial generations, stores their predicted objective vectors in an evolving archive, and redirects selected trajectories toward new targets drawn from the archive's non-dominated regions. Redirected trajec-

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.

tories retain their current generative states, so ARC requires no retraining, restarts, true-objective queries during generation, or additional backbone evaluations.
These two decisions have separable effects: reallocation improves the frontier, while archive-guided targeting primarily improves the convergence of the returned set. On 48 Off-MOO-Bench tasks (Xue et al., 2024), ARC improves full-set hypervolume by 4.49% over a compute-matched open-loop control and improves all 15 engineering tasks relative to its backbone. Archive-guided targets show no reliable frontier advantage over random targets, but improve 50th-percentile hypervolume by 15.87% and produce fronts that dominate random-target fronts more often (C-metric 0.36 vs. 0.16). The gain also depends on how much of sampling remains steerable: restricting ARC to the final 20% of sampling on a diffusion backbone cuts its mean full-set gain from $+4.56\%$ to $+1.05\%$. A flow backbone whose direct guidance acts only in the final 20% of integration shows weaker, non-significant effects, consistent with this limit.
Our contributions are threefold:

- We introduce ARC, a controller that audits partial generations and redirects ongoing trajectories toward archive-supported targets, without retraining, restarts, or additional backbone evaluations.
- Under identical checkpoints, noise, schedules, and evaluation budgets, we show that closing the inference loop involves two mechanisms with distinct effects: reallocation improves the Pareto frontier, while archive-guided target selection improves convergence of the returned set toward the best-known trade-offs.
- We show that the gain depends on how much of sampling remains steerable: restricting ARC to the final 20% of sampling on a diffusion backbone removes most of it, consistent with the weaker, non-significant effects on a flow backbone whose direct guidance acts only late in integration.

2 Background and Preliminaries

Offline multi-objective optimization. Let $f : \mathcal{X} \to \mathbb{R}^m$ map each design x to m objectives, all formulated for minimization. A design x dominates x' if $f_j(x) \leq f_j(x')$ for every objective j, with strict inequality for at least one objective. The set of non-dominated designs is the Pareto set, and their corresponding objective vectors form the Pareto front (Miettinen, 1999). Offline MOO assumes access only to a fixed dataset

$$\mathcal{D} = \{(x_i, f(x_i))\}_{i=1}^{N},$$

with no additional queries to f during optimization (Xue et al., 2024). The goal is to return a diverse set of designs that closely approximates the Pareto front.

We evaluate a finite set of objective vectors Y using hypervolume:

$$\mathrm{HV}(Y; z) = \mathrm{vol}\left(\bigcup_{y \in Y} [y, z]\right),$$

where z is a fixed reference point dominated by every point under consideration. Hypervolume captures both solution quality and coverage: each point contributes only the objective-space region that is not already dominated by the remaining points in the set.

Conditional diffusion models. Diffusion models generate samples by gradually denoising an initial random state. A conditional model predicts the denoising direction using both the current state and a conditioning request c, which here represents a desired objective trade-off. Classifier-free guidance combines conditional and unconditional predictions,

$$\widehat{\epsilon}_{\mathrm{cfg}} = (1 + \gamma)\widehat{\epsilon}_\theta(x_t, t, c) - \gamma\widehat{\epsilon}_\theta(x_t, t, \varnothing),$$

where γ controls conditioning strength (Ho et al., 2020; Ho & Salimans, 2022). Pareto Conditioned Diffusion (PCD) applies this formulation to offline MOO by conditioning generation directly on target objective vectors (Shrestha et al., 2026).

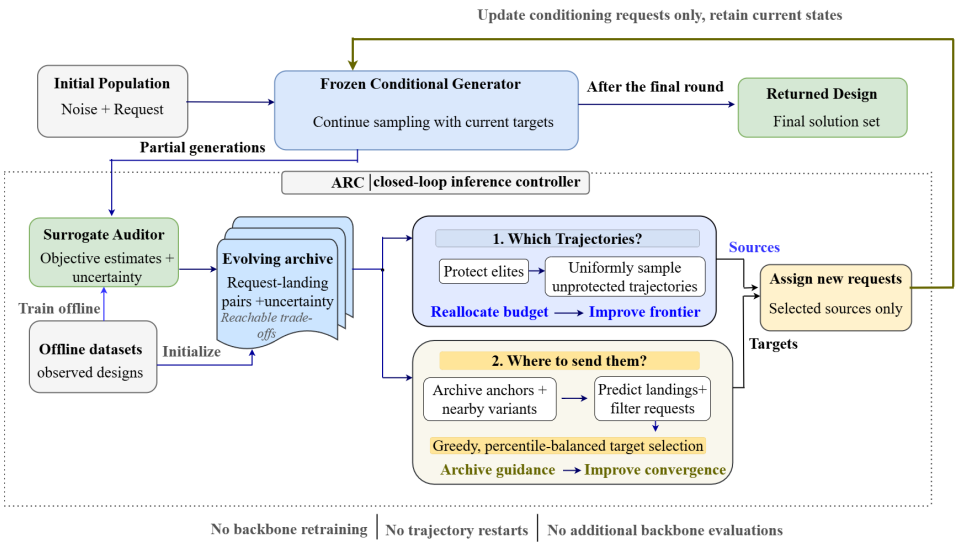

Figure 1: An overview of the ARC closed-loop inference controller. The frozen generator samples noise-request pairs and returns the final design set after the last round. Partial generations are audited and logged into an evolving archive, which guides two decisions, shown in blue and yellow: which trajectories to redirect, and where to send them. These are combined to assign new requests, while trajectories retain their generative states.

Conditional flow models. Flow-matching models generate designs by integrating a learned time-dependent vector field from a simple base distribution to the data distribution (Lipman et al., 2023). Given a request c, a trajectory evolves according to

$$\frac{dx_t}{dt} = v_\theta(x_t, t, c).$$

ParetoFlow adapts this process to offline MOO using multi-objective predictor guidance and neighboring evolution (Yuan et al., 2025). Although diffusion and flow models use different dynamics, both produce an intermediate state conditioned on a request. ARC operates on this shared structure: it updates the request between inference rounds while retaining the trajectory's current state.

3 METHOD

ARC wraps a pretrained PCD backbone (Shrestha et al., 2026), dividing sampling into R rounds and revising requests from intermediate audits while preserving generative states and backbone-evaluation counts. True objectives are used only for final evaluation. Figure 1 separates source selection from destination selection; Section 5.2 tests their frontier and convergence effects.

3.1 AUDITING & ARCHIVE STATE

After round r the controller holds an archive

$$\mathcal{A}_r = \mathcal{A}_0 \cup \{(c_j, \widehat{y}_j, u_j)\}_{j \le rN},$$

where c_j is the request that was sent, $\widehat{y}_j$ is the auditor's estimate of the resulting design's objectives, and u_j is the auditor's uncertainty. The archive is initialized with the non-dominated slice of the offline dataset; those initial entries use their observed objectives and zero ensemble uncertainty.

The auditor is an ensemble of $E = 3$ bootstrap-resampled MLPs per objective, trained per task and seed on the offline dataset only. It is used *ordinally*: the controller needs to know which candidate landing is better, not its exact value. Further details of the auditor are provided in Section D.4.

3.2 A SINGLE BALANCED OBJECTIVE

The benchmark evaluates the returned set at three percentiles. Optimizing only the full-set hypervolume may favor a few exceptional solutions while giving less attention to the overall quality of the returned set. We therefore freeze three archive strata at the start of each feedback round: $\mathcal{A}_r^{(100)}$, $\mathcal{A}_r^{(75)}$, and $\mathcal{A}_r^{(50)}$, containing the best $q\%$ of the archive according to the summed normalized objectives. For a candidate landing y, its normalized marginal gain within stratum q is

$$G_q(y; \mathcal{A}_r) = \frac{\mathrm{HV}\Big(\mathcal{A}_r^{(q)} \cup \{y\}\Big) - \mathrm{HV}\Big(\mathcal{A}_r^{(q)}\Big)}{\mathrm{HV}\Big(\mathcal{A}_r^{(q)}\Big) + \varepsilon}.$$

ARC combines these gains into a single balanced utility:

$$\widehat{U}_{\mathrm{bal}}(y; \mathcal{A}_r) = \omega_{100} G_{100}(y) + \omega_{75} G_{75}(y) + \omega_{50} G_{50}(y), \tag{1}$$

where $(\omega_{100}, \omega_{75}, \omega_{50}) = (0.50, 0.30, 0.20)$ is fixed across all tasks. Normalization prevents a stratum from dominating solely through its HV scale. The strata remain fixed during each round so that candidate selection operates on a fixed set objective, preserving the submodularity used in Section 4.

3.3 REALLOCATION: UNIFORM SOURCES, ARCHIVE-GUIDED DESTINATIONS

Each round protects elites, constructs and filters archive-guided destinations, uniformly samples eligible source trajectories, and greedily assigns new requests.

Protecting strong trajectories. ARC first identifies trajectories that should retain their current requests. For each live trajectory i, it computes its leave-one-out hypervolume contribution across the three frozen archive strata:

$$S_i = \sum_{q \in \{100,75,50\}} \omega_q \frac{\mathrm{HV}(\mathcal{B}_q) - \mathrm{HV}(\mathcal{B}_q \setminus \{\widehat{y}_i\})}{\mathrm{HV}(\mathcal{B}_q) + \varepsilon}, \qquad \mathcal{B}_q = \mathcal{A}_{r-1}^{(q)} \cup Y_r,$$

where Y_r contains the audited objective vectors of the current live trajectories. A trajectory that is among the top contributors in any percentile view is protected. Protected trajectories continue toward their current requests, while the remaining trajectories become eligible for reallocation.

Constructing archive-guided destinations. ARC constructs candidate destinations from the nondominated objective vectors in the current archive:

$$\mathcal{F}_r = \mathrm{ND}(\{\widehat{y} : (c, \widehat{y}, u) \in \mathcal{A}_r\}).$$

These are previously reached regions according to the auditor. ARC samples without replacement at most $K_{\max} = 64$ anchors from $\mathcal{F}_r$. Letting

$$K = \min\{K_{\max}, |\mathcal{F}_r|\},$$

the sampled anchor set is

$$\mathcal{P}_r = \{b_1, \ldots, b_K\}.$$

ARC retains each anchor as a candidate request and generates two independently sampled, scale-controlled variants in the improving direction. The resulting pool $\mathcal{C}_r$ contains at most $3K_{\max} = 192$ candidates. Anchors provide supported requests; nearby variants expand small fronts to avoid underusing the reallocation budget. Appendix A.1 gives the exact construction.

Each candidate derives from one anchor, never coordinatewise minima across designs. Requests need not be realized exactly.

Predicting candidate landings. Because a generated design may not reach its requested objective vector, ARC predicts where each candidate request is likely to land. It learns this relationship from the request–landing pairs stored in the archive:

$$\{(c_i, \widehat{y}_i) : (c_i, \widehat{y}_i, u_i) \in \mathcal{A}_r\},$$

where c_i is a previous conditioning request and $\widehat{y}_i$ is the auditor's estimate of the objectives subsequently reached by that trajectory. The first reallocation occurs only after the initial trajectories have been audited, so request-landing observations are available before this model is used.

ARC fits a ridge-regularized affine model within each reference-direction cell and uses a global affine model as a fallback. A local model is used only when its cell contains sufficient data; otherwise, ARC uses the global model. For a candidate v, let $\widetilde{y}(v)$ denote its predicted landing and let $u_{\mathrm{real}}(v)$ denote the held-out prediction error of the model used.

If the normalized distance between v and $\widetilde{y}(v)$ is too large, ARC treats the request as unlikely to be realized and discards it. The candidates that pass this check form the retained destination pool $\mathcal{C}_r^{\mathrm{ret}}$. Appendix A.2 specifies the realization models, reliability conditions, and filtering rule.

Selecting sources and destinations. Let $\mathcal{U}_r$ contain the unprotected live trajectories, let ρ be the target reallocation rate, and let Q be the total number of live trajectories. The number redirected in round r is

$$B_r = \min\left\{\lceil \rho Q \rceil,\ |\mathcal{U}_r|,\ |\mathcal{C}_r^{\mathrm{ret}}|\right\}.$$

ARC samples B_r source trajectories uniformly from $\mathcal{U}_r$. Therefore, S_i is used only to determine which trajectories are protected, not to rank the unprotected trajectories.

ARC next selects B_r replacement destinations from $\mathcal{C}_r^{\mathrm{ret}}$. Given the set Y of predicted landings already selected in the current round, it scores candidate v by

$$a(v;Y) = \sum_{q\in\{100,75,50\}} \omega_q \frac{\mathrm{HV}\Big(\mathcal{A}_r^{(q)} \cup Y \cup \{\widetilde{y}(v)\}\Big) - \mathrm{HV}\Big(\mathcal{A}_r^{(q)} \cup Y\Big)}{\mathrm{HV}\Big(\mathcal{A}_r^{(q)}\Big) + \varepsilon} - \lambda_u u_{\mathrm{real}}(v), \quad (2)$$

Here $\lambda_u = 0.5$. The hypervolume term rewards destinations that improve the three percentile views, while the residual penalty lowers the score of candidates whose predicted landings are less reliable.

Starting with $Y = \varnothing$, ARC repeatedly selects

$$v^\star = \arg\max_{v\in\mathcal{C}_r^{\mathrm{ret}}} a(v;Y), \quad (3)$$

adds $\widetilde{y}(v^\star)$ to Y, removes $v^\star$ from the candidate pool, and recomputes the remaining scores. Once a region is represented in Y, nearby candidates add less hypervolume, encouraging later selections to cover different regions.

Finally, ARC assigns the B_r selected requests one-to-one to the B_r uniformly sampled source trajectories. Only their conditioning requests are replaced; every trajectory continues from its current diffusion state.

4 Theoretical Analysis

We analyze one feedback round. Reallocation replaces the potential landings of redirected trajectories with new landings. The following exact decomposition states when this improves the returned set.

Proposition 1 (Exact reallocation decomposition)**.** *Let L be the landings obtained without redirection, J the redirected trajectories, L_{-J} the landings of those left untouched, and Y_J their realized replacements. Then*

$$\mathrm{HV}(L_{-J} \cup Y_J) - \mathrm{HV}(L) = g(Y_J \mid L_{-J}) - \ell(J \mid L),$$

where g is the hypervolume gained by the replacements over L_{-J} and ℓ is the hypervolume lost by abandoning J. Hence, reallocation improves the returned set if and only if $g(Y_J \mid L_{-J}) > \ell(J \mid L)$.

Source selection determines the potential loss, while destination selection determines the replacement gain. Because the unredirected outcomes of released trajectories are not observed, the proposition characterizes improvement but cannot certify it from a single run. As an algebraic identity, it requires no assumption about how the replacement landings are generated.

Algorithm 1 ARC: archive-guided reallocation

Input: frozen backbone G_θ, auditor, $N = Q$ trajectories, R rounds, guidance $\gamma = 2.5$
Initialize $\mathcal{A}_0$ with the non-dominated slice of $\mathcal{D}$; initialize all trajectories with their native requests
for $r = 1, \ldots, R$ **do**
    Advance every trajectory through round r at fixed γ
    Audit the current landings Y_r and update $\mathcal{A}_r \leftarrow \mathcal{A}_{r-1} \cup Y_r$
    if $r < R$ **then**
        Compute S_i against $\mathcal{A}_{r-1} \cup Y_r$; protect the multi-quantile elites
        Build candidates from audited non-dominated anchors and their scale-controlled improving-direction variants
        Score each candidate landing by Eq. 2 against $\mathcal{A}_r$
        Set $B_r = \min\{\lceil \rho N \rceil, |\mathcal{C}_r^{\text{ret}}|\}$; draw B_r sources **uniformly**
        Greedily select their destinations by Eq. (3)
        Replace the selected sources' requests; retain their current states
    end if
end for
return the final Q designs ▷ oracle used only for scoring

Destination selection. For fixed archive strata, let $F(Y)$ denote the percentile-balanced hypervolume gained by a landing set Y.

Lemma 1 (Balanced hypervolume is submodular)**.** *For fixed archive strata and reference point, F is normalized, monotone non-decreasing, and submodular.*

The greedy guarantee requires each candidate request to have a fixed landing and ARC's predicted scores to approximate the corresponding true gains.

Assumption 1 (Source-independent landings)**.** *Each request v has a fixed realized landing $y(v)$, independent of its receiving trajectory and the other assignments made in the round.*

Assumption 2 (α-accurate scores)**.** *At every greedy step, the predicted score of every remaining candidate differs from its true marginal gain in F by at most α.*

Theorem 1 (Conditional destination-selection bound)**.** *Suppose ARC greedily selects B requests from $\mathcal{C}$, where $|\mathcal{C}| \geq B$, and each selection is within δ of the largest predicted score. Under Assumptions 1 and 2, let Y^{ARC} be the realized selected landings and $Y^\star$ the best set of at most B realized landings available from $\mathcal{C}$. Then*

$$F(Y^{\text{ARC}}) \geq \left(1 - \frac{1}{e}\right) F(Y^\star) - B(\delta + 2\alpha).$$

The factor $1 - 1/e$ is the standard greedy-submodular guarantee (Nemhauser et al., 1978). ARC scores every retained candidate, so $\delta = 0$ up to numerical precision. The result remains conditional: α is unmeasured because unselected candidates are not realized, and Assumption 1 is approximate because a landing may depend on the trajectory state and remaining sampling time. The bound applies to one round, compares ARC only with the best subset of its candidate pool, and covers destination gain rather than source loss. Complete proofs are given in Appendix B.

5 Main Results

Scope and evaluation. We compare ARC against frozen PCD, a compute-matched open-loop control, published baselines, and single-switch ablations. The paired ARC–PCD evaluation covers 48 tasks across synthetic, engineering, scientific-design, MO-NAS, and MO-RL domains (Xue et al., 2024; Tanabe & Ishibuchi, 2020; Xu et al., 2020). The all-method ranking includes the 44 tasks for which every method completed its full seed budget. Main PCD feedback comparisons use all 48 tasks; the balanced-utility and candidate-push ablations use 47, while the oracle-auditor and steering-window experiments use 45. All 48 attempted tasks pass the flow-fidelity gate (Section 5.4).

Table 1: Average rank on the 44 tasks completed by every method; lower is better. Generative methods use five seeds and Off-MOO-Bench baselines use three. Values are the mean rank ± the seed-to-seed standard error, with methods ranked independently within each seed. $\mathcal{D}(\text{best})$ is an offline-data reference, not an optimization method.

Method	HV(100th)	HV(75th)	HV(50th)
ARC (ours)	$\mathbf{4.02 \pm 0.21}$	$\mathbf{5.16 \pm 0.26}$	5.45 ± 0.27
PCD	6.41 ± 0.20	7.02 ± 0.05	6.95 ± 0.15
ParetoFlow	9.43 ± 0.26	8.07 ± 0.30	8.32 ± 0.12
Offline data ($\mathcal{D}(\text{best})$; reference)	10.41 ± 0.04	7.34 ± 0.06	$\mathbf{5.02 \pm 0.14}$
End2End	5.80 ± 0.16	6.41 ± 0.19	6.23 ± 0.20
End2End–GradNorm	9.18 ± 0.03	9.23 ± 0.29	9.52 ± 0.16
End2End–PcGrad	6.27 ± 0.14	6.98 ± 0.25	7.52 ± 0.18
MultiHead	6.86 ± 0.08	6.98 ± 0.25	6.84 ± 0.17
MultiHead–GradNorm	11.25 ± 0.27	10.95 ± 0.23	11.82 ± 0.53
MultiHead–PcGrad	9.09 ± 0.28	9.16 ± 0.25	9.39 ± 0.42
MultipleModels	6.45 ± 0.29	6.55 ± 0.15	6.11 ± 0.32
MultipleModels–IOM	7.14 ± 0.21	7.30 ± 0.32	7.82 ± 0.23
MultipleModels–ICT	8.59 ± 0.13	8.43 ± 0.28	8.18 ± 0.32

Controls and settings. Paired ARC arms share checkpoint, initial noise, native requests, $Q = 256$ outputs, schedule, and backbone-evaluation count. Generative methods use five seeds; Off-MOO-Bench baselines use three. All use the same clip-then-score path. Full-set HV measures frontier performance; HV of the best 75% or 50% of designs, measures the quality of the returned set beyond its few best points. $\mathcal{D}(\text{best})$ is the non-dominated offline-data reference; surrogate baselines use End2End, MultiHead, or MultipleModels with the reported gradient-balancing or regularization variants, and ParetoFlow is a separate flow generator (Shrestha et al., 2026; Yuan et al., 2025; Chen et al., 2018; Yu et al., 2020). Controller hyperparameters are fixed globally rather than tuned per task; auditors and landing maps are fitted per run, which is distinct from hyperparameter selection (Appendix C).

5.1 Main comparison

Table 1 places the mechanism study in the wider benchmark. ARC achieves the best average rank among evaluated methods at the 100th and 75th percentiles. At the 50th percentile, the non-dominated offline-data reference ranks first and ARC ranks second; the visual separation in the table clarifies that this reference is not a method trained or optimized at test time.

Compared with PCD, ARC improves full-set hypervolume on 36 tasks, with a mean gain of $+4.23\%$, a median gain of $+0.13\%$, and a 10% trimmed-mean gain of $+2.54\%$. These aggregate values provide context rather than the central claim: the effect is heterogeneous and modest on many problems. The clearest frontier result is the coherent engineering family, where ARC improves all 15 tasks with a $+8.13\%$ mean gain (Table 5). The mechanism tests below examine what produces this gain and what instead improves the convergence of the returned set.

5.2 Mechanism and ablations

We first test whether ARC benefits from feedback itself rather than from additional inference compute. Table 2 compares ARC with PCD and with a compute-matched variant that follows the same sampling schedule without reallocating trajectories.

Panel A shows that the no-reallocation variant performs similarly to PCD, whereas ARC improves mean hypervolume at all three percentiles. Panel B directly isolates the effect of feedback: ARC improves hypervolume by $+4.49\%$, $+4.86\%$, and $+3.83\%$ at the 100th, 75th, and 50th percentiles, respectively. These improvements are statistically significant ($p = 0.0025$, 0.0017, and 0.0036), with ARC winning on 30/48, 32/48, and 35/48 tasks. Since the two arms differ only in whether conditioning requests are revised, the improvement is attributable to closed-loop reallocation rather than additional sampling compute.

The effects in Panel B are obtained by first computing the relative difference between the two methods on each task and then averaging across tasks. They may therefore differ slightly from a direct subtraction of the aggregate values in Panel A.

Having established the effect of reallocation, we next examine the decisions made within the con-

Table 2: Compute-matched mechanism comparison over 48 tasks and five seeds. Panel A reports the mean relative hypervolume of each variant with respect to PCD. Panel B directly compares ARC with the no-reallocation variant. Positive values indicate improvement over the reference method. All variants use the same checkpoint, initial noise, sampling schedule, output budget, and number of network evaluations.

Comparison	HV(100th)	HV(75th)	HV(50th)
Panel A: Performance relative to PCD			
Archive guided (ARC)	$+4.23\%$	$+4.87\%$	$+3.72\%$
Uniform random destinations	$+3.48\%$	$+0.41\%$	-5.46%
No reallocation	-0.24%	-0.04%	-0.08%
Panel B: Direct effect of closed-loop reallocation			
ARC vs. no reallocation	$+4.49\%$	$+4.86\%$	$+3.83\%$

Table 3: Effects of individual ARC components. Each entry is the mean relative hypervolume of the full method over a variant in which the indicated component is changed or removed. Positive values indicate that retaining the component improves performance.

Component	Isolating comparison	Effect
Reallocation mechanism		
Closed-loop feedback	vs. compute-matched open loop	$+4.49\%$
Archive-guided destinations (100th)	vs. uniform random destinations	$+0.78\%$
Archive-guided destinations (50th)	vs. uniform random destinations	$+15.87\%$
Contribution-based source ordering	vs. uniform random sources	-0.25%
Retained controller components		
Elite protection (100th)	vs. no elite protection	$+0.01\%$
Elite protection (50th)	vs. no elite protection	$+0.61\%$
Balanced utility (100th)	vs. top-percentile utility	$+0.05\%$
Balanced utility (50th)	vs. top-percentile utility	$+0.19\%$

troller. Table 3 isolates archive-guided destination selection, source ordering, elite protection, and the balanced utility.

Archive-guided destination selection has little effect on the extreme frontier: compared with random destinations, it changes 100th-percentile hypervolume by $+0.78\%$ ($p = 0.32$). Its benefit is instead concentrated in the better-ranked half of the returned set, where it improves 50th-percentile hypervolume (the hypervolume of the 128 best designs) by $+15.87\%$ ($p < 0.0001$).
Because percentile hypervolume is not a direct front-quality measure, we also compare the two destination rules using two front-based metrics (Table 4). The inverted generational distance (IGD) is the average distance from each point of a reference front to the nearest point of a method's front (Ishibuchi et al., 2015); lower means closer to the best-known trade-offs. The set-coverage metric (Zitzler & Thiele, 1999), or C-metric, is the fraction of one front's points that are dominated by the other front. Archive guidance lowers IGD, and its front dominates the random-destination front more often ($C = 0.36$ vs. 0.16), while the two arms occupy a similar number of front cells. The benefit therefore comes from converging on and dominating the reference front more effectively, not from occupying more regions. Overall, reallocation accounts for most of the frontier gain, while archive guidance determines how effectively the redirected computation improves the convergence of the returned set.

Additionally, Contribution-based source ordering performs slightly worse than uniform source selection (-0.25%; Table 3), so ARC samples eligible source trajectories uniformly. Elite protection and the balanced utility are retained as safeguards rather than independently validated contributions. Their largest effects occur at the 50th percentile ($+0.61\%$ and $+0.19\%$), consistent with their intended roles in preserving strong trajectories and preventing destination selection from concentrating entirely near the extreme frontier.
Appendix Tables 6 and 7 give confidence intervals, win counts, and tests. Removing reallocation is the only core ablation whose bootstrap interval excludes zero at every percentile; halving the round-level budget or removing candidate pushes produces no conclusive change in final hypervolume. The pushes do redirect a median of 31 more trajectories per run and raise auditor-measured archive-hypervolume growth by 4.67%, filling otherwise unused reallocation slots, but their effect on final hypervolume is not established.

Table 4: Front-based comparison of archive-guided and random destinations. IGD is measured to the reference front (lower is better); C is the asymmetric fraction of the other arm's front that is dominated (higher is better). Paired differences are medians of per-task differences, not differences of the column medians, with 95% intervals; p is from a two-sided paired Wilcoxon signed-rank test.

Metric	Archive	Random	Paired difference	p
IGD ↓	0.030	0.052	-0.004 $[-0.012, -0.001]$	0.010
C against the other arm ↑	0.359	0.163	$+0.136$ $[+0.051, +0.259]$	0.0001
Occupied front cells	11.70	11.00	$+0.10$ $[-0.30, +0.80]$	0.255

Table 5: ARC relative to PCD at the 100th percentile, by task family (48 paired tasks, five seeds averaged).

Family	n	Median	Mean	Wins
RE engineering	15	$+2.32\%$	$+8.13\%$	15/15
Synthetic	11	$+0.02\%$	$+6.83\%$	7/11
Scientific design	3	$+0.15\%$	-1.34%	2/3
MO-NAS	17	$+0.04\%$	$+0.66\%$	11/17
MO-RL	2	-0.50%	-0.50%	1/2

5.3 Where ARC works and where it underperforms

Table 5 shows where ARC's frontier gain occurs. Gains are strongest and most consistent on RE engineering. Synthetic tasks contain several large gains but a near-zero median, gains are small on most MO-NAS tasks, and ARC shows no consistent benefit on scientific design or MO-RL. The weaker cases coincide with limited room for useful reallocation: on some tasks the backbone reaches similar regions whatever the target, on others the archive holds too few distinct landings, and auditor rank correlation falls below the usability threshold on three tasks, including two scientific-design tasks. These are plausible limitations, not causal explanations for individual tasks (Appendix D).

5.4 Transfer to a flow-matching backbone

We test whether ARC extends beyond diffusion by integrating it with ParetoFlow (Yuan et al., 2025), while preserving its flow state, offspring construction, filtering rules, and network-evaluation budget. The diffusion frontier gain does not transfer: over 48 tasks and five seeds, the median 100th-percentile effect against compute-matched open-loop ParetoFlow is -0.11% ($p = 0.179$), while the largest median gain occurs at the 50th percentile ($+0.51\%$; 31/48 wins; $p = 0.072$). No effect is statistically significant, so the experiment establishes architectural compatibility rather than a confirmed improvement (Table 44, Appendix H). The weaker result may reflect how much of the sampling process remains steerable. ParetoFlow's direct predictor guidance operates only during the final 20% of integration, although filtering and neighboring evolution remain active throughout. We test this explanation on the diffusion backbone by restricting ARC to the final $k\%$ of sampling. Its gain remains nearly unchanged at $k = 50\%$, but its mean full-set gain falls from $+4.56\%$ to $+1.05\%$ at $k = 20\%$ and $+0.34\%$ at $k = 10\%$ (Table 13, Appendix D.7). Thus, ARC benefits from having sufficient sampling time remaining after redirection. This comparison is qualitative: reducing k also compresses ARC's feedback rounds, while ParetoFlow filters and evolves samples before direct guidance begins.

6 Summary and Limitations

This paper introduces Archive-Guided Reallocation (ARC), a closed-loop controller that redirects partial generations from a frozen conditional model without retraining, restarting, or additional backbone evaluations. On PCD, reallocation improves the frontier, while archive guidance improves convergence. Gains shrink as less sampling remains steerable, consistent with the weaker flow-backbone results. Benefits are task- and backbone-dependent, require an offline auditor and realization model, cannot exceed the backbone's reachable outcomes, and Theorem 1 rests on idealized assumptions. Future work should improve auditor calibration and backbone-adaptive feedback.

AI USE STATEMENT

In this work, we used generative AI tools to draft selected sections of the paper and to polish the writing for clarity, grammar, and organization. We did not use generative AI to generate research ideas, experimental results, statistical analyses, tables, or scientific claims; the remaining disclosure categories are not applicable to this work. The authors reviewed, revised, and verified all AI-assisted text. We take full responsibility for the final content of this work, including all text produced with the aid of generative AI.

ETHICS STATEMENT

This work studies offline multi-objective optimization on public benchmark suites. It involves no human subjects, no personally identifiable information, and no newly collected data. The optimization procedure is a training-free controller placed around an existing generative model and introduces no capability beyond that of the backbone it wraps. We see no direct path to harmful application beyond the general dual-use considerations of design optimization, and the benchmarks used are released for research under their original licenses.

REPRODUCIBILITY STATEMENT

The controller, backbone interface, and analysis scripts will be released upon acceptance. Appendix A specifies the reallocation rule, candidate construction, and realizability checks. Appendix C documents the environment, shared checkpoints, fixed hyperparameters, seeds, task coverage, and measured controller overhead. Appendix B provides the assumptions and complete proofs. Experimental arms share checkpoints and differ by a single switch, statistical tests are paired across tasks, and exclusions based on fidelity or coverage criteria are reported with their reasons.

REFERENCES

Christof Angermueller, David Belanger, Andreea Gane, Zelda Mariet, David Dohan, Kevin Murphy, Lucy Colwell, and D. Sculley. Population-based black-box optimization for biological sequence design. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*. PMLR, 2020.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pp. 794–803. PMLR, 2018.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.

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.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified distance calculation in generational distance and inverted generational distance. In *International conference on evolutionary multi-criterion optimization*, pp. 110–125. Springer, 2015.

Moksh Jain, Emmanuel Bengio, Alex-Hernandez Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ekbote, Jie Fu, Tianyu Zhang, Maksym Korablyov, and Yoshua Bengio. Multi-objective gflownets. In *International Conference on Machine Learning*, 2023.

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

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Kaisa Miettinen. *Nonlinear Multiobjective Optimization*, volume 12. Kluwer Academic Publishers, 1999.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14:265–294, 1978.

Han Qi, Yi Su, Aviral Kumar, and Sergey Levine. Data-driven offline decision-making via invariant representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 13226–13237, 2022.

Jatan Shrestha, Santeri Heiskanen, Kari Hepola, Severi Rissanen, Pekka Jääskeläinen, and Joni Pajarinen. Pareto-conditioned diffusion models for offline multi-objective optimization. *arXiv preprint arXiv:2602.00737*, 2026.

Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020.

Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative objective models for effective offline model-based optimization. In *International Conference on Machine Learning*, pp. 10358–10368. PMLR, 2021.

Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-bench: Benchmarks for data-driven offline model-based optimization. In *International Conference on Machine Learning*, pp. 21658–21676. PMLR, 2022.

Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *International conference on machine learning*, pp. 10607–10616. PMLR, 2020.

Ke Xue, Rong-Xi Tan, Xiaobin Huang, and Chao Qian. Offline multi-objective optimization. *arXiv preprint arXiv:2406.03722*, 2024.

Sihyun Yu, Sungsoo Ahn, Le Song, and Jinwoo Shin. Roma: Robust model adaptation for offline model-based optimization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4619–4631, 2021.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Ye Yuan, Can Chen, Christopher Pal, and Xue Liu. Paretoflow: Guided flows in multi-objective optimization. In *International Conference on Learning Representations*, volume 2025, pp. 72594–72620, 2025.

Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

A ARC IMPLEMENTATION DETAILS

A.1 SCALE-CONTROLLED CANDIDATE EXPANSION

At feedback round r, let

$$\mathcal{F}_r = \mathrm{ND}(\{\widehat{y} : (c, \widehat{y}, u) \in \mathcal{A}_r\})$$

be the non-dominated audited archive landings. ARC samples without replacement $K = \min\{K_{\max}, |\mathcal{F}_r|\}$ anchors, where $K_{\max} = 64$, and writes $\mathcal{P}_r = \{b_1, \ldots, b_K\}$. It retains the anchors and generates two independent improving-direction variants of each. The scale of objective j is

$$s_j = \eta \max\left\{ \frac{1}{K} \sum_{k=1}^{K} |b_{k,j}|, 10^{-6} \right\}, \qquad \eta = 0.25, \tag{4}$$

and, for $\ell \in \{1, 2\}$,

$$v_{k,j}^{(\ell)} = b_{k,j} - |z_{k,j}^{(\ell)}| s_j, \qquad z_{k,j}^{(\ell)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1). \tag{5}$$

All objectives are minimized, so subtraction moves each coordinate in the improving direction. The resulting pool is

$$\mathcal{C}_r = \mathcal{P}_r \cup \{v_k^{(1)} : b_k \in \mathcal{P}_r\} \cup \{v_k^{(2)} : b_k \in \mathcal{P}_r\},$$

and contains at most $3K_{\max} = 192$ requests. The displacement is scale-controlled rather than deterministically bounded because the half-normal distribution has unbounded support. ARC never combines coordinatewise minima from different designs, which could manufacture an unattained utopian point.

The variants are a budget-utilization device. A front with fewer than 64 anchors can exhaust the front-only pool before the desired reallocation budget is used. Across 238 paired runs, pushes increase the median number of redirected trajectories from 345.5 to 384 (paired difference $+31$, $p = 5 \times 10^{-29}$) and auditor-measured archive-HV growth from 35.34% to 44.18% (paired difference $+4.67\%$, $p = 3 \times 10^{-18}$). These are internal outcomes; Table 6 shows that the oracle-scored final-HV effect is unresolved.

A.2 REQUEST-TO-LANDING REALIZATION MODEL

A candidate v is an objective-space request, whereas the auditor evaluates a design. ARC therefore learns where requests tend to land from the archived pairs $(c_i, \widehat{y}_i)$. Let $k(v)$ be the reference-direction cell containing v. Within cell k, ARC fits by ridge regression

$$\widetilde{y}_k(v) = A_k v + b_k^{\mathrm{real}}, \tag{6}$$

and fits the global fallback

$$\widetilde{y}_{\mathrm{glob}}(v) = A_{\mathrm{glob}} v + b_{\mathrm{glob}} \tag{7}$$

using all audited pairs. The intercept b_k^{real} is distinct from the archive anchor b_k above. A local model is used only if its cell contains at least $n_{\min} = 10$ pairs, its condition number is at most 10^4, and its normalized held-out residual is at most one:

$$\widetilde{y}(v) = \begin{cases} A_{k(v)} v + b_{k(v)}^{\mathrm{real}}, & n_{k(v)} \geq 10,\ \mathrm{cond}_{k(v)} \leq 10^4,\ \rho_{k(v)}/\sigma_{k(v)} \leq 1, \\ A_{\mathrm{glob}} v + b_{\mathrm{glob}}, & \text{otherwise.} \end{cases} \tag{8}$$

Here ρ_k is held-out prediction error and σ_k is the objective scale. At the first feedback boundary, the just-audited live trajectories supply request–landing pairs for the global fit; later rounds accumulate more pairs and may enable cell-specific fits.

ARC discards candidates whose predicted request–landing discrepancy fails the implementation's realizability check. For a retained request, the uncertainty term in Eq. 2 is the residual of the map actually used,

$$u_{\mathrm{real}}(v) = \begin{cases} \rho_{k(v)}, & \text{if the local model is used,} \\ \rho_{\mathrm{glob}}, & \text{if the global fallback is used.} \end{cases} \tag{9}$$

Thus hypervolume is evaluated at $\widetilde{y}(v)$, not at the optimistic raw request v, and unreliable realization fits receive the residual penalty.

B THEORY

B.1 MATHEMATICAL PRELIMINARIES

Lemma 2 (Basic properties of hypervolume)**.** *Fix a reference point z and a finite $Y \subset \mathbb{R}^m$ with $y \leq z$ for every $y \in Y$. Then: (i) $\cup_{y \in Y}[y, z]$ is compact, so $\mathrm{HV}(Y)$ is finite and $\mathrm{HV}(\varnothing) = 0$; (ii) $Y \subseteq Y'$ implies $\mathrm{HV}(Y) \leq \mathrm{HV}(Y')$; and (iii), for $v \leq z$,*

$$\mathrm{HV}(Y \cup \{v\}) - \mathrm{HV}(Y) = \lambda\left([v, z] \setminus \bigcup_{y \in Y}[y, z]\right).$$

Proof. A finite union of closed bounded boxes is compact, Borel measurable, and has finite Lebesgue measure. Set inclusion gives inclusion of the corresponding unions of boxes, proving monotonicity. For (iii), let $R = \cup_{y \in Y}[y, z]$ and use the disjoint decomposition $R \cup [v, z] = R \cup ([v, z] \setminus R)$. □

Definition 1 (Normalized monotone submodular)**.** A function $F : 2^\Omega \to \mathbb{R}$ on a finite ground set is normalized if $F(\varnothing) = 0$, monotone if $F(Y) \leq F(Y')$ for $Y \subseteq Y'$, and submodular if, for $Y \subseteq Y'$ and $v \notin Y'$,

$$F(Y \cup \{v\}) - F(Y) \geq F(Y' \cup \{v\}) - F(Y').$$

Lemma 3 (Nonnegative closure)**.** *If $F_1, \ldots, F_n$ are normalized, monotone, and submodular, $\omega_i \geq 0$, and $\kappa_i > 0$, then $F = \sum_i (\omega_i/\kappa_i) F_i$ has the same three properties.*

Proof. Each defining equality or inequality is preserved by nonnegative scaling and summation. Moreover,

$$F(\varnothing) = \sum_i (\omega_i/\kappa_i) F_i(\varnothing) = 0.$$

□

Lemma 4 (Approximate-greedy recursion)**.** *Let $B \geq 1$, let $F : 2^\Omega \to \mathbb{R}$ be normalized, monotone, and submodular, and let $Y^\star$ maximize F over sets of size at most B. Starting from $Y_0 = \varnothing$, add one element at each step. If the selected element's true marginal gain is within η of the best remaining marginal, then, for $t = 1, \ldots, B$,*

$$F(Y^\star) - F(Y_t) \leq \left(1 - \frac{1}{B}\right)[F(Y^\star) - F(Y_{t-1})] + \eta,$$

and consequently

$$F(Y_B) \geq (1 - 1/e) F(Y^\star) - B\eta.$$

Proof. Write $\Delta(v \mid Y) = F(Y \cup \{v\}) - F(Y)$. By monotonicity and submodularity,

$$F(Y^\star) - F(Y_{t-1}) \leq \sum_{v \in Y^\star \setminus Y_{t-1}} \Delta(v \mid Y_{t-1}).$$

The sum has at most B terms, so one available element has marginal gain at least $[F(Y^\star) - F(Y_{t-1})]/B$. The selected element therefore gains at least this quantity minus η, which gives the stated recursion. Writing $G_t = F(Y^\star) - F(Y_t)$ and unrolling yields

$$G_B \leq \left(1 - \frac{1}{B}\right)^B G_0 + \eta \sum_{j=0}^{B-1} \left(1 - \frac{1}{B}\right)^j.$$

Normalization gives $G_0 = F(Y^\star)$. Applying $(1 - 1/B)^B \leq e^{-1}$ and bounding the geometric sum by B proves the result. □

B.2 PROOFS OF THE MAIN RESULTS

Proof of Lemma 1. For a fixed stratum q, define

$$F_q(Y) = \mathrm{HV}(\mathcal{A}^{(q)} \cup Y) - \mathrm{HV}(\mathcal{A}^{(q)}).$$

It is normalized and monotone by Lemma 2. For $Y \subseteq Y'$ and $v \notin Y'$, define

$$R = \bigcup_{y \in \mathcal{A}^{(q)} \cup Y} [y, z], \qquad R' = \bigcup_{y \in \mathcal{A}^{(q)} \cup Y'} [y, z].$$

Since $R \subseteq R'$,

$$F_q(Y \cup \{v\}) - F_q(Y) = \lambda([v, z] \setminus R) \geq \lambda([v, z] \setminus R') = F_q(Y' \cup \{v\}) - F_q(Y').$$

Thus every F_q is submodular. Lemma 3 applies because $\omega_q \geq 0$ and the frozen denominators $\mathrm{HV}(\mathcal{A}^{(q)}) + \varepsilon$ are positive constants. □

Proof of Theorem 1. At step t, let $\widetilde{Y}_{t-1}$ be ARC's predicted working set and Y_{t-1} the corresponding realized landing set. Define

$$\Delta_t^{\mathrm{score}}(v) = a(v; \widetilde{Y}_{t-1}), \qquad \Delta_t^{\mathrm{true}}(v) = \Delta F(y(v) \mid Y_{t-1}).$$

By Assumption 2, their difference is at most α. Let v_t be ARC's selected request and $v_t^\star$ the remaining request with the largest true marginal gain. Then

$$\begin{aligned} \Delta_t^{\mathrm{true}}(v_t) &\geq \Delta_t^{\mathrm{score}}(v_t) - \alpha \\ &\geq \Delta_t^{\mathrm{score}}(v_t^\star) - \delta - \alpha \\ &\geq \Delta_t^{\mathrm{true}}(v_t^\star) - \delta - 2\alpha. \end{aligned}$$

The first and third inequalities use Assumption 2, while the second uses the δ-approximate score maximizer. Hence every step is approximate greedy with $\eta = \delta + 2\alpha$.

Assumption 1 makes $y(v)$ a fixed landing for each request. Therefore, the induced request-set objective

$$\overline{F}(S) = F(\{y(v) : v \in S\})$$

is well defined and inherits normalization, monotonicity, and submodularity from F. Applying Lemma 4 to $\overline{F}$ gives

$$F(Y^{\mathrm{ARC}}) \geq (1 - 1/e)F(Y^\star) - B(\delta + 2\alpha).$$

□

Proof of Proposition 1. Add and subtract $\mathrm{HV}(L_{-J})$:

$$\begin{aligned} \mathrm{HV}(L_{-J} \cup Y_J) - \mathrm{HV}(L) &= [\mathrm{HV}(L_{-J} \cup Y_J) - \mathrm{HV}(L_{-J})] \\ &\quad - [\mathrm{HV}(L) - \mathrm{HV}(L_{-J})] \\ &= g(Y_J \mid L_{-J}) - \ell(J \mid L). \end{aligned}$$

Its sign is positive exactly when $g(Y_J \mid L_{-J}) > \ell(J \mid L)$. □

B.3 ADDITIONAL THEORETICAL DISCUSSION

Proposition 1 clarifies the roles of ARC's two decisions. Source selection determines the potential loss from abandoning the original trajectories, represented by $\ell(J \mid L)$. Elite protection may reduce this loss but cannot guarantee $\ell(J \mid L) = 0$. Destination selection determines the gain produced by the replacement landings, $g(Y_J \mid L_{-J})$. This gain is observable, but the loss cannot be measured directly because the no-redirection outcomes of redirected trajectories are counterfactual.

Theorem 1 and Proposition 1 answer different questions. The theorem compares ARC's selected destinations with the best destinations in the same candidate pool under source-independent landings, fixed archive strata, and bounded score error. The proposition instead gives the exact condition under which redirection improves final hypervolume. Therefore, good destination selection alone cannot guarantee improvement: its gain must exceed the value sacrificed by redirecting the sources.

The observed 50th-percentile benefit of archive guidance supports its role in improving destination quality, but does not validate Assumption 1 or the unmeasured error bound α.

Lemma 1 applies only within a single round. With fixed strata, adding landings to an already represented region provides diminishing marginal value. It does not imply convergence across rounds because the archive, candidate pool, realization model, and live population all change. The round-budget ablation is consistent with this intuition but is not a theoretical consequence of the lemma.

C Experimental protocols and benchmark coverage

C.1 Environment, backbone, and controller settings

Experiments use Python 3.8.18, PyTorch 2.0.1+cu118 (CUDA 11.8, cuDNN 8700), NumPy 1.22.3, SciPy 1.10.1, pymoo 0.5.0, and pygmo 2.19.5. Runs use two NVIDIA TITAN RTX GPUs (24 GB each), a 16-core host, 62 GB RAM, and driver 580.126.20. Hypervolume is computed with pygmo's exact routine; leave-one-out hypervolume contributions use its exclusive-contribution pass.

The primary backbone is PCD (Shrestha et al., 2026). ARC and PCD share one trained diffusion checkpoint for each task and seed. ARC uses $Q = 256$ live trajectories and 1024 denoising steps divided into $R = 8$ rounds. The following hyperparameters are fixed once and never tuned per task: classifier-free guidance $\gamma = 2.5$, reallocation rate $\rho = 0.25$, protected-elite fraction 0.20, balanced weights $(0.50, 0.30, 0.20)$, 16 realization cells, ridge coefficient $\nu = 10^{-2}$, uncertainty weight $\lambda_u = 0.5$ and push fraction 0.25. Paired controller arms share checkpoint, seed, initial noise, native targets, output budget, round boundaries, and total denoising-network evaluations. Redirection changes only the conditioning request and preserves the current diffusion state.

The auditor contains three bootstrap-resampled MLPs per objective, trained for each task and seed using only the offline dataset. It is never trained on true objectives queried during generation. The oracle is evaluated only after the final Q designs are returned. The experiment collection contains 245 shared diffusion checkpoints and 240 auditor ensembles, together with 245 flow-matching models and 700 proxies. Every ablation changes one switch on the same checkpoints, so controller comparisons are not confounded by retraining. The compute-matched claim concerns network-evaluation count, not equal wall-clock time: auditing, archive maintenance, realization fitting, and hypervolume scoring add CPU/GPU memory and elapsed-time overhead.

C.2 Benchmarks and Evaluation Protocol

The paired ARC-PCD block contains 48 Off-MOO-Bench tasks (Xue et al., 2024): 11 synthetic functions; two MORL policy-design tasks from PGMORL (Xu et al., 2020); 15 RE engineering problems (Tanabe & Ishibuchi, 2020); three scientific-design tasks; and 17 MO-NAS tasks. Combinatorial tasks requiring a different diffusion/decoding procedure are outside scope. Every method returns 256 designs. We report hypervolume at the 100th percentile and after retaining the best 75% or 50% by summed normalized objective, using the benchmark reference point and a common clip-then-score path. Hypervolume measures the volume of objective space dominated by a solution set relative to a reference point.

Generative methods, ARC arms, and $\mathcal{D}(\text{best})$ use five seeds $(1000, 2000, 3000, 4000, 5000)$; Off-MOO-Bench baselines use their shipped budget of three $(1000, 2000, 3000)$. Comparison tables are restricted to tasks on which every included method completes its full budget. Failures and exclusions are reported explicitly; no missing result is silently dropped or imputed.

Results are first averaged across seeds within each task. Statistical significance is then assessed across tasks using a two-sided paired Wilcoxon signed-rank test. Bootstrap confidence intervals are computed by resampling tasks, and the corresponding p-values are reported.

C.3 Flow-backbone determinism

We reproduced ParetoFlow's original sampling procedure before integrating ARC. With ARC disabled, our implementation was required to match the reference implementation exactly in both the returned designs and hypervolume at all three percentiles. This fidelity check succeeded on all of 48

tasks, which form the flow evaluation set. Exact reproduction required preserving two implementation details: constructing the same temporary network to maintain PyTorch's random-number state, and applying design normalization before objective normalization.

C.4 Baselines and omissions

$\mathcal{D}(\text{best})$ is the non-dominated slice of the offline dataset. PCD is the frozen diffusion backbone. ParetoFlow is a guided flow-matching generator with multi-objective predictor guidance, local filtering, and neighboring evolution (Yuan et al., 2025). End2End predicts all objectives with one model; MultiHead shares a representation and uses objective-specific heads; MultipleModels trains one predictor per objective. GradNorm and PcGrad are multi-task gradient-balancing variants, while IOM and ICT are offline model-based-optimization regularizers, as implemented by Off-MOO-Bench. The compute-matched open-loop control uses ARC's checkpoint and schedule but never rewrites requests. Random-destination, no-protection, top-percentile-only, front-only, and reduced-round arms are one-switch ablations.

Three incomplete variants are not included in the common comparison: MultipleModels+RoMA failed on all but one task; MultipleModels+COMs and MultipleModels+Tri-Mentoring were stopped before results were inspected because their 50 particle-gradient steps per candidate accounted for roughly 43% of the remaining baseline compute. Their absence is an omission, not an inferred performance result.

D Further component ablations

This appendix examines individual ARC components, distinguishes consistent effects from unresolved ones, and characterizes where the overall improvement occurs. Unless stated otherwise, each comparison uses 48 tasks and five seeds, with seeds averaged within each task.

D.1 Core controller components

Table 6 reports single-switch ablations of reallocation, the round-level reallocation budget, and candidate pushes. Each entry is the relative hypervolume of the ablated method compared with full ARC; therefore, negative values indicate that removing the component hurts performance.

As shown in Table 6, removing reallocation is the only ablation whose confidence interval excludes zero at all three percentiles. This provides the clearest component-level evidence that ARC's benefit comes from closing the inference loop. Halving the round-level budget and removing candidate pushes produce much smaller effects whose intervals include zero. The push mechanism therefore improves candidate availability and budget utilization, but a dedicated single-switch test finds no independent effect on final oracle-scored hypervolume (Table 11).

A separate round-count ablation gives mean relative full-set hypervolume changes of $-.0042$ and $-.0004$ with three and six feedback rounds, respectively, whereas eliminating effective feedback gives $-.0265$. These results suggest that obtaining some feedback is more important than repeatedly reallocating after useful coverage has already been acquired.

Table 7 evaluates the two retained safeguards: elite protection and the percentile-balanced utility. Positive values indicate that retaining the component improves performance.

Both components exhibit their largest effects at the 50th percentile in Table 7, consistent with their intended role in protecting the quality of the broader returned set. Elite protection improves mean hypervolume by 0.609% and wins on 30 of 48 tasks, while the balanced utility provides a smaller improvement of 0.186%. These results suggest that both components support broader-set quality, although their effects are modest relative to the improvement produced by closed-loop reallocation.

D.2 Closed-loop feedback versus using the auditor once

ARC uses an objective auditor that native PCD does not. To determine whether ARC benefits merely from having this auditor, we construct an open-loop control that generates twice as many

PCD samples and uses the same auditor only once, at the end, to select the final Q designs. Unlike ARC, this control never feeds the audited information back into generation.

Table 8 shows that applying the auditor only after sampling does not improve mean hypervolume, despite using twice the sampling budget. ARC, which uses the auditor repeatedly to revise inference allocation, obtains a $+4.23\%$ mean improvement. This comparison indicates that the main benefit comes from feeding intermediate audit information back into generation rather than merely adding a surrogate or post-selection stage.

D.3 Distribution of gains across tasks

Because ARC's average improvement may be influenced by a small number of large wins, Table 9 summarizes the complete distribution of task-level gains relative to PCD.

As Table 9 shows, ARC improves 36 of 48 tasks, but the mean gain is larger than the median because the distribution has a positive right tail. Removing VLMOP1, the largest improvement, still leaves 35 wins among 47 tasks and a $+2.18\%$ trimmed-mean gain. Large improvements occur on RE42, RE36, RE37, RE41, and ZDT2, whereas losses occur on molecule, VLMOP2, and C10MOP4. These examples characterize the observed variation and should not be interpreted as a universal rule about task type.

D.4 Diversity and auditor cost

Table 10 examines whether ARC's hypervolume improvement is accompanied by a loss of diversity. The comparison is restricted to the 16 tasks shared by the corresponding diversity evaluation.

Table 10 shows that ARC approximately preserves PCD's spread and objective-space extent while slightly improving $\mathrm{IGD}_{\mathrm{offline}}$. Its hypervolume gain is therefore not accompanied by an obvious collapse of the returned set in this 16-task evaluation. ParetoFlow returns a larger fraction of non-dominated designs but has worse Δ-spread and offline-front coverage in the same comparison.

ARC is not surrogate-free: its three-member bootstrap ensemble requires additional training and inference relative to native PCD. Across the 48 tasks, the auditor's held-out rank correlation has a median of .995, while RE31, molecule, and IN1KMOP2 fall below .5. Table 8 demonstrates that using the auditor only for final selection does not explain ARC's gain. Nevertheless, the uniform marginal-score error α required by Theorem 1 is not measured empirically, so the theoretical bound should be interpreted as conditional rather than as a certified numerical guarantee.

Table 6: Core single-switch ablations over 48 tasks and five seeds. Entries report the relative hypervolume of the ablated arm against full ARC, with 95% bootstrap confidence intervals over tasks. Negative values favor ARC.

Ablation	HV(100th)	HV(75th)	HV(50th)
Reallocation removed	$-.0314[-.049, -.016]$	$-.0177[-.029, -.007]$	$-.0088[-.017, -.001]$
Half round budget	$-.0028[-.008, +.002]$	$-.0015[-.005, +.002]$	$-.0014[-.005, +.002]$
Candidate pushes removed	$-.0025[-.008, +.003]$	$+.0010[-.004, +.006]$	$+.0008[-.003, +.005]$

Table 7: Elite-protection and balanced-utility ablations. Positive effects favor retaining the component. W/L/T gives the number of task-level wins, losses, and ties.

Component	pct	n	mean	median	W/L/T
Elite protection	100th	48	$+.006\%$	$-.002\%$	22/26/0
	75th	48	$+.089\%$	$-.001\%$	23/25/0
	50th	48	$+.609\%$	$+.143\%$	30/18/0
Balanced utility	100th	47	$+.047\%$	$+.000\%$	15/11/21
	75th	47	$-.081\%$	$+.000\%$	8/18/21
	50th	47	$+.186\%$	$+.000\%$	18/9/20

Table 8: Auditor-at-the-end control over 48 tasks and five seeds, measured relative to PCD at the 100th percentile. Both methods use the same objective auditor, but only ARC feeds its predictions back into sampling.

Arm	Mean relative HV	Median relative HV	Wins/tasks
PCD + auditor, open loop, $2\times$ budget	$-.04\%$	$+.09\%$	34/48
ARC, closed loop	$+4.23\%$	$+.13\%$	36/48

Table 9: Distribution of task-level relative full-set-hypervolume gains over PCD. Results are first averaged across five seeds within each task and then summarized across tasks.

Set	n	p05	p25	median	p75	p95	mean	10% trimmed
All tasks	48	-1.90%	$-.00\%$	$+.13\%$	$+4.65\%$	$+24.43\%$	$+4.23\%$	$+2.54\%$
Without VLMOP1	47	-1.94%	$-.01\%$	$+.11\%$	$+3.43\%$	$+19.30\%$	$+3.48\%$	$+2.18\%$

Table 10: Diversity indicators, reported as medians over 16 shared tasks. $\mathrm{IGD}_{\mathrm{offline}}$ measures the distance from each point on the offline Pareto front to its nearest returned point. Arrows indicate the preferred direction.

Method	$\mathrm{IGD}_{\mathrm{offline}}\downarrow$	Δ-spread$\downarrow$	ND fraction$\uparrow$	Extent$\uparrow$
ARC	3.4971	.630	.076	.793
PCD	3.5125	.630	.087	.795
ParetoFlow	3.5830	1.574	.255	.734

D.5 CANDIDATE PUSHES

This ablation examines the scale-controlled pushes used to expand ARC's destination pool on the PCD backbone. In the altered arm, candidates are drawn only from the audited non-dominated front; the full ARC arm additionally includes two improving-direction variants of each sampled anchor. The two arms are otherwise identical. Table 11 reports the effect of retaining these pushes on final hypervolume.

As shown in Table 11, the pushes do not produce a statistically reliable change in final oracle-scored hypervolume at any percentile. They nevertheless affect ARC's internal budget utilization: by expanding a small archive front with nearby candidate destinations, they allow more trajectories to be redirected and increase auditor-measured archive-hypervolume growth (Appendix A.1). We therefore interpret the pushes as a candidate-pool expansion mechanism rather than an independently validated source of final performance improvement.

D.6 IS THE AUDITOR THE BOTTLENECK?

We examine whether errors in the learned auditor limit ARC's performance. If the auditor frequently misranks candidate destinations, improvements to the destination-selection mechanism may have little effect. We test this possibility by replacing the learned auditor with the true objectives during reallocation. This oracle-auditor variant is not a valid offline method because it queries the oracle during generation; it is used only to measure how ARC would perform without auditor error. Table 12 compares the oracle and learned auditors.

Table 12 provides no evidence that replacing the learned auditor with true objectives improves ARC. The mean and median effects remain close to zero, and the oracle-auditor arm wins on fewer than half of the tasks at every percentile. None of the paired tests is statistically significant. Under this experimental protocol, auditor error is therefore unlikely to be ARC's primary performance bottleneck, and the weak effects of the other controller components cannot be explained solely by inaccuracies in the learned auditor. The next subsection examines a different limitation: how much sampling time remains after a trajectory is redirected.

D.7 WHAT BOUNDS THE GAIN: THE STEERABLE FRACTION

The weaker result on ParetoFlow may be explained by when its conditioning signal becomes active. ParetoFlow applies direct multi-objective predictor guidance only during the final 20% of integration, although its weight-dependent filtering and neighboring-evolution operations run throughout sampling. ARC's revised conditioning requests therefore have a shorter period in which to influence the generated designs.

We test whether this shorter steering window limits ARC by restricting when it may intervene on the PCD backbone. Let k denote the percentage of the sampling trajectory during which ARC is active. The first $(100 - k)\%$ of sampling proceeds open loop, without auditing or reallocation, and ARC's eight feedback rounds are placed within the final $k\%$. Thus, $k = 100\%$ corresponds to the standard ARC procedure, while $k = 20\%$ allows ARC to act only during the final fifth of sampling. Table 13 reports the results.

Table 13 shows that ARC's gain decreases when less of the sampling trajectory remains available for steering. Allowing ARC to operate during the final 50% produces results close to those obtained with the full trajectory. Restricting it to the final 20%, however, reduces the mean gain to approximately 1%, and restricting it to the final 10% reduces the gain below 0.5% at every percentile. The association between the available steering window and ARC's gain is positive at all three percentiles ($\rho = +0.80$, one-sided $p \leq 0.0009$).

These results support the hypothesis that reallocation becomes less effective when little sampling time remains after a trajectory is redirected. They also provide a plausible explanation for ARC's weaker effect on ParetoFlow, whose direct predictor guidance is active only near the end of integration.

The comparison is not exact, however. When k decreases, ARC's eight rounds are placed closer together, leaving less time for each redirected trajectory to respond before the next feedback round.

Table 11: Effect of retaining candidate pushes on PCD, paired per task over five seeds. Positive values favour retaining the pushes. The reported p-values are from two-sided paired Wilcoxon signed-rank tests across tasks.

Percentile	n	Mean	Median	W/L	p
100th	47	$+0.27\%$	-0.00%	23/24	0.741
75th	47	$+0.62\%$	-0.00%	23/24	0.434
50th	47	$+0.09\%$	$+0.01\%$	25/22	0.358

Table 12: Oracle auditor versus the learned auditor on PCD over the 45 tasks for which both arms completed all five seeds. Effects are the relative hypervolume of the oracle-auditor arm over the learned-auditor arm. Positive values favour the oracle auditor, whereas negative values favour the learned auditor. The reported p-values are from two-sided paired Wilcoxon signed-rank tests across tasks.

Percentile	n	Mean	Median	W/L	p
100th	45	-0.46%	-0.01%	22/23	0.721
75th	45	-0.40%	-0.00%	22/23	0.814
50th	45	$+0.37\%$	-0.06%	20/25	0.973

Table 13: Effect of the available steering window on PCD over the 45 tasks for which every arm completed all five seeds. Each row reports the mean relative hypervolume of ARC over its compute-matched open-loop control, together with task-level wins and losses and a two-sided paired Wilcoxon p-value. Positive mean values favour ARC. The trend row reports the median task-level Spearman correlation between k and ARC's gain, with a one-sided Wilcoxon test for a positive association. This is a separate experiment, so the $k = 100\%$ result is computed on this 45-task subset and differs slightly from the main result.

	HV(100th)			HV(75th)			HV(50th)		
k	Mean	W/L	p	Mean	W/L	p	Mean	W/L	p
100%	$+4.56\%$	29/16	0.004	$+4.92\%$	31/14	0.002	$+3.85\%$	34/11	0.002
50%	$+4.17\%$	33/12	< 0.001	$+4.89\%$	33/12	< 0.001	$+3.89\%$	31/14	0.002
20%	$+1.05\%$	28/16	0.019	$+1.13\%$	26/18	0.097	$+1.16\%$	28/16	0.027
10%	$+0.34\%$	28/14	0.055	$+0.38\%$	25/19	0.599	$+0.48\%$	28/16	0.065
Trend	$\rho = +0.80$, $p = 0.0001$			$\rho = +0.80$, $p = 0.0009$			$\rho = +0.80$, $p = 0.0004$		

Moreover, ParetoFlow continues to use weight-dependent filtering and neighboring evolution before its direct predictor guidance becomes active. The experiment therefore supports a qualitative conclusion-that inference-time reallocation depends on the amount of steerable generation remaining-rather than a precise prediction of ARC's effect on ParetoFlow.

E SECOND-BACKBONE EVALUATION

The primary study wraps PCD, so a second architecture is needed to test whether the observed mechanism is diffusion-specific. ParetoFlow instead uses flow matching, multi-objective predictor guidance, reference-direction weights, local hypercone filtering, and neighboring evolution (Yuan et al., 2025). ARC is inserted only at feedback boundaries: flow state and ParetoFlow's native offspring, repair, filtering, and neighbor selection are preserved, while selected conditioning/reference requests are rewritten. The controller-off implementation is required to reproduce stock ParetoFlow exactly before a task is admitted.

All 48 attempted tasks pass this fidelity check. Comparisons use five seeds and the same percentile-HV protocol, with matched checkpoints, initial states, integration schedules, batch construction, and network-evaluation budgets. This experiment tests whether ARC can be integrated with a second architecture; it is not an independent benchmark suite. Complete positive, null, and negative results are reported in Appendix H.

F RELATED WORK

Offline multi-objective optimization. Offline MOO optimizes multiple objectives from a fixed evaluated dataset and cannot query the oracle during search (Xue et al., 2024). Its surrogate methods extend offline model-based optimization, where aggressive search can exploit predictor error (Trabucco et al., 2021; 2022). Off-MOO-Bench organizes shared-model, multi-head, and per-objective predictors with multi-task and conservative variants. ARC also uses an offline auditor, but uses it to allocate a frozen generator's remaining inference rather than to run a separate design-space optimizer.

Conditional generation for optimization. Generative approaches learn a design distribution conditioned or guided by desired reward/objective information. Diffusion models have been used for offline black-box optimization (Krishnamoorthy et al., 2023), and multi-objective GFlowNets target diverse trade-offs (Jain et al., 2023). PCD conditions a diffusion model on reference-direction objective targets and uses multi-objective reweighting (Shrestha et al., 2026). These approaches broaden the set of generated designs, but their target allocation is fixed before each sampling trajectory begins.

Diffusion and flow backbones. Denoising diffusion models iteratively transform noise into data (Ho et al., 2020); classifier-free guidance combines conditional and unconditional predictions (Ho & Salimans, 2022). Flow matching learns a continuous transport vector field (Lipman et al., 2023). ParetoFlow adapts this framework to offline MOO using multi-objective predictor guidance, local filtering, and neighboring evolution (Yuan et al., 2025). ARC does not replace these training objectives; it changes conditioning assignments during their inference trajectories.

Inference-time feedback and archive guidance. Evolutionary MOO has long used nondominated archives and hypervolume to retain and compare sets (Deb et al., 2002). Greedy maximization of monotone submodular objectives has the classical $(1 - 1/e)$ guarantee (Nemhauser et al., 1978). ARC imports these established ingredients at a different intervention point: a partial generative state is preserved while archive feedback reallocates its future conditioning request. The novelty claim is the closed-loop, compute-matched controller around a frozen conditional generator, not archives, hypervolume, or submodularity in isolation.

G PER-TASK HYPERVOLUME RESULTS

All tables below cover the 44 reported tasks, use the symmetric feasibility protocol, and report raw HV as mean $\pm$ standard deviation. The 100th percentile uses all returned designs; the 75th and 50th retain the best corresponding fraction by summed normalized objective. Coverage qualifications are listed in Appendix C.

Table 14: Synthetic (part 1): hypervolume at the 100th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	dtlz1	dtlz7	omnitest	vlmop2	vlmop3
MultiHead-GradNorm	10.62 ± 0.00	8.95 ± 0.00	4.55 ± 0.11	3.26 ± 0.05	40.61 ± 1.79
MultiHead-PcGrad	10.63 ± 0.00	8.95 ± 0.00	4.70 ± 0.03	3.09 ± 0.08	45.40 ± 0.48
MultiHead	10.63 ± 0.00	8.95 ± 0.00	4.76 ± 0.00	3.33 ± 0.02	45.74 ± 0.00
MultipleModels-ICT	10.63 ± 0.00	8.95 ± 0.00	4.71 ± 0.07	3.36 ± 0.00	45.74 ± 0.00
MultipleModels-IOM	10.63 ± 0.00	8.95 ± 0.00	4.76 ± 0.00	3.31 ± 0.05	45.74 ± 0.00
MultipleModels	10.63 ± 0.00	8.95 ± 0.00	4.76 ± 0.00	3.35 ± 0.01	45.74 ± 0.00
End2End	10.63 ± 0.01	8.95 ± 0.00	4.76 ± 0.00	3.36 ± 0.01	45.74 ± 0.00
End2End-GradNorm	10.63 ± 0.00	8.95 ± 0.00	4.60 ± 0.02	2.91 ± 0.14	45.73 ± 0.00
End2End-PcGrad	10.63 ± 0.00	8.95 ± 0.00	4.76 ± 0.00	3.36 ± 0.00	45.74 ± 0.00
$\mathcal{D}$(best)	10.60 ± 0.00	8.56 ± 0.00	4.53 ± 0.00	1.78 ± 0.00	45.65 ± 0.00
ParetoFlow	10.62 ± 0.01	8.95 ± 0.15	$\mathbf{4.78 \pm 0.00}$	$\mathbf{4.22 \pm 0.02}$	$\mathbf{45.92 \pm 0.00}$
PCD	$\mathbf{10.65 \pm 0.00}$	9.28 ± 0.02	4.75 ± 0.01	3.28 ± 0.13	45.91 ± 0.01
ARC (ours)	$\mathbf{10.65 \pm 0.00}$	$\mathbf{10.37 \pm 0.03}$	4.75 ± 0.01	3.15 ± 0.11	$\mathbf{45.92 \pm 0.01}$

Table 15: Synthetic (part 2): hypervolume at the 100th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	zdt1	zdt2	zdt3	zdt4	zdt6
MultiHead-GradNorm	4.17 ± 0.00	4.68 ± 0.00	5.15 ± 0.00	5.49 ± 0.00	3.95 ± 0.46
MultiHead-PcGrad	2.80 ± 0.80	2.84 ± 0.03	4.72 ± 0.60	5.47 ± 0.01	3.91 ± 0.49
MultiHead	4.17 ± 0.00	4.68 ± 0.00	5.14 ± 0.00	5.45 ± 0.01	4.61 ± 0.00
MultipleModels-ICT	4.17 ± 0.00	4.68 ± 0.00	5.15 ± 0.00	5.48 ± 0.00	4.60 ± 0.01
MultipleModels-IOM	4.17 ± 0.00	4.68 ± 0.00	5.14 ± 0.00	5.49 ± 0.01	4.61 ± 0.00
MultipleModels	4.17 ± 0.00	4.68 ± 0.00	5.14 ± 0.00	5.47 ± 0.01	4.61 ± 0.00
End2End	4.17 ± 0.00	4.68 ± 0.00	5.14 ± 0.00	$\mathbf{5.50 \pm 0.00}$	4.61 ± 0.00
End2End-GradNorm	4.16 ± 0.01	4.67 ± 0.01	5.13 ± 0.02	5.48 ± 0.00	4.61 ± 0.00
End2End-PcGrad	4.17 ± 0.00	4.68 ± 0.00	5.15 ± 0.00	5.47 ± 0.02	4.61 ± 0.00
$\mathcal{D}$(best)	4.17 ± 0.00	4.68 ± 0.00	5.15 ± 0.00	5.46 ± 0.00	4.61 ± 0.00
ParetoFlow	4.13 ± 0.04	4.66 ± 0.04	5.18 ± 0.06	5.06 ± 0.06	$\mathbf{4.73 \pm 0.04}$
PCD	4.36 ± 0.03	4.94 ± 0.01	5.49 ± 0.03	5.25 ± 0.07	4.67 ± 0.09
ARC (ours)	$\mathbf{4.84 \pm 0.00}$	$\mathbf{5.64 \pm 0.03}$	$\mathbf{5.77 \pm 0.03}$	5.18 ± 0.03	4.61 ± 0.09

Table 16: RE (part 1): hypervolume at the 100th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	re21	re22	re23	re24	re25
MultiHead-GradNorm	4.12 ± 0.04	4.81 ± 0.00	4.53 ± 0.32	4.13 ± 0.06	4.79 ± 0.00
MultiHead-PcGrad	4.17 ± 0.00	4.81 ± 0.00	4.75 ± 0.00	3.93 ± 0.20	4.10 ± 0.21
MultiHead	4.17 ± 0.00	4.81 ± 0.00	4.75 ± 0.00	$\mathbf{4.60 \pm 0.00}$	4.79 ± 0.00
MultipleModels-ICT	4.17 ± 0.00	4.81 ± 0.00	4.75 ± 0.00	$\mathbf{4.60 \pm 0.00}$	4.79 ± 0.00
MultipleModels-IOM	4.17 ± 0.00	4.81 ± 0.00	4.75 ± 0.00	$\mathbf{4.60 \pm 0.00}$	4.79 ± 0.00
MultipleModels	4.17 ± 0.00	4.81 ± 0.00	4.75 ± 0.00	$\mathbf{4.60 \pm 0.00}$	4.79 ± 0.00
End2End	4.17 ± 0.00	4.81 ± 0.00	4.75 ± 0.00	$\mathbf{4.60 \pm 0.00}$	4.79 ± 0.00
End2End-GradNorm	4.17 ± 0.00	4.77 ± 0.03	4.75 ± 0.00	$\mathbf{4.60 \pm 0.00}$	4.79 ± 0.00
End2End-PcGrad	4.17 ± 0.00	4.81 ± 0.00	4.75 ± 0.00	$\mathbf{4.60 \pm 0.00}$	4.79 ± 0.00
$\mathcal{D}$(best)	4.10 ± 0.00	4.78 ± 0.00	4.75 ± 0.00	$\mathbf{4.60 \pm 0.00}$	4.79 ± 0.00
ParetoFlow	4.10 ± 0.04	4.78 ± 0.05	4.79 ± 0.01	$\mathbf{4.60 \pm 0.00}$	4.82 ± 0.01
PCD	4.17 ± 0.02	$\mathbf{4.84 \pm 0.00}$	4.83 ± 0.00	$\mathbf{4.60 \pm 0.00}$	$\mathbf{4.84 \pm 0.00}$
ARC (ours)	$\mathbf{4.57 \pm 0.01}$	$\mathbf{4.84 \pm 0.00}$	$\mathbf{4.84 \pm 0.00}$	$\mathbf{4.60 \pm 0.00}$	$\mathbf{4.84 \pm 0.00}$

Table 17: RE (part 2): hypervolume at the 100th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	re31	re32	re33	re34	re35
MultiHead-GradNorm	10.59 ± 0.01	10.56 ± 0.02	10.56 ± 0.00	9.54 ± 0.04	10.20 ± 0.01
MultiHead-PcGrad	10.60 ± 0.01	10.61 ± 0.00	10.58 ± 0.00	9.65 ± 0.00	10.20 ± 0.00
MultiHead	10.61 ± 0.00	10.61 ± 0.00	10.58 ± 0.00	9.65 ± 0.00	10.20 ± 0.00
MultipleModels-ICT	10.60 ± 0.00	10.61 ± 0.00	10.58 ± 0.00	9.65 ± 0.00	10.20 ± 0.00
MultipleModels-IOM	10.61 ± 0.00	10.61 ± 0.00	10.57 ± 0.00	9.65 ± 0.00	10.20 ± 0.00
MultipleModels	10.61 ± 0.00	10.61 ± 0.00	10.58 ± 0.00	9.65 ± 0.00	10.20 ± 0.00
End2End	10.61 ± 0.00	10.61 ± 0.00	10.58 ± 0.00	9.65 ± 0.00	10.20 ± 0.00
End2End-GradNorm	10.58 ± 0.01	10.56 ± 0.00	10.57 ± 0.00	9.65 ± 0.00	10.20 ± 0.00
End2End-PcGrad	10.60 ± 0.00	10.60 ± 0.00	10.58 ± 0.00	9.65 ± 0.00	10.20 ± 0.00
$\mathcal{D}$(best)	10.60 ± 0.00	10.56 ± 0.00	10.56 ± 0.00	9.30 ± 0.00	10.08 ± 0.00
ParetoFlow	10.58 ± 0.05	10.62 ± 0.01	10.50 ± 0.09	9.41 ± 0.09	10.33 ± 0.01
PCD	$\mathbf{10.64 \pm 0.01}$	10.64 ± 0.00	10.38 ± 0.10	9.49 ± 0.03	10.29 ± 0.01
ARC (ours)	$\mathbf{10.64 \pm 0.01}$	$\mathbf{10.65 \pm 0.00}$	$\mathbf{10.62 \pm 0.00}$	$\mathbf{9.92 \pm 0.15}$	$\mathbf{10.50 \pm 0.01}$

Table 18: RE (part 3): hypervolume at the 100th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	re36	re37	re41	re61
MultiHead-GradNorm	7.61 ± 0.16	5.87 ± 0.01	18.24 ± 0.09	102.09 ± 0.02
MultiHead-PcGrad	7.76 ± 0.00	5.90 ± 0.00	18.66 ± 0.01	102.19 ± 0.03
MultiHead	7.76 ± 0.00	5.90 ± 0.00	18.66 ± 0.00	102.15 ± 0.04
MultipleModels-ICT	7.74 ± 0.03	5.90 ± 0.00	18.67 ± 0.00	102.05 ± 0.01
MultipleModels-IOM	7.76 ± 0.00	5.90 ± 0.00	18.66 ± 0.00	102.17 ± 0.04
MultipleModels	7.76 ± 0.00	5.90 ± 0.00	18.66 ± 0.00	102.16 ± 0.01
End2End	7.76 ± 0.00	5.90 ± 0.00	18.66 ± 0.00	102.17 ± 0.01
End2End-GradNorm	7.69 ± 0.03	5.90 ± 0.00	18.67 ± 0.01	102.16 ± 0.03
End2End-PcGrad	7.76 ± 0.00	5.90 ± 0.00	18.66 ± 0.00	102.13 ± 0.02
$\mathcal{D}$(best)	7.61 ± 0.00	5.57 ± 0.00	18.27 ± 0.00	97.49 ± 0.00
ParetoFlow	8.52 ± 0.09	5.67 ± 0.27	17.80 ± 0.22	104.42 ± 3.36
PCD	8.05 ± 0.08	5.53 ± 0.16	17.49 ± 0.48	95.46 ± 2.16
ARC (ours)	$\mathbf{10.19 \pm 0.09}$	$\mathbf{6.67 \pm 0.02}$	$\mathbf{20.38 \pm 0.28}$	$\mathbf{107.70 \pm 1.12}$

Table 19: Sci-Design: hypervolume at the 100th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	molecule	regex	zinc
MultiHead-GradNorm	2.91 ± 0.00	4.28 ± 0.00	4.65 ± 0.02
MultiHead-PcGrad	2.91 ± 0.00	4.50 ± 0.24	4.62 ± 0.04
MultiHead	2.91 ± 0.00	3.98 ± 0.00	4.66 ± 0.00
MultipleModels-ICT	2.91 ± 0.00	4.50 ± 0.24	4.69 ± 0.06
MultipleModels-IOM	2.91 ± 0.00	4.83 ± 0.00	4.72 ± 0.03
MultipleModels	2.91 ± 0.00	3.98 ± 0.00	4.66 ± 0.00
End2End	2.91 ± 0.00	3.98 ± 0.00	4.71 ± 0.04
End2End-GradNorm	2.75 ± 0.07	4.50 ± 0.24	4.61 ± 0.06
End2End-PcGrad	2.91 ± 0.00	4.50 ± 0.24	$\mathbf{4.76 \pm 0.01}$
$\mathcal{D}$(best)	2.91 ± 0.00	3.96 ± 0.00	4.52 ± 0.00
ParetoFlow	2.15 ± 0.42	3.36 ± 0.19	4.43 ± 0.09
PCD	$\mathbf{3.97 \pm 0.48}$	$\mathbf{5.78 \pm 0.22}$	4.70 ± 0.04
ARC (ours)	3.79 ± 0.61	$\mathbf{5.78 \pm 0.20}$	4.72 ± 0.06

Table 20: MO-NAS (part 1): hypervolume at the 100th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	c10mop1	c10mop2	c10mop3	c10mop4	c10mop5
MultiHead-GradNorm	4.47 ± 0.01	10.45 ± 0.01	9.57 ± 0.22	19.07 ± 0.23	48.79 ± 0.01
MultiHead-PcGrad	4.46 ± 0.00	10.52 ± 0.06	$\mathbf{10.20 \pm 0.00}$	21.67 ± 0.08	49.17 ± 0.05
MultiHead	4.45 ± 0.01	10.47 ± 0.02	10.18 ± 0.01	21.45 ± 0.13	49.13 ± 0.08
MultipleModels-ICT	4.46 ± 0.01	10.51 ± 0.07	9.91 ± 0.06	20.75 ± 0.29	48.97 ± 0.17
MultipleModels-IOM	4.46 ± 0.01	10.42 ± 0.03	10.14 ± 0.02	21.59 ± 0.13	$\mathbf{49.27 \pm 0.06}$
MultipleModels	4.45 ± 0.01	10.51 ± 0.03	$\mathbf{10.20 \pm 0.01}$	$\mathbf{21.81 \pm 0.02}$	48.92 ± 0.02
End2End	4.47 ± 0.00	10.55 ± 0.06	10.17 ± 0.02	21.67 ± 0.12	49.00 ± 0.16
End2End-GradNorm	4.48 ± 0.01	10.49 ± 0.03	9.99 ± 0.04	21.12 ± 0.16	48.90 ± 0.06
End2End-PcGrad	4.46 ± 0.01	10.54 ± 0.06	10.19 ± 0.00	21.69 ± 0.02	49.25 ± 0.03
$\mathcal{D}$(best)	4.47 ± 0.00	10.47 ± 0.00	9.21 ± 0.00	18.62 ± 0.00	45.11 ± 0.00
ParetoFlow	4.45 ± 0.02	10.51 ± 0.05	9.45 ± 0.12	19.13 ± 0.39	47.61 ± 1.15
PCD	$\mathbf{4.49 \pm 0.01}$	10.59 ± 0.01	9.73 ± 0.13	20.47 ± 0.24	42.38 ± 1.90
ARC (ours)	$\mathbf{4.49 \pm 0.01}$	$\mathbf{10.60 \pm 0.02}$	9.66 ± 0.17	20.02 ± 0.36	46.69 ± 0.93

Table 21: MO-NAS (part 2): hypervolume at the 100th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	c10mop6	c10mop8	c10mop9	in1kmop3	in1kmop4
MultiHead-GradNorm	80.35 ± 0.41	4.39 ± 0.06	8.47 ± 0.84	8.97 ± 0.12	4.32 ± 0.06
MultiHead-PcGrad	106.00 ± 0.24	4.54 ± 0.12	9.50 ± 0.20	10.13 ± 0.04	4.33 ± 0.13
MultiHead	105.81 ± 0.53	4.69 ± 0.16	10.09 ± 0.07	10.20 ± 0.03	$\mathbf{4.53 \pm 0.03}$
MultipleModels-ICT	106.06 ± 0.60	4.65 ± 0.07	9.33 ± 0.15	10.10 ± 0.02	4.39 ± 0.03
MultipleModels-IOM	$\mathbf{106.24 \pm 0.33}$	$\mathbf{4.86 \pm 0.05}$	$\mathbf{10.19 \pm 0.07}$	10.24 ± 0.01	4.35 ± 0.03
MultipleModels	105.67 ± 0.11	4.81 ± 0.04	10.06 ± 0.03	10.24 ± 0.03	4.50 ± 0.05
End2End	105.26 ± 1.65	4.76 ± 0.11	9.59 ± 0.10	$\mathbf{10.28 \pm 0.02}$	4.52 ± 0.07
End2End-GradNorm	104.48 ± 0.80	4.65 ± 0.03	9.33 ± 0.13	9.92 ± 0.02	4.39 ± 0.07
End2End-PcGrad	106.16 ± 0.56	4.71 ± 0.10	9.29 ± 0.13	10.18 ± 0.01	4.39 ± 0.05
$\mathcal{D}$(best)	103.78 ± 0.00	4.70 ± 0.00	9.64 ± 0.00	10.06 ± 0.00	4.15 ± 0.00
ParetoFlow	104.47 ± 0.68	4.61 ± 0.12	9.80 ± 0.14	9.97 ± 0.08	4.36 ± 0.05
PCD	105.64 ± 0.95	4.79 ± 0.03	9.94 ± 0.12	10.17 ± 0.02	4.33 ± 0.03
ARC (ours)	105.67 ± 0.29	4.81 ± 0.02	10.05 ± 0.09	10.19 ± 0.02	4.33 ± 0.05

Table 22: MO-NAS (part 3): hypervolume at the 100th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	in1kmop5	in1kmop6	in1kmop7	in1kmop8	in1kmop9
MultiHead-GradNorm	4.53 ± 0.05	9.35 ± 0.24	3.62 ± 0.56	8.14 ± 0.71	11.80 ± 2.13
MultiHead-PcGrad	4.51 ± 0.11	9.65 ± 0.02	4.44 ± 0.06	9.11 ± 0.10	19.23 ± 0.22
MultiHead	$\mathbf{4.63 \pm 0.04}$	9.95 ± 0.09	$\mathbf{4.61 \pm 0.02}$	9.57 ± 0.11	19.77 ± 0.12
MultipleModels-ICT	4.48 ± 0.09	9.69 ± 0.02	4.36 ± 0.09	8.98 ± 0.02	18.91 ± 0.31
MultipleModels-IOM	4.49 ± 0.02	9.60 ± 0.09	4.37 ± 0.21	$\mathbf{9.82 \pm 0.02}$	$\mathbf{20.27 \pm 0.05}$
MultipleModels	4.59 ± 0.05	$\mathbf{10.02 \pm 0.03}$	4.51 ± 0.02	9.64 ± 0.02	19.64 ± 0.39
End2End	$\mathbf{4.63 \pm 0.03}$	9.86 ± 0.10	4.53 ± 0.02	9.61 ± 0.08	19.90 ± 0.24
End2End-GradNorm	4.53 ± 0.05	9.65 ± 0.04	4.42 ± 0.04	8.97 ± 0.12	18.80 ± 0.16
End2End-PcGrad	4.49 ± 0.04	9.75 ± 0.01	4.44 ± 0.12	9.50 ± 0.07	19.69 ± 0.21
$\mathcal{D}$(best)	4.30 ± 0.00	9.15 ± 0.00	3.94 ± 0.00	9.28 ± 0.00	18.87 ± 0.00
ParetoFlow	4.44 ± 0.05	9.58 ± 0.12	3.80 ± 0.11	9.31 ± 0.09	18.62 ± 0.40
PCD	4.51 ± 0.03	9.67 ± 0.13	4.30 ± 0.07	9.32 ± 0.07	19.12 ± 0.33
ARC (ours)	4.51 ± 0.03	9.70 ± 0.19	4.31 ± 0.06	9.32 ± 0.05	19.44 ± 0.30

Table 23: MORL: hypervolume at the 100th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	mo_hopper_v2	mo_swimmer_v2
MultiHead-GradNorm	5.46 ± 0.17	3.02 ± 0.86
MultiHead-PcGrad	5.03 ± 0.39	1.07 ± 0.13
MultiHead	5.49 ± 0.58	2.94 ± 0.98
MultipleModels-ICT	5.86 ± 0.24	3.65 ± 0.00
MultipleModels-IOM	4.76 ± 0.02	1.87 ± 0.43
MultipleModels	5.92 ± 0.19	1.90 ± 0.14
End2End	5.56 ± 0.58	1.03 ± 0.06
End2End-GradNorm	6.17 ± 0.04	3.66 ± 0.01
End2End-PcGrad	6.16 ± 0.03	3.66 ± 0.00
$\mathcal{D}$(best)	6.33 ± 0.00	$\mathbf{3.70 \pm 0.00}$
ParetoFlow	5.92 ± 0.01	3.45 ± 0.13
PCD	$\mathbf{6.44 \pm 0.30}$	3.61 ± 0.09
ARC (ours)	6.37 ± 0.29	3.61 ± 0.10

Table 24: Synthetic (part 1): hypervolume at the 75th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	dtlz1	dtlz7	omnitest	vlmop2	vlmop3
MultiHead-GradNorm	10.60 ± 0.01	8.85 ± 0.01	3.81 ± 0.41	2.80 ± 0.20	38.94 ± 2.33
MultiHead-PcGrad	10.63 ± 0.00	8.86 ± 0.00	4.41 ± 0.34	3.04 ± 0.14	42.84 ± 2.48
MultiHead	10.62 ± 0.00	8.86 ± 0.00	4.68 ± 0.05	3.32 ± 0.02	45.72 ± 0.00
MultipleModels-ICT	10.54 ± 0.07	8.86 ± 0.00	4.68 ± 0.11	3.36 ± 0.00	45.72 ± 0.00
MultipleModels-IOM	10.63 ± 0.00	8.86 ± 0.00	4.76 ± 0.00	3.30 ± 0.06	45.72 ± 0.00
MultipleModels	10.61 ± 0.00	8.86 ± 0.00	4.72 ± 0.05	3.35 ± 0.01	45.72 ± 0.00
End2End	10.32 ± 0.12	8.86 ± 0.00	4.72 ± 0.05	3.35 ± 0.01	45.72 ± 0.00
End2End-GradNorm	10.63 ± 0.00	8.86 ± 0.00	4.05 ± 0.43	2.46 ± 0.17	45.23 ± 0.70
End2End-PcGrad	10.63 ± 0.00	8.86 ± 0.00	4.76 ± 0.00	3.35 ± 0.01	45.72 ± 0.00
$\mathcal{D}$(best)	10.60 ± 0.00	8.56 ± 0.00	4.53 ± 0.00	1.78 ± 0.00	45.65 ± 0.00
ParetoFlow	10.48 ± 0.02	8.76 ± 0.26	$\mathbf{4.78 \pm 0.00}$	$\mathbf{4.22 \pm 0.02}$	$\mathbf{45.89 \pm 0.02}$
PCD	10.63 ± 0.01	9.20 ± 0.07	4.36 ± 0.06	2.47 ± 0.14	45.82 ± 0.03
ARC (ours)	$\mathbf{10.64 \pm 0.00}$	$\mathbf{10.15 \pm 0.27}$	4.28 ± 0.13	1.99 ± 0.11	45.85 ± 0.02

Table 25: Synthetic (part 2): hypervolume at the 75th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	zdt1	zdt2	zdt3	zdt4	zdt6
MultiHead-GradNorm	4.15 ± 0.00	4.68 ± 0.00	5.11 ± 0.05	5.20 ± 0.32	3.69 ± 0.09
MultiHead-PcGrad	2.20 ± 0.14	2.83 ± 0.03	4.46 ± 0.88	4.75 ± 0.04	3.91 ± 0.50
MultiHead	4.16 ± 0.00	4.68 ± 0.00	5.13 ± 0.01	4.87 ± 0.10	3.82 ± 0.01
MultipleModels-ICT	4.15 ± 0.00	4.68 ± 0.00	5.14 ± 0.00	4.72 ± 0.17	4.03 ± 0.82
MultipleModels-IOM	4.16 ± 0.00	4.67 ± 0.00	4.70 ± 0.51	4.82 ± 0.04	4.33 ± 0.39
MultipleModels	4.16 ± 0.00	4.68 ± 0.00	5.13 ± 0.01	4.87 ± 0.14	3.83 ± 0.01
End2End	4.16 ± 0.00	4.68 ± 0.00	5.14 ± 0.00	4.81 ± 0.08	4.34 ± 0.37
End2End-GradNorm	2.81 ± 0.24	3.17 ± 0.03	3.51 ± 0.01	4.85 ± 0.40	4.23 ± 0.52
End2End-PcGrad	4.16 ± 0.00	4.68 ± 0.00	5.14 ± 0.01	4.74 ± 0.17	3.76 ± 1.20
$\mathcal{D}$(best)	4.17 ± 0.00	4.68 ± 0.00	5.15 ± 0.00	$\mathbf{5.46 \pm 0.00}$	4.61 ± 0.00
ParetoFlow	4.13 ± 0.04	4.64 ± 0.03	5.07 ± 0.07	4.82 ± 0.06	$\mathbf{4.67 \pm 0.11}$
PCD	4.28 ± 0.03	4.80 ± 0.01	5.31 ± 0.02	4.86 ± 0.02	4.36 ± 0.01
ARC (ours)	$\mathbf{4.66 \pm 0.05}$	$\mathbf{5.27 \pm 0.05}$	$\mathbf{5.47 \pm 0.08}$	4.85 ± 0.02	4.33 ± 0.03

Table 26: RE (part 1): hypervolume at the 75th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	re21	re22	re23	re24	re25
MultiHead-GradNorm	3.53 ± 0.43	3.98 ± 0.53	4.21 ± 0.11	4.03 ± 0.08	4.47 ± 0.07
MultiHead-PcGrad	4.15 ± 0.01	3.87 ± 0.66	4.27 ± 0.12	3.88 ± 0.13	4.09 ± 0.21
MultiHead	4.15 ± 0.01	4.72 ± 0.00	4.55 ± 0.15	4.19 ± 0.03	4.38 ± 0.09
MultipleModels-ICT	4.14 ± 0.00	4.02 ± 0.50	4.51 ± 0.00	4.20 ± 0.02	4.21 ± 0.14
MultipleModels-IOM	4.14 ± 0.01	4.06 ± 0.40	4.51 ± 0.00	4.13 ± 0.05	4.32 ± 0.03
MultipleModels	4.15 ± 0.01	4.45 ± 0.39	4.48 ± 0.00	4.20 ± 0.02	4.38 ± 0.03
End2End	4.14 ± 0.01	4.52 ± 0.15	4.57 ± 0.13	4.17 ± 0.06	4.45 ± 0.02
End2End-GradNorm	4.14 ± 0.00	4.13 ± 0.24	4.30 ± 0.02	4.21 ± 0.00	4.46 ± 0.23
End2End-PcGrad	4.13 ± 0.00	4.07 ± 0.47	4.49 ± 0.02	4.21 ± 0.00	4.58 ± 0.30
$\mathcal{D}$(best)	4.10 ± 0.00	4.78 ± 0.00	4.75 ± 0.00	$\mathbf{4.60 \pm 0.00}$	4.79 ± 0.00
ParetoFlow	4.04 ± 0.04	4.73 ± 0.06	4.75 ± 0.00	4.51 ± 0.03	4.81 ± 0.01
PCD	4.14 ± 0.02	4.82 ± 0.01	4.83 ± 0.00	$\mathbf{4.60 \pm 0.00}$	4.83 ± 0.00
ARC (ours)	$\mathbf{4.57 \pm 0.01}$	$\mathbf{4.83 \pm 0.01}$	$\mathbf{4.84 \pm 0.00}$	$\mathbf{4.60 \pm 0.00}$	$\mathbf{4.84 \pm 0.00}$

Table 27: RE (part 2): hypervolume at the 75th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	re31	re32	re33	re34	re35
MultiHead-GradNorm	9.41 ± 0.92	9.33 ± 0.13	9.68 ± 0.44	8.82 ± 0.45	10.09 ± 0.05
MultiHead-PcGrad	9.70 ± 0.92	9.24 ± 0.21	10.41 ± 0.07	9.60 ± 0.00	9.66 ± 0.09
MultiHead	9.12 ± 1.01	10.13 ± 0.52	10.06 ± 0.31	9.60 ± 0.00	9.96 ± 0.13
MultipleModels-ICT	8.56 ± 0.08	8.88 ± 0.36	10.50 ± 0.01	9.60 ± 0.00	10.05 ± 0.00
MultipleModels-IOM	10.32 ± 0.22	9.29 ± 0.16	10.35 ± 0.16	9.60 ± 0.00	10.05 ± 0.00
MultipleModels	10.38 ± 0.11	9.75 ± 0.48	10.37 ± 0.11	9.60 ± 0.00	10.04 ± 0.15
End2End	8.72 ± 0.33	9.75 ± 0.50	10.29 ± 0.07	9.60 ± 0.00	9.93 ± 0.15
End2End-GradNorm	8.51 ± 0.22	9.32 ± 0.02	10.49 ± 0.01	9.60 ± 0.00	10.05 ± 0.00
End2End-PcGrad	8.30 ± 0.51	9.20 ± 0.05	10.50 ± 0.01	9.60 ± 0.00	9.40 ± 0.99
$\mathcal{D}$(best)	10.60 ± 0.00	10.56 ± 0.00	10.56 ± 0.00	9.30 ± 0.00	10.08 ± 0.00
ParetoFlow	10.44 ± 0.08	10.60 ± 0.02	10.42 ± 0.13	9.31 ± 0.10	10.31 ± 0.02
PCD	10.63 ± 0.00	10.63 ± 0.00	10.37 ± 0.12	9.44 ± 0.05	10.28 ± 0.01
ARC (ours)	$\mathbf{10.64 \pm 0.01}$	$\mathbf{10.64 \pm 0.00}$	$\mathbf{10.62 \pm 0.00}$	$\mathbf{9.89 \pm 0.15}$	$\mathbf{10.50 \pm 0.01}$

Table 28: RE (part 3): hypervolume at the 75th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	re36	re37	re41	re61
MultiHead-GradNorm	5.55 ± 0.34	5.62 ± 0.03	14.29 ± 0.41	99.57 ± 1.00
MultiHead-PcGrad	5.73 ± 0.16	5.61 ± 0.00	16.85 ± 0.35	99.84 ± 0.02
MultiHead	5.74 ± 0.22	5.61 ± 0.00	16.73 ± 0.01	99.77 ± 0.03
MultipleModels-ICT	5.96 ± 0.02	5.61 ± 0.00	17.18 ± 0.08	101.42 ± 0.12
MultipleModels-IOM	5.75 ± 0.21	5.61 ± 0.00	16.74 ± 0.06	100.66 ± 0.58
MultipleModels	5.76 ± 0.20	5.61 ± 0.00	16.68 ± 0.02	100.96 ± 0.07
End2End	5.61 ± 0.00	5.61 ± 0.00	16.73 ± 0.07	100.66 ± 0.51
End2End-GradNorm	5.89 ± 0.02	5.61 ± 0.00	16.69 ± 0.03	100.16 ± 0.32
End2End-PcGrad	5.61 ± 0.00	5.61 ± 0.00	16.81 ± 0.03	100.64 ± 0.79
$\mathcal{D}$(best)	7.61 ± 0.00	5.57 ± 0.00	18.27 ± 0.00	97.49 ± 0.00
ParetoFlow	8.17 ± 0.11	5.55 ± 0.26	17.51 ± 0.36	103.79 ± 3.49
PCD	7.79 ± 0.03	5.52 ± 0.16	17.42 ± 0.44	95.25 ± 2.08
ARC (ours)	$\mathbf{9.46 \pm 0.33}$	$\mathbf{6.66 \pm 0.02}$	$\mathbf{20.26 \pm 0.32}$	$\mathbf{107.54 \pm 1.14}$

Table 29: Sci-Design: hypervolume at the 75th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	molecule	regex	zinc
MultiHead-GradNorm	2.36 ± 0.05	2.99 ± 0.00	4.55 ± 0.07
MultiHead-PcGrad	1.90 ± 0.29	4.35 ± 0.40	4.49 ± 0.02
MultiHead	2.41 ± 0.01	2.99 ± 0.00	4.52 ± 0.00
MultipleModels-ICT	2.46 ± 0.06	3.46 ± 0.66	4.55 ± 0.06
MultipleModels-IOM	1.48 ± 0.06	4.83 ± 0.00	4.53 ± 0.01
MultipleModels	2.46 ± 0.06	2.99 ± 0.00	$\mathbf{4.59 \pm 0.04}$
End2End	2.41 ± 0.01	2.99 ± 0.00	4.56 ± 0.04
End2End-GradNorm	1.51 ± 0.00	3.46 ± 0.66	4.45 ± 0.08
End2End-PcGrad	2.50 ± 0.06	3.42 ± 0.61	4.52 ± 0.02
$\mathcal{D}$(best)	$\mathbf{2.91 \pm 0.00}$	3.96 ± 0.00	4.52 ± 0.00
ParetoFlow	1.95 ± 0.54	3.26 ± 0.00	4.30 ± 0.30
PCD	2.01 ± 0.33	4.25 ± 0.66	4.49 ± 0.03
ARC (ours)	1.93 ± 0.30	$\mathbf{5.22 \pm 0.62}$	4.51 ± 0.03

Table 30: MO-NAS (part 1): hypervolume at the 75th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	c10mop1	c10mop2	c10mop3	c10mop4	c10mop5
MultiHead-GradNorm	4.45 ± 0.01	10.44 ± 0.01	8.97 ± 0.28	18.69 ± 0.25	45.27 ± 1.06
MultiHead-PcGrad	4.44 ± 0.01	10.50 ± 0.06	$\mathbf{10.13 \pm 0.01}$	21.15 ± 0.20	48.74 ± 0.04
MultiHead	4.44 ± 0.01	10.44 ± 0.00	10.08 ± 0.06	21.07 ± 0.27	48.98 ± 0.06
MultipleModels-ICT	4.44 ± 0.01	10.45 ± 0.00	9.83 ± 0.04	20.59 ± 0.30	46.78 ± 0.67
MultipleModels-IOM	4.42 ± 0.04	10.42 ± 0.03	10.11 ± 0.01	$\mathbf{21.48 \pm 0.09}$	49.02 ± 0.07
MultipleModels	4.43 ± 0.01	10.46 ± 0.01	10.10 ± 0.02	21.31 ± 0.02	48.76 ± 0.03
End2End	4.46 ± 0.00	$\mathbf{10.51 \pm 0.04}$	10.11 ± 0.01	21.33 ± 0.13	48.78 ± 0.01
End2End-GradNorm	4.46 ± 0.01	10.44 ± 0.01	9.85 ± 0.04	21.07 ± 0.16	48.19 ± 0.59
End2End-PcGrad	4.45 ± 0.00	10.49 ± 0.10	$\mathbf{10.13 \pm 0.02}$	21.38 ± 0.03	$\mathbf{49.12 \pm 0.01}$
$\mathcal{D}$(best)	$\mathbf{4.47 \pm 0.00}$	10.47 ± 0.00	9.21 ± 0.00	18.62 ± 0.00	45.11 ± 0.00
ParetoFlow	4.41 ± 0.02	10.45 ± 0.04	9.34 ± 0.09	19.04 ± 0.33	45.46 ± 3.94
PCD	4.44 ± 0.01	10.45 ± 0.02	9.56 ± 0.21	20.29 ± 0.29	40.04 ± 0.43
ARC (ours)	4.43 ± 0.00	10.47 ± 0.05	9.30 ± 0.22	19.64 ± 0.62	40.82 ± 0.45

Table 31: MO-NAS (part 2): hypervolume at the 75th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	c10mop6	c10mop8	c10mop9	in1kmop3	in1kmop4
MultiHead-GradNorm	74.51 ± 5.15	4.23 ± 0.05	7.94 ± 1.11	8.97 ± 0.12	4.23 ± 0.07
MultiHead-PcGrad	104.45 ± 0.66	4.39 ± 0.12	9.12 ± 0.18	10.10 ± 0.02	4.27 ± 0.13
MultiHead	104.89 ± 0.79	4.56 ± 0.18	9.90 ± 0.12	10.18 ± 0.02	$\mathbf{4.45 \pm 0.04}$
MultipleModels-ICT	104.53 ± 1.24	4.40 ± 0.08	9.04 ± 0.12	10.09 ± 0.03	4.32 ± 0.03
MultipleModels-IOM	$\mathbf{105.45 \pm 0.67}$	$\mathbf{4.77 \pm 0.08}$	$\mathbf{9.95 \pm 0.08}$	10.21 ± 0.01	4.29 ± 0.03
MultipleModels	104.36 ± 0.58	4.64 ± 0.11	9.77 ± 0.08	10.22 ± 0.03	4.44 ± 0.06
End2End	104.05 ± 2.44	4.59 ± 0.09	9.05 ± 0.10	$\mathbf{10.26 \pm 0.02}$	$\mathbf{4.45 \pm 0.06}$
End2End-GradNorm	91.99 ± 0.07	4.37 ± 0.06	8.80 ± 0.33	9.84 ± 0.08	4.30 ± 0.06
End2End-PcGrad	104.88 ± 0.39	4.59 ± 0.13	8.81 ± 0.12	10.16 ± 0.01	4.27 ± 0.07
$\mathcal{D}$(best)	103.78 ± 0.00	4.70 ± 0.00	9.64 ± 0.00	10.06 ± 0.00	4.15 ± 0.00
ParetoFlow	102.22 ± 1.48	4.44 ± 0.03	9.57 ± 0.32	9.93 ± 0.10	4.23 ± 0.11
PCD	103.99 ± 0.96	4.67 ± 0.04	9.53 ± 0.09	10.15 ± 0.02	4.18 ± 0.03
ARC (ours)	104.42 ± 0.48	4.67 ± 0.02	9.62 ± 0.10	10.18 ± 0.02	4.17 ± 0.03

Table 32: MO-NAS (part 3): hypervolume at the 75th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	in1kmop5	in1kmop6	in1kmop7	in1kmop8	in1kmop9
MultiHead-GradNorm	4.46 ± 0.05	8.84 ± 0.63	3.46 ± 0.56	8.08 ± 0.75	11.73 ± 2.07
MultiHead-PcGrad	4.43 ± 0.11	9.38 ± 0.12	4.35 ± 0.08	8.98 ± 0.12	19.16 ± 0.24
MultiHead	$\mathbf{4.57 \pm 0.02}$	$\mathbf{9.75 \pm 0.11}$	$\mathbf{4.56 \pm 0.01}$	9.52 ± 0.11	19.41 ± 0.29
MultipleModels-ICT	4.36 ± 0.04	9.41 ± 0.13	4.22 ± 0.06	8.92 ± 0.03	18.78 ± 0.25
MultipleModels-IOM	4.45 ± 0.04	9.40 ± 0.05	4.19 ± 0.32	$\mathbf{9.77 \pm 0.01}$	$\mathbf{20.21 \pm 0.04}$
MultipleModels	4.53 ± 0.06	$\mathbf{9.75 \pm 0.08}$	4.44 ± 0.03	9.61 ± 0.02	19.60 ± 0.40
End2End	4.56 ± 0.02	9.63 ± 0.07	4.48 ± 0.04	9.59 ± 0.08	19.82 ± 0.24
End2End-GradNorm	4.07 ± 0.31	9.27 ± 0.19	4.33 ± 0.04	8.94 ± 0.12	18.74 ± 0.13
End2End-PcGrad	4.40 ± 0.04	9.46 ± 0.07	4.35 ± 0.16	9.47 ± 0.07	19.63 ± 0.19
$\mathcal{D}$(best)	4.30 ± 0.00	9.15 ± 0.00	3.94 ± 0.00	9.28 ± 0.00	18.87 ± 0.00
ParetoFlow	4.43 ± 0.04	9.46 ± 0.15	3.65 ± 0.17	9.22 ± 0.10	18.57 ± 0.41
PCD	4.36 ± 0.05	9.29 ± 0.13	3.86 ± 0.11	9.28 ± 0.05	19.03 ± 0.30
ARC (ours)	4.37 ± 0.04	9.21 ± 0.12	3.96 ± 0.04	9.26 ± 0.05	19.27 ± 0.40

Table 33: MORL: hypervolume at the 75th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	mo_hopper_v2	mo_swimmer_v2
MultiHead-GradNorm	5.00 ± 0.36	1.16 ± 0.22
MultiHead-PcGrad	4.74 ± 0.00	0.87 ± 0.02
MultiHead	4.75 ± 0.00	2.21 ± 1.11
MultipleModels-ICT	4.75 ± 0.00	2.79 ± 0.74
MultipleModels-IOM	4.75 ± 0.00	1.07 ± 0.16
MultipleModels	5.57 ± 0.04	0.90 ± 0.03
End2End	5.30 ± 0.40	0.88 ± 0.00
End2End-GradNorm	5.58 ± 0.03	3.61 ± 0.01
End2End-PcGrad	5.62 ± 0.02	3.63 ± 0.01
$\mathcal{D}$(best)	$\mathbf{6.33 \pm 0.00}$	$\mathbf{3.70 \pm 0.00}$
ParetoFlow	5.59 ± 0.03	2.49 ± 0.15
PCD	5.18 ± 0.16	2.80 ± 0.07
ARC (ours)	5.17 ± 0.05	2.90 ± 0.04

Table 34: Synthetic (part 1): hypervolume at the 50th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	dtlz1	dtlz7	omnitest	vlmop2	vlmop3
MultiHead-GradNorm	10.04 ± 0.04	8.77 ± 0.00	2.82 ± 0.40	2.27 ± 0.07	37.22 ± 2.41
MultiHead-PcGrad	10.28 ± 0.06	8.77 ± 0.00	3.97 ± 0.94	2.77 ± 0.20	40.64 ± 2.80
MultiHead	10.18 ± 0.01	8.77 ± 0.00	4.68 ± 0.05	3.15 ± 0.01	45.70 ± 0.00
MultipleModels-ICT	10.23 ± 0.07	8.77 ± 0.00	4.64 ± 0.10	3.35 ± 0.01	44.74 ± 1.34
MultipleModels-IOM	10.36 ± 0.03	8.77 ± 0.00	4.74 ± 0.01	3.29 ± 0.07	45.70 ± 0.00
MultipleModels	10.17 ± 0.01	8.77 ± 0.00	4.71 ± 0.05	3.35 ± 0.01	$\mathbf{45.71 \pm 0.00}$
End2End	9.96 ± 0.07	8.77 ± 0.00	4.71 ± 0.05	3.35 ± 0.01	$\mathbf{45.71 \pm 0.00}$
End2End-GradNorm	10.33 ± 0.04	8.77 ± 0.00	3.46 ± 0.35	2.23 ± 0.14	44.28 ± 0.03
End2End-PcGrad	10.31 ± 0.01	8.77 ± 0.00	4.76 ± 0.00	3.32 ± 0.03	45.69 ± 0.01
$\mathcal{D}$(best)	10.60 ± 0.00	8.56 ± 0.00	4.53 ± 0.00	1.78 ± 0.00	45.65 ± 0.00
ParetoFlow	9.85 ± 0.73	8.46 ± 0.25	$\mathbf{4.78 \pm 0.00}$	$\mathbf{4.21 \pm 0.02}$	45.33 ± 0.32
PCD	10.60 ± 0.02	9.08 ± 0.07	3.87 ± 0.06	2.05 ± 0.06	$\mathbf{45.71 \pm 0.04}$
ARC (ours)	$\mathbf{10.62 \pm 0.00}$	$\mathbf{9.22 \pm 0.37}$	3.75 ± 0.12	1.51 ± 0.06	45.66 ± 0.04

Table 35: Synthetic (part 2): hypervolume at the 50th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	zdt1	zdt2	zdt3	zdt4	zdt6
MultiHead-GradNorm	3.63 ± 0.74	4.67 ± 0.00	5.12 ± 0.00	4.67 ± 0.04	3.68 ± 0.09
MultiHead-PcGrad	2.18 ± 0.15	2.82 ± 0.03	4.04 ± 0.80	4.67 ± 0.02	3.66 ± 0.16
MultiHead	4.15 ± 0.00	4.67 ± 0.00	4.12 ± 0.71	4.77 ± 0.13	3.80 ± 0.01
MultipleModels-ICT	4.14 ± 0.01	4.67 ± 0.00	5.13 ± 0.00	4.66 ± 0.17	3.47 ± 0.43
MultipleModels-IOM	4.11 ± 0.06	4.67 ± 0.00	3.63 ± 0.01	4.70 ± 0.08	3.81 ± 0.03
MultipleModels	4.15 ± 0.00	4.67 ± 0.00	4.62 ± 0.64	4.74 ± 0.17	3.82 ± 0.01
End2End	4.15 ± 0.00	4.67 ± 0.00	4.23 ± 0.63	4.72 ± 0.11	3.85 ± 0.03
End2End-GradNorm	2.62 ± 0.00	3.14 ± 0.03	3.47 ± 0.03	4.61 ± 0.14	3.40 ± 0.38
End2End-PcGrad	4.11 ± 0.05	4.67 ± 0.00	3.35 ± 0.02	4.70 ± 0.19	3.18 ± 0.80
$\mathcal{D}$(best)	4.17 ± 0.00	4.68 ± 0.00	5.15 ± 0.00	$\mathbf{5.46 \pm 0.00}$	$\mathbf{4.61 \pm 0.00}$
ParetoFlow	4.12 ± 0.04	4.58 ± 0.04	5.07 ± 0.07	4.66 ± 0.11	4.39 ± 0.05
PCD	4.18 ± 0.04	4.74 ± 0.02	5.20 ± 0.02	4.71 ± 0.02	4.31 ± 0.02
ARC (ours)	$\mathbf{4.26 \pm 0.09}$	$\mathbf{5.03 \pm 0.14}$	$\mathbf{5.23 \pm 0.02}$	4.70 ± 0.02	3.81 ± 0.90

Table 36: RE (part 1): hypervolume at the 50th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	re21	re22	re23	re24	re25
MultiHead-GradNorm	3.51 ± 0.40	3.92 ± 0.57	4.15 ± 0.08	3.92 ± 0.07	4.42 ± 0.09
MultiHead-PcGrad	4.12 ± 0.01	3.54 ± 0.22	4.21 ± 0.08	3.84 ± 0.07	4.06 ± 0.16
MultiHead	4.11 ± 0.02	4.09 ± 0.28	4.47 ± 0.07	4.18 ± 0.03	4.36 ± 0.09
MultipleModels-ICT	4.09 ± 0.01	3.77 ± 0.18	4.47 ± 0.01	4.13 ± 0.05	4.18 ± 0.16
MultipleModels-IOM	4.10 ± 0.03	3.71 ± 0.06	4.34 ± 0.13	4.00 ± 0.03	4.30 ± 0.03
MultipleModels	4.11 ± 0.02	3.93 ± 0.06	4.47 ± 0.01	4.19 ± 0.02	4.36 ± 0.03
End2End	4.09 ± 0.02	4.28 ± 0.30	4.45 ± 0.03	4.17 ± 0.07	4.43 ± 0.02
End2End-GradNorm	4.11 ± 0.01	3.91 ± 0.03	4.27 ± 0.02	4.21 ± 0.00	4.34 ± 0.14
End2End-PcGrad	4.09 ± 0.01	3.95 ± 0.39	4.46 ± 0.04	4.21 ± 0.00	4.27 ± 0.18
$\mathcal{D}$(best)	4.10 ± 0.00	4.78 ± 0.00	4.75 ± 0.00	$\mathbf{4.60 \pm 0.00}$	4.79 ± 0.00
ParetoFlow	3.99 ± 0.04	4.57 ± 0.02	4.66 ± 0.01	4.42 ± 0.02	4.80 ± 0.00
PCD	4.11 ± 0.02	4.78 ± 0.01	4.81 ± 0.00	$\mathbf{4.60 \pm 0.00}$	4.81 ± 0.00
ARC (ours)	$\mathbf{4.47 \pm 0.08}$	$\mathbf{4.82 \pm 0.01}$	$\mathbf{4.82 \pm 0.01}$	$\mathbf{4.60 \pm 0.00}$	$\mathbf{4.82 \pm 0.01}$

Table 37: RE (part 2): hypervolume at the 50th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	re31	re32	re33	re34	re35
MultiHead-GradNorm	8.46 ± 0.87	9.33 ± 0.12	9.68 ± 0.44	5.82 ± 0.14	8.45 ± 1.13
MultiHead-PcGrad	8.81 ± 0.95	9.21 ± 0.21	10.09 ± 0.19	9.52 ± 0.00	7.49 ± 0.34
MultiHead	8.19 ± 0.40	9.38 ± 0.08	9.72 ± 0.19	9.52 ± 0.00	7.44 ± 0.59
MultipleModels-ICT	8.56 ± 0.08	8.77 ± 0.36	10.40 ± 0.06	9.52 ± 0.00	9.76 ± 0.01
MultipleModels-IOM	8.92 ± 1.07	9.11 ± 0.24	9.96 ± 0.00	9.52 ± 0.00	9.77 ± 0.00
MultipleModels	9.15 ± 0.81	9.35 ± 0.11	10.14 ± 0.13	9.52 ± 0.00	9.94 ± 0.23
End2End	8.49 ± 0.03	9.27 ± 0.12	10.04 ± 0.13	9.52 ± 0.00	8.93 ± 1.32
End2End-GradNorm	8.48 ± 0.21	9.31 ± 0.01	10.38 ± 0.07	9.52 ± 0.00	9.77 ± 0.00
End2End-PcGrad	8.29 ± 0.51	9.15 ± 0.04	10.40 ± 0.07	9.52 ± 0.00	8.66 ± 0.97
$\mathcal{D}$(best)	10.60 ± 0.00	10.56 ± 0.00	10.56 ± 0.00	9.30 ± 0.00	10.08 ± 0.00
ParetoFlow	10.32 ± 0.07	10.58 ± 0.02	10.40 ± 0.12	8.97 ± 0.16	10.23 ± 0.06
PCD	10.59 ± 0.01	10.61 ± 0.01	10.36 ± 0.11	9.28 ± 0.06	10.22 ± 0.02
ARC (ours)	$\mathbf{10.63 \pm 0.01}$	$\mathbf{10.62 \pm 0.01}$	$\mathbf{10.62 \pm 0.00}$	$\mathbf{9.87 \pm 0.14}$	$\mathbf{10.46 \pm 0.02}$

Table 38: RE (part 3): hypervolume at the 50th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	re36	re37	re41	re61
MultiHead-GradNorm	5.25 ± 0.39	4.59 ± 0.76	12.97 ± 0.41	89.83 ± 5.31
MultiHead-PcGrad	5.59 ± 0.02	5.37 ± 0.00	16.54 ± 0.38	95.99 ± 1.38
MultiHead	5.74 ± 0.22	5.37 ± 0.00	16.18 ± 0.14	95.76 ± 0.87
MultipleModels-ICT	5.96 ± 0.02	5.38 ± 0.00	16.82 ± 0.18	99.87 ± 0.62
MultipleModels-IOM	5.75 ± 0.21	5.37 ± 0.00	16.19 ± 0.29	97.46 ± 1.70
MultipleModels	5.76 ± 0.20	5.37 ± 0.00	16.29 ± 0.06	96.54 ± 1.72
End2End	5.61 ± 0.00	5.37 ± 0.00	16.31 ± 0.03	96.50 ± 1.55
End2End-GradNorm	5.81 ± 0.12	5.38 ± 0.01	16.49 ± 0.03	97.45 ± 1.08
End2End-PcGrad	5.61 ± 0.00	5.37 ± 0.00	16.42 ± 0.17	98.59 ± 1.12
$\mathcal{D}$(best)	7.61 ± 0.00	5.57 ± 0.00	18.27 ± 0.00	97.49 ± 0.00
ParetoFlow	7.79 ± 0.12	5.27 ± 0.27	17.44 ± 0.37	101.19 ± 4.24
PCD	7.65 ± 0.04	5.43 ± 0.14	17.36 ± 0.44	94.96 ± 2.18
ARC (ours)	$\mathbf{8.66 \pm 0.19}$	$\mathbf{6.63 \pm 0.02}$	$\mathbf{20.04 \pm 0.31}$	$\mathbf{106.97 \pm 1.71}$

Table 39: Sci-Design: hypervolume at the 50th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	molecule	regex	zinc
MultiHead-GradNorm	1.59 ± 0.09	2.99 ± 0.00	4.41 ± 0.10
MultiHead-PcGrad	1.44 ± 0.00	3.42 ± 0.61	4.27 ± 0.05
MultiHead	2.01 ± 0.36	2.99 ± 0.00	4.43 ± 0.01
MultipleModels-ICT	2.27 ± 0.04	2.99 ± 0.00	4.37 ± 0.05
MultipleModels-IOM	1.44 ± 0.01	3.42 ± 0.61	4.38 ± 0.02
MultipleModels	2.30 ± 0.00	2.99 ± 0.00	4.40 ± 0.03
End2End	2.29 ± 0.01	2.99 ± 0.00	4.42 ± 0.05
End2End-GradNorm	1.51 ± 0.00	2.99 ± 0.00	4.36 ± 0.11
End2End-PcGrad	1.73 ± 0.36	2.99 ± 0.00	4.41 ± 0.02
$\mathcal{D}$(best)	$\mathbf{2.91 \pm 0.00}$	3.96 ± 0.00	$\mathbf{4.52 \pm 0.00}$
ParetoFlow	1.63 ± 0.12	3.26 ± 0.00	4.22 ± 0.34
PCD	1.55 ± 0.05	4.10 ± 0.36	4.27 ± 0.05
ARC (ours)	1.55 ± 0.05	$\mathbf{4.77 \pm 0.23}$	4.28 ± 0.14

Table 40: MO-NAS (part 1): hypervolume at the 50th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	c10mop1	c10mop2	c10mop3	c10mop4	c10mop5
MultiHead-GradNorm	4.40 ± 0.04	10.40 ± 0.04	8.68 ± 0.49	16.40 ± 0.26	37.75 ± 0.71
MultiHead-PcGrad	4.42 ± 0.01	10.45 ± 0.03	10.03 ± 0.01	19.98 ± 0.13	47.82 ± 0.37
MultiHead	4.39 ± 0.03	10.41 ± 0.01	$\mathbf{10.04 \pm 0.02}$	20.15 ± 0.11	$\mathbf{48.83 \pm 0.09}$
MultipleModels-ICT	4.41 ± 0.00	10.43 ± 0.00	9.67 ± 0.12	19.13 ± 0.66	45.50 ± 0.41
MultipleModels-IOM	4.31 ± 0.05	10.39 ± 0.02	9.91 ± 0.08	20.12 ± 0.26	48.25 ± 0.33
MultipleModels	4.37 ± 0.01	10.43 ± 0.01	9.99 ± 0.02	20.15 ± 0.53	48.57 ± 0.09
End2End	4.43 ± 0.01	10.38 ± 0.06	$\mathbf{10.04 \pm 0.01}$	$\mathbf{20.55 \pm 0.07}$	48.61 ± 0.05
End2End-GradNorm	4.43 ± 0.01	10.32 ± 0.07	9.68 ± 0.07	19.85 ± 0.38	46.95 ± 0.66
End2End-PcGrad	4.43 ± 0.01	10.35 ± 0.17	$\mathbf{10.04 \pm 0.01}$	20.22 ± 0.37	48.70 ± 0.04
$\mathcal{D}$(best)	$\mathbf{4.47 \pm 0.00}$	$\mathbf{10.47 \pm 0.00}$	9.21 ± 0.00	18.62 ± 0.00	45.11 ± 0.00
ParetoFlow	4.35 ± 0.08	10.41 ± 0.02	9.02 ± 0.20	18.73 ± 0.47	38.18 ± 1.24
PCD	4.42 ± 0.01	10.39 ± 0.03	9.38 ± 0.19	18.35 ± 0.37	39.59 ± 0.43
ARC (ours)	4.42 ± 0.00	10.39 ± 0.02	9.18 ± 0.14	17.90 ± 0.47	39.64 ± 0.50

Table 41: MO-NAS (part 2): hypervolume at the 50th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	c10mop6	c10mop8	c10mop9	in1kmop3	in1kmop4
MultiHead-GradNorm	68.84 ± 5.61	4.04 ± 0.06	7.72 ± 1.09	8.71 ± 0.19	4.06 ± 0.19
MultiHead-PcGrad	97.16 ± 4.36	4.30 ± 0.14	8.71 ± 0.26	10.04 ± 0.02	4.23 ± 0.14
MultiHead	102.49 ± 1.41	4.44 ± 0.19	9.57 ± 0.19	10.16 ± 0.02	$\mathbf{4.39 \pm 0.05}$
MultipleModels-ICT	103.43 ± 1.33	4.29 ± 0.07	8.76 ± 0.15	10.02 ± 0.04	4.24 ± 0.04
MultipleModels-IOM	103.73 ± 1.39	4.61 ± 0.07	$\mathbf{9.77 \pm 0.05}$	10.19 ± 0.01	4.11 ± 0.08
MultipleModels	103.74 ± 0.75	4.52 ± 0.16	9.40 ± 0.21	10.20 ± 0.03	$\mathbf{4.39 \pm 0.07}$
End2End	102.88 ± 2.29	4.49 ± 0.11	8.80 ± 0.14	$\mathbf{10.23 \pm 0.02}$	4.38 ± 0.09
End2End-GradNorm	88.45 ± 0.22	4.26 ± 0.08	8.42 ± 0.51	9.53 ± 0.19	4.22 ± 0.04
End2End-PcGrad	103.36 ± 0.23	4.51 ± 0.12	8.28 ± 0.05	10.12 ± 0.02	4.12 ± 0.07
$\mathcal{D}$(best)	$\mathbf{103.78 \pm 0.00}$	$\mathbf{4.70 \pm 0.00}$	9.64 ± 0.00	10.06 ± 0.00	4.15 ± 0.00
ParetoFlow	100.55 ± 1.44	4.20 ± 0.17	9.23 ± 0.30	9.65 ± 0.24	4.00 ± 0.13
PCD	101.04 ± 1.14	4.58 ± 0.04	9.26 ± 0.15	10.02 ± 0.03	4.10 ± 0.04
ARC (ours)	101.47 ± 2.40	4.55 ± 0.01	9.35 ± 0.10	10.10 ± 0.07	4.08 ± 0.02

Table 42: MO-NAS (part 3): hypervolume at the 50th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	in1kmop5	in1kmop6	in1kmop7	in1kmop8	in1kmop9
MultiHead-GradNorm	4.38 ± 0.02	8.61 ± 0.69	3.33 ± 0.60	7.72 ± 0.75	10.77 ± 2.10
MultiHead-PcGrad	4.38 ± 0.08	9.28 ± 0.09	4.18 ± 0.14	8.94 ± 0.11	18.99 ± 0.22
MultiHead	$\mathbf{4.52 \pm 0.02}$	9.59 ± 0.18	$\mathbf{4.45 \pm 0.05}$	9.46 ± 0.14	19.24 ± 0.25
MultipleModels-ICT	4.28 ± 0.05	9.25 ± 0.10	4.12 ± 0.07	8.78 ± 0.08	18.46 ± 0.57
MultipleModels-IOM	4.33 ± 0.04	9.13 ± 0.05	4.11 ± 0.28	$\mathbf{9.72 \pm 0.01}$	$\mathbf{20.07 \pm 0.08}$
MultipleModels	4.45 ± 0.07	$\mathbf{9.67 \pm 0.08}$	4.35 ± 0.03	9.43 ± 0.11	19.31 ± 0.44
End2End	4.48 ± 0.02	9.39 ± 0.08	4.40 ± 0.06	9.49 ± 0.08	19.54 ± 0.39
End2End-GradNorm	3.41 ± 0.08	9.06 ± 0.09	3.99 ± 0.21	8.43 ± 0.15	17.81 ± 0.21
End2End-PcGrad	4.30 ± 0.05	9.24 ± 0.08	4.25 ± 0.12	9.41 ± 0.07	19.48 ± 0.15
$\mathcal{D}$(best)	4.30 ± 0.00	9.15 ± 0.00	3.94 ± 0.00	9.28 ± 0.00	18.87 ± 0.00
ParetoFlow	4.05 ± 0.06	9.17 ± 0.11	3.54 ± 0.16	8.95 ± 0.39	18.27 ± 0.40
PCD	4.18 ± 0.12	8.98 ± 0.08	3.37 ± 0.13	8.82 ± 0.08	17.53 ± 0.57
ARC (ours)	4.26 ± 0.03	8.92 ± 0.19	3.66 ± 0.05	8.92 ± 0.07	18.39 ± 0.63

Table 43: MORL: hypervolume at the 50th percentile. Values are mean ± standard deviation (three seeds for Off-MOO-Bench baselines and five for generative methods); higher is better.

Method	mo_hopper_v2	mo_swimmer_v2
MultiHead-GradNorm	4.75 ± 0.00	0.97 ± 0.09
MultiHead-PcGrad	4.74 ± 0.00	0.86 ± 0.01
MultiHead	4.75 ± 0.00	1.28 ± 0.36
MultipleModels-ICT	4.75 ± 0.00	1.10 ± 0.16
MultipleModels-IOM	4.75 ± 0.00	0.90 ± 0.02
MultipleModels	5.00 ± 0.36	0.88 ± 0.00
End2End	5.01 ± 0.37	0.87 ± 0.00
End2End-GradNorm	4.84 ± 0.13	1.03 ± 0.16
End2End-PcGrad	4.75 ± 0.00	3.59 ± 0.02
$\mathcal{D}$(best)	$\mathbf{6.33 \pm 0.00}$	$\mathbf{3.70 \pm 0.00}$
ParetoFlow	5.40 ± 0.31	2.16 ± 0.27
PCD	4.84 ± 0.06	2.50 ± 0.07
ARC (ours)	4.84 ± 0.02	2.54 ± 0.04

H ARC WITH THE FLOW-MATCHING BACKBONE

This appendix reports the ARC-ParetoFlow block described in Appendix E. All 48 tasks pass the stock-sampler fidelity gate, with five seeds per arm. Hypervolume is reported at the 100th, 75th, and 50th percentiles under the same symmetric evaluation path. Every paired arm uses the same checkpoint, initial state, number of integration steps, batch and offspring construction, and network-evaluation budget. We report two-sided paired Wilcoxon signed-rank p-values across tasks.

The pattern transfers in direction: ARC has its largest median improvement at the 50th percentile ($+0.51\%$, with 31 of 48 tasks improved), while its median frontier effect is negative (-0.11%). However, none of the twelve comparisons is statistically significant at the 0.05 level, so the evidence on this backbone remains suggestive rather than established. Scoring candidates using the raw request rather than the predicted landing is also indistinguishable.

Appendix D.7 examines whether the weaker effect is related to the available steering window. ParetoFlow activates its direct multi-objective predictor guidance only after $t > 0.8$, although its weight-dependent filtering and neighboring-evolution mechanisms operate throughout sampling. Restricting ARC on PCD to a comparable late intervention window removes most of its gain. These results suggest that the shorter direct-guidance window contributes to the weaker flow result, while differences in ParetoFlow's filtering and evolution mechanisms may also matter. The experiment supports the architectural portability of the intervention and a recurring 50th-percentile-weighted improvement pattern, but it does not establish a backbone-independent performance improvement.

Table 44: ARC on ParetoFlow over 48 fidelity-qualified tasks and five seeds. Positive values favour the first arm. p-values are from two-sided paired Wilcoxon signed-rank tests across tasks.

Comparison	pct	mean	median	W/L	p
ARC vs. open loop	100th	-1.63%	-0.11%	17/30	0.179
	75th	$+3.38\%$	$+0.01\%$	24/23	0.472
	50th	$+3.89\%$	$+0.51\%$	31/16	0.072
Archive vs. random dest.	100th	$+0.00\%$	$+0.01\%$	25/21	0.797
	75th	$+10.07\%$	$+0.01\%$	24/22	0.674
	50th	$+3.18\%$	$+0.09\%$	26/20	0.441
Uniform vs. ranked sources	100th	$+1.82\%$	$+0.09\%$	29/18	0.078
	75th	$+7.70\%$	-0.08%	21/25	0.848
	50th	$+0.86\%$	$+0.06\%$	26/20	0.530
Elite protection vs. none	100th	$+0.18\%$	$+0.07\%$	28/20	0.182
	75th	$+0.18\%$	$+0.01\%$	25/22	0.767
	50th	-0.55%	$+0.03\%$	25/22	0.526