ADAPTIVE EVOLUTIONARY PARETO SEARCH WITH ONLINE TRANSFERABILITY ESTIMATION FOR RESOURCE-AWARE LLM ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting large language models to changing tasks and deployment requirements requires balancing accuracy, latency, and memory under limited evaluation resources. Although parameter-efficient fine-tuning (PEFT) reduces adaptation cost, selecting which layers to adapt together with rank and dropout creates a combinatorial multi-objective search problem. Under a fixed evaluation budget, reusing configurations from previous tasks can further waste evaluations when their usefulness on the new target is unknown. We present Adaptive Evolutionary Pareto Search with Online Transferability Estimation (AEPS-OT), a framework that allocates a fixed evaluation budget across multiple proposal mechanisms and historical configurations. A controller uses the current search state and objective weights to select among genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO), and estimation of distribution algorithm (EDA) proposals, while a Pareto archive preserves search quality. To reuse experience from prior tasks, AEPS-OT first probes historical configurations on the target task and estimates transferability separately for accuracy, latency, and memory from rank agreement, stability, and performance gains measured on the target task. The resulting scores, combined using the current deployment preference, determine bounded source quotas, with all probes and transfers charged to the same 1,000-slot target budget. On Qwen2-0.5B and the Internet Movie Database (IMDb) dataset, five-seed independent audits give a mean hypervolume of 0.193424, higher than all fourteen external controls, including Bayesian optimization, fixed evolutionary operators, policy-learning baselines, classical multi-objective evolutionary algorithms and randomized search. In terms of mean audited hypervolume, AEPS-OT outperforms the base Adaptive Evolutionary Pareto Search (AEPS) by 4.4%. Search using target probes without subsequently adding historical configurations reaches 0.192410, indicating that evaluating historical configurations on the target task accounts for most of the observed transfer gain. These results support evaluating historical configurations on the current task before reusing them in resource-aware PEFT search under the evaluated single-model setting with one target task and four historical source tasks.

1 INTRODUCTION

Resource-aware adaptation of large language models (LLMs) requires selecting configurations that balance accuracy, latency, and memory under a finite evaluation budget. Parameter-efficient fine-tuning (PEFT) makes this trade-off explicit. Adapter placement, rank, and regularization jointly determine task performance and deployment cost (Hu et al., 2022; Dettmers et al., 2023), while every candidate must still be instantiated, fine-tuned, validated, and measured on the target workload. The resulting problem is a budget-constrained Pareto search in which each evaluation is both informative and expensive.

LLM adaptation is also a repeated decision problem. Organizations often adapt the same model family across tasks, data domains, and deployment preferences, creating a growing history of evaluated configurations. This history can guide a new search, yet its value is target dependent. A configuration that ranks highly on one task may rank differently on another, and transferability can vary across accuracy, latency, and memory. Deployment preferences further determine which of these objective-

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.

specific relationships matter. Consequently, assigning an evaluation slot to historical experience carries an opportunity cost. Effective search therefore requires evidence about when historical knowledge is worth the budget, which knowledge should be reused, and how much budget it should receive.

We formulate resource-aware LLM adaptation as an evaluation budget allocation problem across tasks and deployment preferences. At each stage, the search allocates a shared ledger between discovering target configurations and exploiting historical experience whose relevance must be established on the current target. This perspective leads to AEPS-OT, an adaptive evolutionary Pareto search framework that integrates preference-conditioned exploration with experience reuse guided by evaluations on the current task. AEPS-OT evaluates a small set of historical configurations alongside random target anchors, estimates objective-specific transferability from rank agreement, stability, and gains measured on the target task, and turns this evidence into source quotas under the active deployment preference. All probes, transferred configurations, and newly proposed candidates consume the same target evaluation budget.

Within this allocation framework, the preference-conditioned controller chooses among complementary GA, DE, PSO, and EDA proposal mechanisms, and a Pareto archive based on measured objectives preserves the evolving Pareto front. These mechanisms support discovery of new configurations on the target task. The central transfer decision is governed by evidence collected on the target. The resulting search process values prior PEFT evaluations against new target evaluations using evidence collected on the current target.

The contributions are threefold:

- We formulate resource-aware LLM adaptation as a budget-constrained Pareto search problem, where limited and expensive evaluation resources are dynamically allocated between discovering new configurations and exploiting historical experience. From this perspective, we introduce AEPS-OT, an adaptive evolutionary Pareto search framework that integrates preference-conditioned evolutionary exploration with experience reuse guided by evaluations on the current task for multi-objective LLM adaptation.
- We propose a transfer mechanism that uses evaluations on the current task to decide which historical PEFT configurations receive further slots during online Pareto search. By estimating objective-specific transferability under the current task and deployment preference, the method assigns zero credibility to source–objective pairs with negative rank agreement in the evaluated setting.
- We evaluate Qwen2-0.5B adaptation to IMDb with four historical source tasks and five paired seeds. Independent full-front audits show that AEPS-OT has the highest mean HV among the evaluated controls under a fixed target budget. The reuse ablation attributes most of its mean gain over AEPS to probing historical configurations on the target.

2 Related Work

PEFT configuration search. Low-rank adaptation (LoRA) learns low-rank updates in selected modules (Hu et al., 2022).Quantized low-rank adaptation (QLoRA) combines low-rank tuning with quantized base models (Dettmers et al., 2023). Adaptive budget allocation for parameter-efficient fine-tuning (AdaLoRA) reallocates a parameter budget during candidate training (Zhang et al., 2023). Automatic configuration search for parameter-efficient fine-tuning (AutoPEFT) searches PEFT module combinations against task performance and parameter cost, including cross-task reuse (Zhou et al., 2024). Lawton et al. (2023) learn PEFT architectures through structured and unstructured pruning. Budget-guided iterative search for parameter-efficient fine-tuning of large pretrained language models (BIPEFT) incorporates parameter budgets into an iterative search over PEFT modules and ranks (Chang et al., 2024). AEPS searches adapter placement, rank, and dropout using measured accuracy, request latency, and inference memory. Its transfer extension evaluates historical configurations on the target before assigning them further search slots.

Multi-objective evolutionary search. Multi-objective evolutionary algorithms (EAs) such as non-dominated sorting genetic algorithm II (NSGA-II), strength Pareto evolutionary algorithm 2 (SPEA2), multi-objective evolutionary algorithm based on decomposition (MOEA/D), and S-metric selection

evolutionary multi-objective algorithm (SMS-EMOA) use non-dominance, strength, decomposition, and hypervolume contribution to guide search (Deb et al., 2002; Zitzler et al., 2001; Zhang & Li, 2007; Beume et al., 2007). Hypervolume summarizes convergence and coverage relative to a fixed reference point (Zitzler & Thiele, 1999). AEPS combines Pareto selection on measured objectives with learned allocation across proposal generators.

Preference-conditioned control and adaptive operators. Preference-conditioned multi-objective reinforcement learning (RL) adapts value estimates to changing objective weights (Abels et al., 2019). Dynamic algorithm configuration learns to adjust algorithm parameters to the current optimization state (Biedenkapp et al., 2020). Multi-agent dynamic algorithm configuration (MA-DAC) uses cooperative RL to configure multiple hyperparameter types of a multi-objective optimizer during search (Xue et al., 2022a). In evolutionary search, bandit and value-based methods adapt operator selection (Song et al., 2024; Sun & Li, 2020; Tian et al., 2023). AEPS instead selects generator/profile actions using PEFT candidate structure, Pareto-archive state, and measured deployment objectives.

Sequential experience transfer. Evolutionary transfer optimization studies how knowledge from previously solved problems can improve subsequent optimization tasks (Tan et al., 2021). Evolutionary sequential transfer optimization specializes this idea to sequential problems and solution reuse (Xue et al., 2022b). AEPS-OT applies solution-level reuse to PEFT search: target probes measure whether source rankings remain informative for each objective, and evidence combined using the current preference allocates subsequent historical proposals within the target budget.

3 Problem Setting

For a fixed language model and adapter type, we define a configuration as follows:

$$x = (L, r, p) \in \mathcal{X}, \qquad 2 \le |L| \le 6, \tag{1}$$

where L is a subset of the 24 transformer layers, $r \in \{4, 8, 16, 32\}$, and $p \in \{0, 0.05, 0.10, 0.15\}$. Rank and dropout are global scalars shared by all selected layers and adapted modules. Fine-tuning and deployment evaluation yield the objective vector:

$$y(x) = \big(A(x), T_{95}(x), M(x)\big), \tag{2}$$

where A is search-validation accuracy, T_{95} is warm end-to-end 95th-percentile (P95) latency per batch-one request, and M is the inference working set. Pareto dominance maximizes A and minimizes T_{95} and M.

For HV, the evaluator maps raw measurements to higher-is-better coordinates

$$\begin{aligned} g(x) &= (g_A(x), g_T(x), g_M(x)), \\ g_A(x) &= \mathrm{clip}(A(x), 0, 1), \\ g_T(x) &= \mathrm{clip}\left(1 - \frac{T_{95}(x)}{R_T + 10^{-8}}, 0, 1\right), \\ g_M(x) &= \mathrm{clip}\left(1 - \frac{M(x)}{R_M + 10^{-8}}, 0, 1\right), \end{aligned} \tag{3}$$

with T_{95} in seconds, M in MiB, $R_T = 1.0$ second, and $R_M = 2048$ MiB. The small constant uses the units of its denominator. Values outside these bounds are clipped to $[0, 1]$ for HV. Raw measurements determine Pareto dominance. Both HV indicators use reference point $(0, 0, 0)$ in this coordinate space. $\mathrm{HV}_{\text{search}}$is computed from the final search-validation archive. For $\mathrm{HV}_{\text{audit}}$, we retrained every point on its non-dominated front three times. Held-out objective vectors are averaged, non-dominance is recomputed, and HV is then computed. Held-out measurements enter only the final audit. This full-front audit reduces dependence on the adaptively reused search-validation measurements.

4 Method

AEPS-OT allocates a fixed target-evaluation budget between newly generated configurations and historical configurations assessed on the target. The adaptive controller selects how to generate new proposals. The transfer estimator assigns eligible historical proposals a bounded share of each batch. Both feed the same evaluator and Pareto archive (Figure 1). Algorithm 1 gives the search procedure.

4.1 Adaptive Evolutionary Pareto Search

AEPS maintains a population P_t and an external archive $\mathcal{A}_t$ of unique successful evaluations, whose non-dominated subset is $\mathcal{F}_t$. Each update selects parents, generates and evaluates proposals, and updates the population and archive. Raw accuracy, latency, and memory define Pareto dominance. A contextual preference guides which improvements the controller pursues.

Context and joint actions. For parent x, the state $s_t(x)$ combines its objectives and structure with archive quality, search progress, diversity, and current preference weights. The preference increases the weight of unmet deployment targets:

$$w_{\mathrm{t,j}} = \frac{w_{0,\mathrm{j}}(1 + \eta d_{\mathrm{t,j}})}{\sum_k w_{0,\mathrm{k}}(1 + \eta d_{\mathrm{t,k}})}, \tag{4}$$

where w_0 is the base preference and $d_{\mathrm{t,j}}$ measures the current front's deficit on objective j. The action space contains $4 \times 3 = 12$ joint actions: each selects a proposal generator (GA, DE, PSO, or EDA) and an exploration profile (explore, balanced, or exploit). The profiles control additional mutation probability and duplicate-repair effort. Appendix B specifies the state and operators.

Objective credit and policy update. The actor samples from $\widetilde{\pi}_\theta(a \mid s) = (1-\epsilon_t)\pi_\theta(a \mid s) + \epsilon_t/12$. Three critics estimate objective-specific immediate rewards. For a successful nonduplicate proposal x' from parent x, each reward combines parent-relative improvement with archive gain and diversity:

$$r_j = \mathrm{clip}\left(\Delta g_j + \lambda_H \Delta \mathrm{HV}(x') + \lambda_\nu \nu + \lambda_\zeta \zeta, -1, 1\right), \tag{5}$$

where ν is the nearest structural distance to an archived configuration, and ζ is the fraction of archive members dominated by the child minus the fraction dominating it. Resource improvements within measurement deadbands receive zero credit. Duplicate proposals and failed evaluations receive fixed penalties. The policy combines the three critic residuals by using the active preference, $\widehat{A} = \mathrm{standardize}(\sum_j w_{\mathrm{t,j}}[r_j - V_j(s)])$, and applies a clipped policy update (Schulman et al., 2017) using the stored mixture-policy probabilities. The update uses the current generation's target transitions and clears their buffer afterward. Appendix B gives the loss, reward scales, and training settings.

Population and budget. Parent tournaments combine fitness computed using the current preference weights with structural novelty. Survivors combine elites and offspring subject to a duplicate cap. The archive retains unique successful evaluations. Initialization and each offspring batch receive N slots, for a total of NG slots over G generations. Historical proposals replace slots within a batch, and the controller fills the remainder. Cache hits and duplicate handling are recorded alongside the nominal slot count.

4.2 Online Transferability Estimation by Objective and Preference

The estimator uses a small set of target evaluations to determine which historical sources are useful under the current objective preference. It combines source–target rank agreement, stability under probe deletion, and improvement over random target anchors.

Source memory and target probes. Source memory contains search-validation Pareto configurations, an archive sample for probe diversification, and evaluation metadata. Each configuration stores relative layer positions, rank, and dropout. Retrieval matches the evaluation protocol and seed stream, then maps configurations to the target search grid. Up to four probes per source cover objective extremes and diverse source-objective vectors. Initialization assigns at most sixteen of its forty slots to probes and the remainder to random anchors. Colliding target structures share one evaluation while preserving source attribution. Appendix B details memory contents and probe planning.

Rank agreement and evidence strength. For source s and objective $j \in \{A, T, M\}$, let n_{sj} denote the number of observed source–target probe pairs. Objective signs make larger values better. Measurement deadbands define tied ranks. The estimator shrinks Spearman rank agreement ρ_{sj} according to the available observations:

$$c_{\mathrm{sj}} = \frac{n_{\mathrm{sj}}}{n_{\mathrm{sj}} + 3} \max(0, \rho_{\mathrm{sj}}). \tag{6}$$

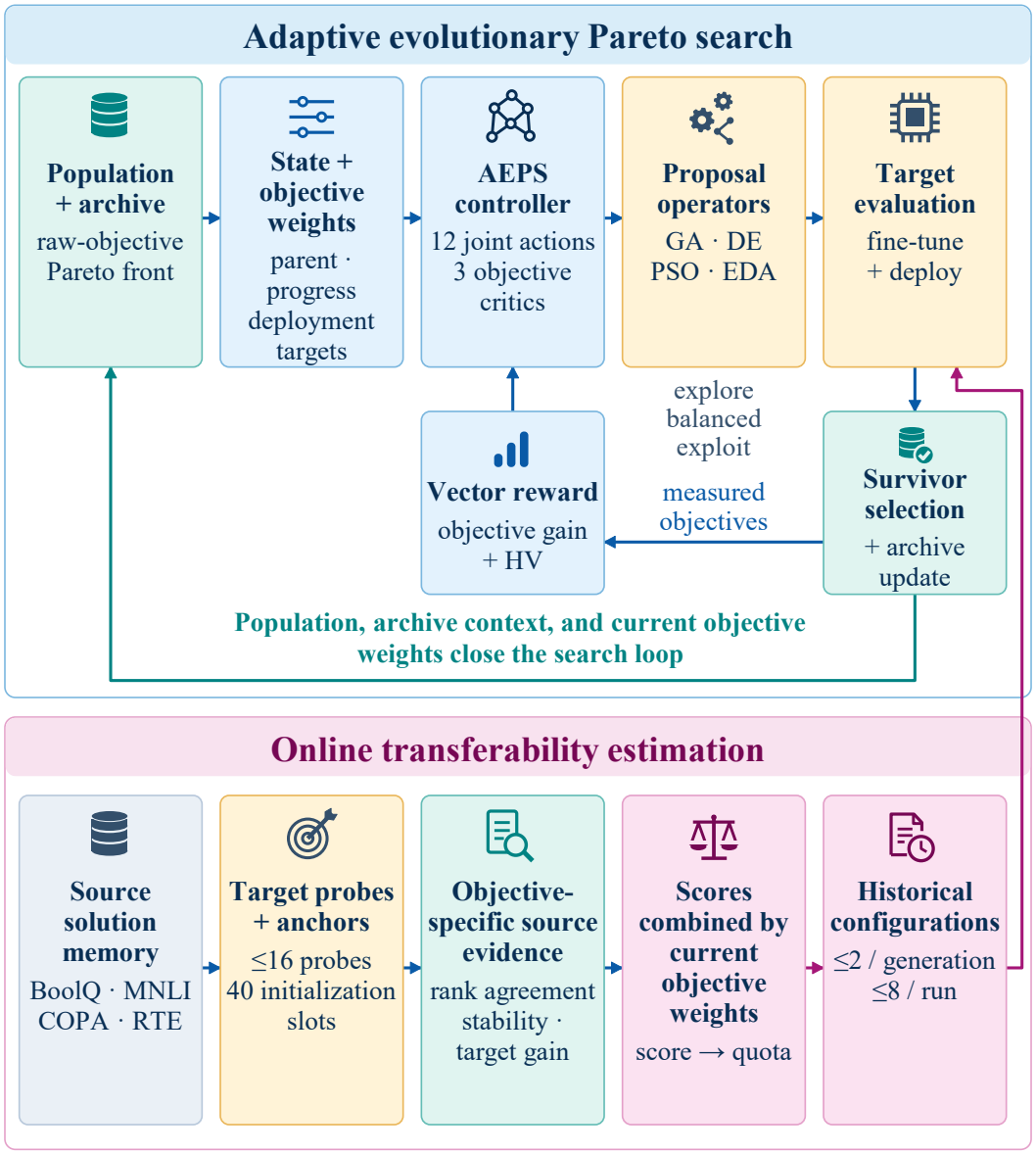

Figure 1: **Adaptive evolutionary Pareto search with online transferability estimation.** The upper branch performs adaptive Pareto search, while the lower branch estimates target-side transferability and allocates bounded historical proposals under the shared evaluation budget.

It further discounts rankings that change sharply when one probe is deleted:

$$e_{\mathrm{sj}} = \frac{n_{\mathrm{sj}}}{n_{\mathrm{sj}}+3}\,\mathrm{clip}\left(1-\frac{\max\rho_{\mathrm{sj}}^{(-i)}-\min\rho_{\mathrm{sj}}^{(-i)}}{2},0,1\right). \tag{7}$$

This expression applies with at least four pairs and two identifiable leave-one-out correlations. Otherwise, $e_{\mathrm{sj}} = [n_s/(n_s+3)]^2$, where n_s counts observed probe pairs for source s. Undefined or negative rank agreement receives zero credibility. Appendix B.1 gives the small-sample rules.

Target utility and source selection. The gain term compares target outcomes of source probes S_{sj} with random anchors R_j:

$$u_{\mathrm{sj}} = \mathrm{clip}\left(\frac{\mathrm{mean}(S_{\mathrm{sj}})-\mathrm{mean}(R_j)}{\max(\mathrm{sd}(R_j),d_j)},-1,1\right), \tag{8}$$

Algorithm 1 AEPS-OT under a shared target-evaluation budget

Require: Search space $\mathcal{X}$, source memory, $N = 40$, $G = 25$, base preference w_0
Retrieve eligible source records and map them to $\mathcal{X}$
Select up to four probes per source, with at most sixteen probe slots
Fill the remaining initialization slots with random anchors
Evaluate the N initialization slots with cache and duplicate accounting
Initialize P_0, archive $\mathcal{A}_0$, actor, and three critics
Record source–target probe pairs and target-anchor outcomes
Set cumulative transfer count $b \leftarrow 0$
for $t = 0, \ldots, G-2$ **do**
    Compute w_t and exploration rate ϵ_t from the current search state
    Recompute $q_s(w_t)$ from probe and anchor evidence
    Apportion at most $\min(2, 8-b)$ slots among sources with positive scores
    Select eligible historical configurations C_t; redistribute unfilled quotas
    Initialize ordinary offspring list $O_t \leftarrow [\,]$
    for each of the $N - |C_t|$ remaining slots **do**
        Select parent x by preference and novelty tournament
        Construct $s_t(x)$ and sample a generator/profile action from $\widetilde{\pi}_\theta$
        Generate, decode, mutate, and repair the proposal; append it to O_t
        Store the parent, state, action, and mixture-policy probability
    end for
    Evaluate O_t and C_t, charging all N nominal slots
    Record target observations and update $\mathcal{A}_{t+1}$, retaining the pre-batch archive
    Assign vector credit to controller proposals using the pre-batch archive
    Update the actor and critics from the current target transitions; clear the transition buffer
    Select survivor population P_{t+1} and update $b \leftarrow b + |C_t|$
end for
return Final non-dominated front for independent full-front audit

where d_j is the objective deadband. The source score combines the three terms under the current preference:

$$q_s(w_t) = \sum_j w_{t,j}\, c_{sj}\, e_{sj}\, \max(0, u_{sj}). \tag{9}$$

A positive score requires identifiable positive rank agreement and a gain on the target task for an objective with nonzero weight. Thus, sources receive credit for preserving useful rankings on objectives that matter to the active deployment preference.

Transfer allocation. Largest-remainder apportionment distributes slots among sources with $q_s(w_t) > 0$. At most two historical configurations enter a generation and at most eight enter a run. Candidate selection excludes structures already evaluated on the target and redistributes unfilled quotas to eligible sources. Scores are recomputed under the current preference using the designated probe pairs and initialization anchors. This connects the source decision to target evidence while keeping transfer within the shared budget.

5 Experiments

5.1 Experimental Setup

Model, data, and search space. We use Qwen2-0.5B (Yang et al., 2024) with LoRA on `q_proj` and `v_proj`. IMDb (Maas et al., 2011) is the target. Disjoint stratified sets of 128 search-training and 128 search-validation examples come from the official training split. 1,000 held-out examples come from the official test split. Each candidate trains for eight epochs with bfloat16 (BF16), batch size 32, Adam with decoupled weight decay (AdamW), learning rate 2×10^{-4}, and maximum sequence length 256. The search selects 2–6 of 24 layers, rank in $\{4, 8, 16, 32\}$, and dropout in $\{0, 0.05, 0.10, 0.15\}$ with $\alpha = 8$.

Source tasks, seeds, and budget. Source records come from Boolean Questions (BoolQ), Multi-Genre Natural Language Inference (MNLI) (Wang et al., 2018), Choice of Plausible Alternatives

(COPA) (Wang et al., 2019), and Recognizing Textual Entailment (RTE) under the same model and evaluator. The target cohort uses seeds $\{1, 2, 5, 23, 71\}$. Each run has population 40 and 25 generations, giving 1,000 nominal candidate slots. The primary preference is $(0.6, 0.2, 0.2)$ for accuracy, latency, and memory. Probes and transfers consume slots within this budget.

Deployment and audit. The evaluator uses label negative log-likelihood (NLL) requests with a batch size of one. It processes 32 prompts, performs five warmup runs, and takes 100 measurements grouped into ten blocks. Latency includes tokenization, host-to-device transfer, unmerged-LoRA inference, device-to-host transfer, and prediction parsing. Memory measures the base-model-plus-adapter inference working set through the Compute Unified Device Architecture (CUDA) caching allocator after subtracting pre-model background. The independent audit retrains every search-validation Pareto configuration three times on the search-training set with independent training seeds. Accuracy is then evaluated on the 1,000 held-out examples, while latency and memory use the same deployment workload as search. Averaged retrain objective vectors define the audited front. Error bars and $\pm$ values report sample standard deviation (SD) across the five algorithm seeds.

Internal controls. AEPS is the base search condition, run without source records. Source records are stored configurations from prior tasks. AEPS-OT extends AEPS by probing these configurations on the target task and allocating search slots to eligible historical configurations according to their estimated transferability. Static reuse places source configurations in 20% of the initial population. Uniform source sampling selects eligible source records uniformly. Selection using offline diagnostics relies on a historical diagnostic report. The condition without later historical transfer evaluates target probes and anchors, then fills subsequent slots with ordinary proposals. AEPS-OT also combines source scores using the current preference weights and adds selected historical configurations.

External controls. The comparison includes thirteen search controls and one AdaLoRA configuration-search control. The search controls are fixed GA, DE, EDA, and PSO; Multi-Critic Parent Selection and ordinary proximal policy optimization (PPO); four classic multi-objective algorithms; random search; uniform-random AEPS; and Gaussian-process Bayesian optimization (GP-BO). GP-BO is a surrogate-based optimization control: a Gaussian-process surrogate guides candidate selection under the same nominal target-evaluation budget. Multi-Critic Parent Selection learns random-mate, Pareto-mate, or tournament parent selection with three objective critics. Its state contains the parent's three directional objective coordinates and rank. It then applies fixed crossover and mutation. Ordinary PPO uses the same state and actions with scalarized reward. AEPS selects the proposal generator and exploration profile using parent, archive, and preference context. These are system-level controls with distinct states and reward rules. The uniform-random AEPS control retains AEPS parent selection, contextual preference, repair, evaluation, and survivor selection, but samples uniformly from the 12 joint operator-profile actions without updating a controller. NSGA-II, SPEA2, MOEA/D, and SMS-EMOA use the pymoo implementations (Blank & Deb, 2020). All runs share the evaluator, five seeds, and 1,000 nominal target slots. The thirteen search controls share the LoRA search grid. AdaLoRA uses the same layer/dropout choices, initial rank 32, and target-rank choices $\{4, 8, 16, 32\}$; its training procedure reallocates rank.

Hardware and reporting. All experiments ran on NVIDIA GeForce RTX 5090 graphics processing units (GPUs) with nominal 32 gigabytes (GB) of memory and CUDA 13.0. Tables report mean and sample SD across the five paired seeds. Appendix A lists the complete software and evaluation settings; Appendix F reports source, search, and audit time separately.

5.2 Comparison with Budget-Matched Search Baselines

AEPS-OT has the highest mean $\mathrm{HV}_{\mathrm{audit}}$among the fourteen baselines in Table 1. GP-BO is closest by mean $\mathrm{HV}_{\mathrm{audit}}$. DE is the strongest fixed-operator baseline. These results compare complete search systems under a common evaluator and target budget. The comparison measures their combined proposal, selection, and reuse mechanisms; the matched AEPS–Uniform AEPS comparison below isolates learned action selection.

Effect of learned action selection. Uniform AEPS retains the same parent selection, contextual preference, repair, evaluator, and survivor selection as AEPS, but samples uniformly from the twelve

Table 1: Budget-matched baseline comparison. Values are mean ± sample SD of HV_{audit}over five paired seeds. Higher is better. Bold marks the highest mean.

Category	Method	HV_{audit}
Adaptive search	AEPS-OT	$\mathbf{0.193424 \pm 0.002855}$
Bayesian optimization	GP-BO	0.192652 ± 0.001413
Fixed operators	GA	0.184993 ± 0.006570
	DE	0.185438 ± 0.003615
	EDA	0.185185 ± 0.005551
	PSO	0.185210 ± 0.005700
Policy baselines	Multi-Critic Parent Selection	0.183157 ± 0.003852
	PPO	0.183725 ± 0.002581
Multi-objective EAs	NSGA-II	0.181265 ± 0.003216
	MOEA/D	0.179264 ± 0.003793
	SPEA2	0.177867 ± 0.001663
	SMS-EMOA	0.179342 ± 0.004106
Randomized search	Random search	0.182648 ± 0.005195
	Uniform AEPS	0.180778 ± 0.002505
PEFT baseline	AdaLoRA	0.150998 ± 0.000339

Table 2: Experience-reuse ablation over five paired seeds. Both HV indicators summarize the full Pareto front. Values are mean ± sample SD.

Condition	HV_{search}	HV_{audit}
AEPS	0.194649 ± 0.003083	0.185317 ± 0.006825
Static reuse	0.194295 ± 0.004332	0.184554 ± 0.005642
Uniform source sampling	0.196090 ± 0.003375	0.186776 ± 0.003580
Offline diagnostic selection	0.196088 ± 0.003137	0.188013 ± 0.003704
Probes without transfer	0.201538 ± 0.001298	0.192410 ± 0.003938
AEPS-OT	$\mathbf{0.202281 \pm 0.001010}$	$\mathbf{0.193424 \pm 0.002855}$

joint generator/profile actions without updating a controller. AEPS has higher mean HV_{audit}(0.185317 versus 0.180778). This comparison separates learned action selection from the diversity benefit of mixing generators.

5.3 Experience-Reuse Ablation

Evaluating historical configurations on the target task accounts for most of the observed improvement over AEPS (Table 2). The condition that evaluates target probes without later transfer closes most of the gap to AEPS-OT, whereas static reuse and the controls without online selection provide smaller gains.

AEPS-OT improves mean HV_{audit}over AEPS by 0.008106 and over static reuse by 0.008870. Its mean $HV_{search}$is also higher than AEPS's (Table 2).

The independent full-front audit preserves the top two internal conditions by mean HV. AEPS-OT ranks first and probes without transfer second under both $HV_{search}$and HV_{audit}. Across the five seeds, the mean decreases from search to audit by 0.008857 for AEPS-OT and 0.009332 for AEPS. The search-to-audit differences jointly reflect held-out evaluation, independent retraining, and repeated deployment measurements.

5.4 Transfer Decisions

The estimator records objective-dependent source usefulness. Across five runs, ten source–objective pairs have negative rank agreement and receive zero credibility. Each run observes 14–16 source

probes and 22–24 unique random anchors after deduplication, identifies at least one source with positive evidence after applying the current preference weights, and uses the eight-slot transfer cap.

Source allocation varies across seeds: RTE and BoolQ receive most historical slots, MNLI receives four, and COPA receives none. Appendix C.1 gives the allocation for each seed. These outcomes show how evaluations on the target task select sources within the fixed transfer cap, which occupies 0.8% of the nominal target budget. Appendix E illustrates the search trajectory and operator allocation in seed 71.

6 Discussion and Limitations

The baseline comparison and reuse ablation address complementary aspects of the proposed framework. Under the evaluated budget, AEPS-OT achieves the highest mean audited HV among the compared methods, while target-side evaluation provides the evidence used to assess the usefulness of historical configurations. The estimator then combines objective-specific transferability evidence with the current deployment preference to allocate source quotas.

The present study focuses on a controlled Qwen2-0.5B-to-IMDb setting with a fixed target-evaluation budget, fixed hardware, and five paired seeds. This design enables matched evaluation accounting and consistent comparison of accuracy, latency, and memory. Future work can extend the framework across additional tasks, model scales, hardware platforms, and PEFT search methods.

Historical reuse is fixed at eight configurations per run, providing a controlled setting for evaluating the contribution of historical configurations. Future extensions can study adaptive transfer budgets and more flexible reuse policies.

7 Conclusion

AEPS allocates PEFT configuration evaluations across a learned proposal portfolio while preserving accuracy, latency, and memory trade-offs in a Pareto archive. AEPS-OT extends this allocation to historical configurations using evidence for each objective from target evaluations and source allocations determined by the current preference. Across five paired Qwen2-0.5B-to-IMDb runs, it has the highest mean audited HV among the external baselines and improves over AEPS and the static reuse control. The reuse ablation further shows that target-side evaluation of historical configurations is a key component of effective reuse. These findings motivate assessing how historical PEFT configurations rank and perform on the current task before committing further search budget to reuse.

Reproducibility statement

Sections 3–5 and Appendix B detail the protocol and implementation. The submitted code/results folder contains the AEPS implementation, online-transfer runner, five source-memory files, per-seed and summary result tables, and 15 compact audit records.

AI use statement

LLMs were used for improving writing clarity, code checking and formatting. The authors take full responsibility for all parts of the manuscript.

References

Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 11–20. PMLR, 2019. URL https://proceedings.mlr.press/v97/abels19a.html.

Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007. doi: 10.1016/j.ejor.2006.08.008.

André Biedenkapp, H. Furkan Bozkurt, Theresa Eimer, Frank Hutter, and Marius Lindauer. Dynamic algorithm configuration: Foundation of a new meta-algorithmic framework. In *Proceedings of the Twenty-fourth European Conference on Artificial Intelligence*, pp. 427–434. IOS Press, 2020. doi: 10.3233/FAIA200122.

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

Aofei Chang, Jiaqi Wang, Han Liu, Parminder Bhatia, Cao Xiao, Ting Wang, and Fenglong Ma. BIPEFT: Budget-guided iterative search for parameter efficient fine-tuning of large pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7429–7440, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.437. URL https://aclanthology.org/2024.findings-emnlp.437/.

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.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115, 2023. doi: 10.52202/075280-0441. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1feb87871436031bdc0f2beaa62a049b-Abstract.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Neal Lawton, Anoop Kumar, Govind Thattai, Aram Galstyan, and Greg Ver Steeg. Neural architecture search for parameter-efficient fine-tuning of large pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8506–8515, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.539. URL https://aclanthology.org/2023.findings-acl.539/.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015/.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347. arXiv preprint arXiv:1707.06347.

Yanjie Song, Yutong Wu, Yangyang Guo, Ran Yan, Ponnuthurai Nagaratnam Suganthan, Yue Zhang, Witold Pedrycz, Swagatam Das, Rammohan Mallipeddi, Oladayo Solomon Ajani, and Qiang Feng. Reinforcement learning-assisted evolutionary algorithm: A survey and research opportunities. *Swarm and Evolutionary Computation*, 86:101517, 2024. doi: 10.1016/j.swevo.2024.101517.

Lei Sun and Ke Li. Adaptive operator selection based on dynamic Thompson sampling for MOEA/D. In *Parallel Problem Solving from Nature – PPSN XVI*, volume 12270 of *Lecture Notes in Computer Science*, pp. 271–284. Springer, 2020. doi: 10.1007/978-3-030-58115-2_19.

Kay Chen Tan, Liang Feng, and Min Jiang. Evolutionary transfer optimization—a new frontier in evolutionary computation research. *IEEE Computational Intelligence Magazine*, 16(1):22–33, 2021. doi: 10.1109/MCI.2020.3039066.

Ye Tian, Xiaopeng Li, Haiping Ma, Xingyi Zhang, Kay Chen Tan, and Yaochu Jin. Deep reinforcement learning-based adaptive operator selection for evolutionary multi-objective optimization. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(4):1051–1064, 2023. doi: 10.1109/TETCI.2022.3146882.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446/.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html.

Ke Xue, Jiacheng Xu, Lei Yuan, Miqing Li, Chao Qian, Zongzhang Zhang, and Yang Yu. Multi-agent dynamic algorithm configuration. In *Advances in Neural Information Processing Systems*, volume 35, 2022a. doi: 10.52202/068431-1465.

Xiaoming Xue, Cuie Yang, Yao Hu, Kai Zhang, Yiu-Ming Cheung, Linqi Song, and Kay Chen Tan. Evolutionary sequential transfer optimization for objective-heterogeneous problems. *IEEE Transactions on Evolutionary Computation*, 26(6):1424–1438, 2022b. doi: 10.1109/TEVC.2021.3133874.

An Yang et al. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671. arXiv preprint arXiv:2407.10671.

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

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

Han Zhou, Xingchen Wan, Ivan Vulić, and Anna Korhonen. AutoPEFT: Automatic configuration search for parameter-efficient fine-tuning. *Transactions of the Association for Computational Linguistics*, 12:525–542, 2024. doi: 10.1162/tacl_a_00662.

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

Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. Technical Report TIK-Report 103, ETH Zurich, 2001.

A Exact Experimental Protocol

Hardware and runtime. All experiments ran on NVIDIA GeForce RTX 5090 GPUs with nominal 32-GB memory and CUDA 13.0. The software stack uses Python 3.10.20, PyTorch 2.13.0+cu130, Transformers 4.46.3, and PEFT 0.13.2. Linux hosts expose 128 logical central processing units (CPUs), with 64 PyTorch intra-op and inter-op threads. These settings accompany the workload definition because end-to-end latency includes host processing.

Table 3: Experimental settings shared by the target runs.

Component	Setting
Model / target	Qwen/Qwen2-0.5B / IMDb
GPU / runtime	NVIDIA GeForce RTX 5090 (32 GB) / CUDA 13.0
Software	Python 3.10.20; PyTorch 2.13.0; Transformers 4.46.3; PEFT 0.13.2
Adapter	LoRA on `q_proj`, `v_proj`; $\alpha = 8$
Configuration	2–6 layers; rank $\{4, 8, 16, 32\}$; dropout $\{0, .05, .10, .15\}$
Fine-tuning	BF16; 8 epochs; batch 32; AdamW; learning rate 2×10^{-4}
Data	128 search-train / 128 search-validation / 1,000 held-out
Search	population 40; 25 generations; 1,000 nominal slots
Seeds	1, 2, 5, 23, 71
Preference	accuracy/latency/memory $= (0.6, 0.2, 0.2)$
Preference gain	$\eta = 0.35$
HV reference point	$(0, 0, 0)$ in evaluator-provided coordinate space
Targets	0.80 accuracy; 15 milliseconds (ms); 1600 mebibytes (MiB)
Deployment	label NLL; batch 1; 5 warmups; 100 measures; 10 blocks
Front audit	full validation front; 3 retrains/configuration
Probe budget	4/source; at most 16 initialization slots
Transfer budget	at most 2/generation; at most 8/run
Deadbands	$1/128$ accuracy; 0.2 ms; 8 MiB

B Implementation Details

State and generation index. Let $t = 0, \ldots, 24$ denote the internal generation index; saved history uses $g = t + 1$. Let H_t be Search HV, P_t the population, and $\mathcal{A}_t$ the successful unique archive. The five parent features are $(g_A, g_T, g_M, r/32, |L|/6)$. The nine archive features are:

$$\left[H_t,\ \mathrm{clip}\left(\frac{\Delta H_t}{0.02}, -1, 1\right),\ \min\left(\frac{|\mathcal{F}_t|}{|P_t|}, 1\right),\ U_t,\ \bar{g}_A, \bar{g}_T, \bar{g}_M,\ \frac{t}{24}, \epsilon_t\right], \tag{10}$$

where $\Delta H_t = H_t - H_{\text{t-1}}$ is temporal search-HV progress and U_t is the number of distinct population structures divided by $|P_t|$. Each $\bar{g}_j$ averages objective j over $\mathcal{A}_t$. The initial HV increment is zero. The implementation rounds parent and archive features to four decimals, then appends the three current preference weights.

Exploration schedule. The base rate is $b_t = 0.12 + 0.53(1 - t/24)$. When $t \geq 3$ and $H_t - H_{\text{t-3}} < 5 \times 10^{-4}$, set $b_t \leftarrow \max(b_t, 0.40)$. Otherwise, when $t \geq 1$ and $H_t - H_{\text{t-1}} > 0.01$, set $b_t \leftarrow \max(0.12, b_t - 0.10)$. Finally, $\epsilon_t = \mathrm{clip}(b_t, 0.12, 0.65)$. The completed runs disable the optional higher-level exploration controller and learned-preference module.

Structural distance and reward. For configurations $x = (L, r, p)$ and $z = (L_z, r_z, p_z)$, define the structural distance:

$$D(x, z) = 0.65\left(1 - \frac{|L \cap L_z|}{|L \cup L_z|}\right) + 0.20 \min\left(\frac{|\log_2 r - \log_2 r_z|}{3}, 1\right) + 0.15 \min\left(\frac{|p - p_z|}{0.15}, 1\right). \tag{11}$$

For the successful archive $\mathcal{A}^-$ before the offspring batch, define the two reward terms:

$$\nu(x') = \min_{z \in \mathcal{A}^-} D(x', z), \tag{12}$$

$$\zeta(x') = \frac{1}{|\mathcal{A}^-|} \sum_{z \in \mathcal{A}^-} \left[\mathbb{1}(x' \succ z) - \mathbb{1}(z \succ x')\right]. \tag{13}$$

For an empty archive, set $\nu = 1$ and $\zeta = 0$. Here $\succ$ denotes Pareto dominance based on measured objectives. The child's marginal HV gain is $\Delta\mathrm{HV}(x') = \mathrm{HV}(\mathcal{A}^- \cup \{x'\}) - \mathrm{HV}(\mathcal{A}^-)$; it is distinct from temporal progress ΔH_t. If parent and child have equal accuracy, define $b = \mathrm{clip}(\tanh m(x') - \tanh m(x), -1, 1)$. The implementation adds $(0.49/128)b$ to the accuracy reward. The label-NLL margin m breaks ties within less than one validation example. Resource deadbands take the larger of the fixed threshold and the parent's or child's recorded 95% interval half-width. A detected duplicate receives the fixed penalty even when repair yields a useful evaluated structure.

Mixed-discrete representation. Each configuration has 26 encoded coordinates, including 24 layer indicators, one rank-grid index, and one dropout-grid index. Decoding thresholds layer entries at 0.5 and rounds and clips categorical indices. Repair enforces two to six layers using ranked layer scores, with random tie-breaking.

Proposal operators. GA applies uniform coordinate crossover with probability 0.9 and mutation probability $1/26$ per coordinate. A layer mutation flips its indicator; categorical mutations choose another legal index. DE uses current/1/bin: $v = z + 0.65(z_a - z_b)$, with distinct population donor indices. Binomial crossover has probability 0.9 and forces one mutant coordinate. PSO uses cognitive and social coefficients 1.49618, velocity clipping at ± 4, and inertia decreasing linearly from 0.9 to 0.4. Layer bits are sampled from sigmoid velocities; categorical positions add velocity before decoding. Pareto dominance updates personal bests, with fitness computed using the current preference weights breaking incomparable cases. Repaired coordinates have their stored velocity multiplied by 0.25. EDA fits the top half of the preference-ranked population once per generation. Layer probabilities are $(\mathrm{hits} + 1)/(n_{\text{elite}} + 2)$; rank and dropout categories receive unit pseudocounts before sampling.

Profile mutation. When additional mutation is triggered, the implementation chooses layer, rank, or dropout uniformly. A layer mutation removes one layer with probability 0.5 when above the minimum; otherwise it adds an unselected layer when below the maximum. Rank and dropout mutations resample a different legal value. Duplicate repair repeats this one-step mutation up to the profile's stated limit.

Contextual preference and exploration profiles. The state has seventeen dimensions: five parent features, nine archive/progress features, and three preference weights. The base preference is $w_0 = (0.6, 0.2, 0.2)$ and the deficit gain is $\eta = 0.35$. For the current front $\mathcal{F}_t$, $d_{\mathrm{t,A}} = \mathrm{clip}((0.80 - \max_{x \in \mathcal{F}_t} A(x))/0.80, 0, 1)$. The latency and memory deficits are the fractions of front members exceeding 15 ms and 1600 MiB, respectively. Explore, balanced, and exploit apply an additional mutation with probabilities 0.55, 0.15, and 0.02 and allow six, three, and one duplicate-repair attempts. Survivor selection reserves a 20% elite allocation.

Reward scales and policy optimization. Equation 5 uses $\lambda_H = 6$, $\lambda_\nu = 0.05$, and $\lambda_\zeta = 0.04$. Duplicate proposals receive -0.08 per coordinate; failed evaluations receive -1. These fixed settings are shared across the AEPS transfer conditions. The clipped actor objective uses the stored mixture-policy probabilities:

$$\mathcal{L}_{\text{actor}} = -\mathbb{E}\Big[\min\Big(R_\theta \widehat{A}, \mathrm{clip}(R_\theta, 0.8, 1.2)\widehat{A}\Big)\Big], \quad R_\theta = \frac{\widetilde{\pi}_\theta(a \mid s)}{\widetilde{\pi}_{\theta_{\text{old}}}(a \mid s)}. \tag{14}$$

A smooth-L_1 critic loss has coefficient 0.5, and the entropy bonus has coefficient 0.04. The two-layer, 96-unit actor–critic network receives up to four optimization passes per update, with early stopping at approximate Kullback–Leibler (KL) divergence 0.03.

Source records and probe planning. After each source search, memory stores protocol metadata, search-validation Pareto configurations, and a small archive sample for probe diversification. Records contain configuration structures and search measurements; source examples, trained adapters, controller trajectories, and held-out outcomes are excluded. Retrieval requires a matching evaluation protocol and seed stream. Relative layer positions are mapped to the target model and projected onto the legal search grid. For each source, the planner selects up to four configurations by choosing objective extremes and then applying farthest-point diversification in source-objective space. Colliding target structures are evaluated once with their source attribution retained. Initialization uses at most sixteen source-probe slots and at least twenty-four random-anchor slots within the ordinary forty-slot budget; it is recorded as generation 1.

B.1 Small-Sample Transfer Evidence

The estimator signs objective values so that higher is better and assigns tied ranks to differences within one validation example for accuracy, 0.2 ms for latency, or 8 MiB for memory. The count $n_{\mathrm{sj}} = n_s$ records observed source–target probe pairs; identifiability is checked separately for each objective. Negative rank agreement is recorded as anti-transfer and contributes zero credibility, as does undefined agreement.

The leave-one-out range is a decision penalty for sensitivity to deleting a probe. It applies with at least four pairs and two identifiable deletion correlations. Undefined deletion correlations are excluded, and fewer than two valid values trigger $e_{\mathrm{sj}} = h_s^2$, where $h_s = n_s/(n_s + 3)$. At four probes, each deletion leaves three observations, making the score sensitive to ties. The constant 3 sets the shrinkage scale: $h_s = 1/2$ at three pairs and $4/7$ at four. With four probes, the combined credibility and evidence multiplier is at most $(4/7)^2$. Transfer exposure is bounded separately by the generation and run caps.

C Transfer Diagnostics

C.1 Observed Probe and Transfer Allocation

Table 4: Observed online-estimator activity. Anchors count unique observed random structures after deduplication.

Seed	Probes	Anchors	Transfers	Transfer allocation
1	16	22	8	BoolQ 4, RTE 4
2	16	22	8	BoolQ 4, RTE 4
5	16	22	8	BoolQ 4, MNLI 4
23	14	24	8	RTE 8
71	15	24	8	BoolQ 4, RTE 4

Across the five runs, ten source–objective pairs have negative rank agreement and contribute zero credibility to Equation 9.

D Audited-Front Sensitivity and Deployment Constraints

We recomputed HV from the independently audited non-dominated points in the exported run summaries for AEPS-OT, AEPS, Uniform AEPS, and DE. The default $(R_T, R_M) = (1.0\,\mathrm{s}, 2048\,\mathrm{MiB})$ calculation reproduces all 20 recorded $\mathrm{HV}_{\mathrm{audit}}$ values exactly. Table 5 varies only these normalization scales after audit. The higher-is-better reference point stays $(0, 0, 0)$, and no search or retraining changes. AEPS-OT remains first among these four methods by mean HV.

Table 5: Mean audited HV across five seeds under post-hoc normalization scales. Higher is better; the reported main result uses the default row.

(R_T, R_M)	AEPS-OT	AEPS	DE	Uniform AEPS
Stricter: $(0.5\,\mathrm{s}, 1800\,\mathrm{MiB})$	**0.108400**	0.100372	0.100054	0.097499
Default: $(1.0\,\mathrm{s}, 2048\,\mathrm{MiB})$	**0.193424**	0.185317	0.185438	0.180778
Looser: $(2.0\,\mathrm{s}, 2560\,\mathrm{MiB})$	**0.316828**	0.308690	0.309455	0.301740

Table 6 selects the best held-out accuracy from each audited front under a 15-ms batch-one P95 latency cap and a 1600-MiB inference working-set cap. Every audited-front point in these four methods meets both caps, so the values equal the unconstrained best-accuracy means. AEPS-OT has the highest mean under these deployment requirements.

Table 6: Best feasible held-out accuracy under the stated batch-one P95 inference-latency and inference-working-set caps. Values average five seeds.

Method	15 ms / 1600 MiB
AEPS-OT	0.810600
AEPS	0.802467
DE	0.805800
Uniform AEPS	0.785867

E Search Dynamics and Operator Allocation

Seed 71 is an illustrative diagnostic run and has the highest final HV_{audit} in the five-seed cohort. Generation 1 is initialization. Search HV rises from 0.1885 to 0.2015, with gains exceeding 0.001 at generations 2, 4, 5, and 15. EDA's share changes from $8/38 = 21.1\%$ in the first offspring batch to $19/40 = 47.5\%$ in the last, while the final batch assigns 20.0% to PSO, 17.5% to GA, and 15.0% to DE. Across all updates, EDA receives 30.8% of operator selections and DE receives 26.2%. Figure 2 pairs the HV trajectory with operator frequencies. The trajectories describe how operator allocation and Pareto quality evolve within this run.

Figure 3 shows the corresponding three-objective configuration set and cumulative Pareto front in every generation.

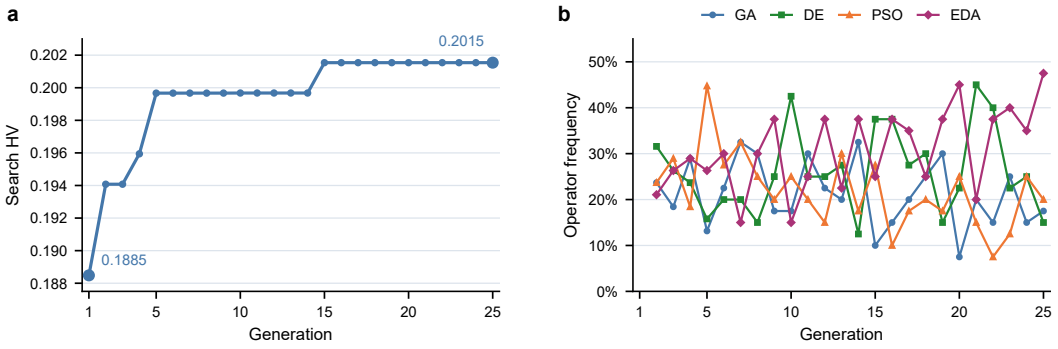

Figure 2: **Illustrative search dynamics of AEPS-OT seed 71. a,** Search HV from saved records. **b,** Operator frequencies from consecutive cumulative-counter differences, expressed as within-interval proportions. Generation 1 is initialization and has no operator frequency. The first four offspring batches contain 38 operator proposals and two transfers each. Later batches contain 40 operator proposals. Curves show unsmoothed observations from this selected run.

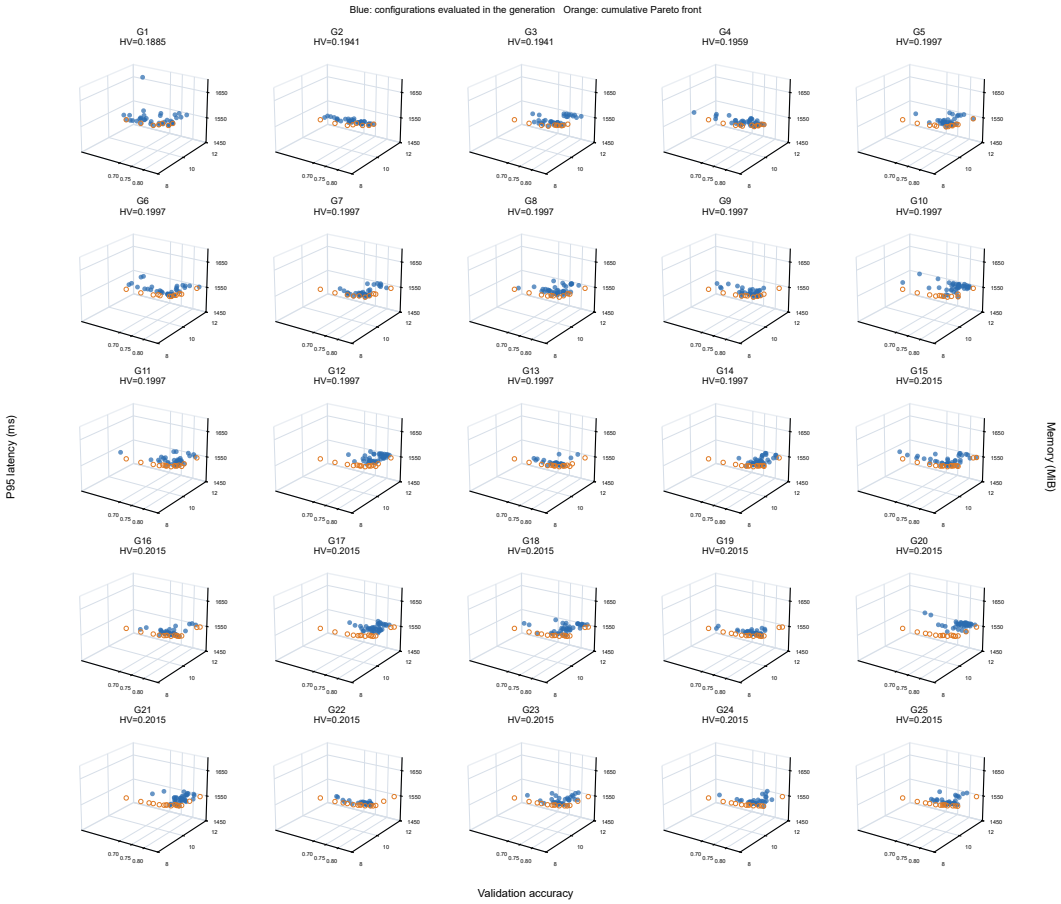

Figure 3: **Per-generation three-objective Pareto evolution of AEPS-OT seed 71.** Blue points are successful configurations evaluated in the indicated generation, and orange points are the cumulative Pareto front. Each panel reports the independently reconstructed search HV for that generation in the evaluator-provided higher-is-better coordinate space with reference point $(0, 0, 0)$.

F SOURCE AND TARGET COST ACCOUNTING

Wall-clock costs are measured on one NVIDIA GeForce RTX 5090 with CUDA 13.0. Table 7 reports four 1,000-slot source searches per algorithm seed. Source audit is disabled, so total time approximately equals search time. Run-log times for seeds 1, 2, 5, 23, and 71 are reported to four decimal places; rounding explains small discrepancies in the sums.

Table 7: One-time source-library construction cost, in hours (h). SD is across the five algorithm seeds.

Seed	BoolQ	MNLI	RTE	COPA	Total
1	2.2544	2.0505	2.0718	1.7127	8.0893
2	2.2363	2.1667	2.1480	1.6417	8.1927
5	2.0523	2.0214	1.9344	1.4309	7.4390
23	2.0478	1.8880	1.9643	1.4570	7.3571
71	2.0775	2.0876	1.9664	1.5737	7.7052
Total mean $\pm$ sample SD					7.7567 ± 0.3755

Table 8: IMDb target task-run time, in hours. The last two columns include search and audit; the final column includes target probing without later historical transfer.

Seed	AEPS-OT total	Search	Audit	AEPS	Without transfer
1	2.1586	2.0921	0.0665	2.2100	2.1672
2	2.0476	1.9649	0.0828	2.0978	1.9562
5	2.0471	1.9369	0.1102	2.0047	2.0768
23	1.8920	1.8365	0.0555	2.0056	2.0170
71	1.9299	1.8253	0.1045	1.7525	2.0612
Mean	2.0150	1.9311	0.0839	2.0141	2.0557
SD	0.1062	0.1087	0.0236	0.1688	0.0780

Each algorithm seed has its own library. Within that seed, the same library supports AEPS-OT and the controls that use historical configurations. Library construction is charged once when those conditions share it. Target probing is included in the target budget and task-run time; it receives no separate extra charge. The totals exclude out-of-run memory validation, process startup, and data preparation. Recorded wall-clock time also reflects cache behavior, so nominal slots remain the controlled search budget.