BOTAS: BAYESIAN OPTIMIZATION FOR ZERO-SHOT SOURCE TASK SELECTION IN MERGING OF LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Model merging combines independently fine-tuned checkpoints into a single model without joint training. However, existing evaluations assess the merged model only on the same in-domain source tasks, leaving zero-shot transfer to an unseen target task underresearched. To this end, we conduct an exhaustive study of source task subset selection for model merging, evaluating all $4{,}083$ subsets of 12 source tasks against 12 disjoint held-out target tasks, each at 40 scaling coefficients. Our results show that transfer is strongly aligned with the task family and that optimal mixtures typically involve fewer than 5 source tasks. We also show that a naive training-free solution based on training-free geometric selection, i.e., a task similarity measured by alignment between task vectors, does not identify the best mixture. To address this, we introduce BOTAS (Bayesian Optimization for TAsk Selection), which models the discrete space of source subsets with a Gaussian process surrogate. Without access to any target training data, BOTAS recovers the true optimal subset in 83% of runs while exploring only 1.5% of the search space. In addition, we show that in four out of twelve target tasks, the best merged subset outperforms full fine-tuning on the target itself. Finally, we release the full evaluation as a reusable benchmark for future source-selection methods.

1 INTRODUCTION

Large Language Models (LLMs) established the "pre-train, fine-tune" paradigm, in which reusing publicly available fine-tuned checkpoints has become a substantially cheaper alternative to training a new model from scratch, requiring neither fresh training data nor the compute budget of pre-training. Public hubs, such as Huggingface (Wolf et al., 2020) or AdapterHub (Pfeiffer et al., 2020), now host tens of thousands of task-specialized models derived from a small number of shared pre-trained backbones. *Task arithmetic* (Ilharco et al., 2023) exploits this abundance by making these checkpoints directly composable, taking the difference between a fine-tuned and a pre-trained checkpoint to form *a task vector*. Merging multiple models then equals the sum of multiple task vectors, yielding a single model that inherits multiple capabilities without joint training or access to the original data. Another line of work builds on top of task arithmetic and refines the computation of the sum (Yadav et al., 2023; Yu et al., 2024; Marczak et al., 2025; Gargiulo et al., 2025).

This abundance of pre-trained checkpoints also raises a natural question whether such pre-trained task vectors can be applied to a new, so far *unseen target task* – a situation which is more common in real-world settings with limited labeled data. Rather than fine-tuning from the pre-trained backbone alone, would merging a subset of the existing checkpoint hub transfer better, zero-shot, with no target training data? Answering this exactly means evaluating every subset of the n available source tasks, and the number of such subsets grows exponentially (reaching 4,083 at $n = 12$ in our setting).

Each candidate subset must be merged and scored before any information about it is available, so the objective is expensive to evaluate, carries no gradient, and is available only as a black box. *Bayesian optimization* addresses exactly this setting, since it fits a surrogate to the evaluations made so far and uses it to select the next candidate to evaluate. Prior work applies it to merging only over continuous mixing coefficients (Lee et al., 2025; Li et al., 2026), and not over the *discrete choice of included source tasks*.

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.

Table 1: Comparison of source task selection methods. Budget is the number of 4,083 candidate mixtures a method scores before choosing the best-performing mixture. Optimum found is the share of cases that reach the best available score across 12 target tasks and, for the stochastic methods, 10 seeds each (see Section 5 for more details). Diff to Optimum is the mean gap in that score, denoted Δ in the remainder of the paper.

	Target Training Set	Target Eval Set	Budget	Optimum Found	Diff to Optimum
Brute Force	✘	✔	4,083	100%	0.000
Target Similarity (Zhang et al., 2025)	✔	✔	4,083	8%	0.147
Cosine similarity (Vu et al., 2022)	✔	✔	4,083	0%	0.163
iso-SAR (Marczak et al., 2025)	✔	✔	4,083	0%	0.155
Random search	✘	✔	63	7%	0.030
BOTAS (ours)	✘	✔	**63**	**83%**	**0.003**

Because of this, we ask the following research question (RQ): **Can Bayesian optimization identify which subset of existing fine-tuned checkpoints to merge for an unseen target task, without any target training data and without evaluating the whole subset space?**

To the best of our knowledge, none of the existing works examined model merging on the unseen target task, nor researched the potential of Bayesian optimization for source task selection. The related work on task selection remains limited to returning a ranking or a single source task instead of a merged model, while transferring source tasks to only one target task (Phang et al., 2018; Achille et al., 2019; Vu et al., 2020; Schulte et al., 2024) and does not search the combinatorial space. Subsequently, the related works on model merging (Ilharco et al., 2023; Yadav et al., 2023; Yu et al., 2024; Marczak et al., 2025; Gargiulo et al., 2025) limit their evaluation to the in-domain source tasks, rather than the held-out target tasks.

To close this gap, we propose BOTAS (Bayesian Optimization for TAsk Selection), which models the combinatorial space of source subsets with a *Gaussian process surrogate*. We can then search for the source task combination that would most likely transfer well to a held-out target task while evaluating only a small fraction of that space, and without any training data for the target task.

Our main contributions are as follows (see also Table 1):

- **Source task selection without target training data.** We address subset selection for a held-out target using only a small held-out evaluation set to score candidate subsets, meaning no target training data, and no additional fine-tuning runs beyond the merges themselves. This is a harder setting than prior source-selection work, which typically assumes at least a few-shot labeled target set (e.g., Huang et al., 2024; Osial et al., 2025).
- **An exhaustive ground truth, reusable as a benchmark.** We provide an extensive sweep of subset selection for task-vector merging: 4,083 source-task subsets $\times$ 40 scaling coefficients $\times$ 12 held-out target tasks (resulting in more than 1.9M prediction runs). Beyond validating our own method, this sweep is, to our knowledge, the first exhaustive ground truth of its kind, and we release it as a reusable artifact[1] for benchmarking future source task selection methods without requiring them to rerun the full combinatorial search.
- **A Gaussian-process selection method that explores a fraction of the search space.** We propose a Gaussian-process-based selection method BOTAS that, on average, requires querying only 1.5% of the search space to find near-optimal subsets. While reducing computation costs considerably, BOTAS recovers the optimal subset/performance in 83% of cases, and in the remaining cases, the difference from the optimal performance is only negligibly lower (0.003 on average).
- **Merged subsets can outperform full fine-tuning.** We find that several selected source-task subsets, merged zero-shot with no access to target training data, outperform full fine-tuning on the target task, suggesting that for at least some target tasks, the checkpoints already contain a better starting point than training from scratch.

[1] https://anonymous.4open.science/r/bayesian-task-selection-C131

2 Related Work

Model merging. Model merging combines fine-tuned checkpoints in weight space without joint training. Model Soups (Wortsman et al., 2022) showed that simple averaging over multiple fine-tuned model weights improves single-task accuracy. Task Vectors (Ilharco et al., 2023) create vectors from pre-trained and fine-tuned weights, demonstrating arithmetic properties. TIES (Yadav et al., 2023), DARE (Yu et al., 2024), and Consensus Task Arithmetic/TIES (Wang et al., 2024) address negative parameter interference through trimming, sparsification, and majority-sign resolution. Other works utilize subspace or isometric alignment, such as Isometric Merging (Marczak et al., 2025), TSV-M (Gargiulo et al., 2025), KnOTS (Stoica et al., 2025), and tangent-space alignment (Ortiz-Jimenez et al., 2023), which relate merge quality to geometric properties. Finally, Yoshida et al. (2026) shows that merging methods are typically benchmarked under conditions that mask their sensitivity to task-vector-norm disparities and low source-model confidence.

Across these methods, the source task pool remains fixed, and each model is scored on the tasks it was built on. Several works relax parts of this setting. Ding et al. (2025) select a subset of domain experts before merging, but only for named entity recognition and by embedding similarity rather than by search. Bolton et al. (2026) predicts the merge operator and order for a fixed set of models, without addressing which sources to include. Lee et al. (2025) and Li et al. (2026) apply Bayesian optimization to continuous quantities, namely scaling coefficients and module-wise regularization strengths, rather than to discrete subset membership.

Source task selection. Orthogonally to model merging, the source task selection approaches aim to informatively select the best subset of source tasks. Out of them, intermediate-task and transferability studies ask which source improves a target without merging. STILTs (Phang et al., 2018) and related work (Pruksachatkun et al., 2020; Poth et al., 2021) fine-tune sequentially through a single intermediate task. The closest related work in scale (Vu et al., 2020) evaluates over $3{,}000$ source-target pairs across 33 datasets, using task embeddings to identify the best pairs without combining source tasks, and Achille et al. (2019) propose Task2Vec embeddings for the same goal. Schulte et al. (2024) scale to $12{,}000$ pairs with Embedding Space Maps, motivated explicitly by the infeasibility of evaluating every pair as the search space grows. DTVG (Zhang et al., 2025) ranks sources by dot-product similarity with dynamic grouping, while Lin et al. (2024) shows substantial variation across both source tasks and training seeds for 130 combinations. In all these cases, the output is a ranking or a single selected source followed by sequential fine-tuning, not a merged subset. Closest to our approach, Guo et al. (2026) searches the combinatorial space of model subsets for stereo matching using a genetic algorithm, but evaluates on the source models' own benchmark pool rather than on a disjoint held-out target.

To our knowledge, **no existing model merging method treats source selection for a held-out target as a combinatorial search problem validated against exhaustive ground truth**. Our contribution, therefore, lies at the intersection of model merging and source task selection, focusing on selecting a subset for merging, evaluating zero-shot performance on a held-out target, and validating against an exhaustive combinatorial sweep.

3 Problem Formulation

Let $\theta_{pre} \in \mathbb{R}^d$ represent vectorized parameters of a pre-trained model and $\theta^t_{ft} \in \mathbb{R}^d$ represent vectorized parameters of a fine-tuned model on a specific task t. Both models share the same architecture and operate in the same parameter space $\mathbb{R}^d$.

Task Vectors. Following the framework of Task Arithmetic (TA) (Ilharco et al., 2023), we represent Task Vector as a difference between pre-trained weights and fine-tuned checkpoint. Specifically for a single task t the task vector is defined as $\tau_t = \theta^t_{ft} - \theta_{pre}$, for $t \in \{1, \ldots, T\}$.

Merging a Subset. Given a subset of source tasks $S \subseteq \{1, \ldots, T\}$ and a shared scaling coefficient $\lambda \in \Lambda$, we obtain a merged model by summing the corresponding task vectors:

$$\theta(S, \lambda) = \theta_{pre} + \lambda \sum_{t \in S} \tau_t. \tag{1}$$

We utilize a single shared coefficient rather than per-task or per-layer weighting, so that any effect we measure is attributable to *which* sources are included in S. Sign-conflict resolution, sparsification, and spectral reweighting (Yadav et al., 2023; Yu et al., 2024; Marczak et al., 2025; Gargiulo et al., 2025) refine Equation 1 and are orthogonal to the selection problem we study.

Zero-Shot Held-Out Target. We are given a target task $g \notin \{1, \ldots, T\}$, which is disjoint from every source task. Throughout our paper, *zero-shot* refers to the absence of any training-time adaptation to g. This means that $\theta(S, \lambda)$ is never fine-tuned, and that no gradient update is ever computed with respect to target labels. A labeled evaluation set for g is used only to *score* candidate merges. We use this same evaluation set both to compute the exhaustive ground truth in Section 5.2 and to score the found mixtures in Section 4, so the two are directly comparable, and the query efficiency we report is not an artifact of an easier split.

Selection Objective. Let $\mathrm{Acc}_g(\theta)$ denote the evaluation metric of task g for a model with parameters θ (Section 5.1 details the per-task metrics used). The subset selection problem is:

$$S^\star, \lambda^\star = \underset{S \subseteq \{1,\ldots,T\},\ \lambda \in \Lambda}{\arg\max} \ \mathrm{Acc}_g\big(\theta(S, \lambda)\big). \tag{2}$$

The outer maximization ranges over $2^T - T - 1$ subsets of size ≥ 2, which is $4{,}083$ at $T = 12$, and each requires one merge-and-evaluate pass per candidate λ. Evaluating Equation 2 exhaustively, therefore, grows exponentially with the number of source tasks. We treat it (Section 5.1) as ground truth for validating a search method, rather than as a procedure that would be feasible in practice. Against that ground truth, we score any returned mixture by its *difference to the optimum*,

$$\Delta(S, \lambda) = \mathrm{Acc}_g\big(\theta(S^\star, \lambda^\star)\big) - \mathrm{Acc}_g\big(\theta(S, \lambda)\big), \tag{3}$$

so $\Delta = 0$ means that an accuracy-optimal mixture was selected. We report Δ when comparing selection methods, since it is measured against the best mixture available in the search space, not against fine-tuning on the target.

Normalized Accuracy Improvement (NAI). To compare across target tasks with different metrics, scales, and evaluation-set sizes, we report NAI (Marczak et al., 2025):

$$\mathrm{NAI}(S, \lambda) = \frac{\mathrm{Acc}_g(\theta(S, \lambda)) - \mathrm{Acc}_g(\theta_{pre})}{\mathrm{Acc}_g(\theta^g_{ft}) - \mathrm{Acc}_g(\theta_{pre})}, \tag{4}$$

where θ^g_{ft} is the model fully fine-tuned on g using labeled target training data, which is unavailable in our own zero-shot setting. A value of $\mathrm{NAI} = 1$ means that the merge matches direct fine-tuning on the target. Values above 1 are possible, and as we show in Section 5.2, they actually also occurred in our experiments on four out of twelve target tasks.

4 BOTAS: Bayesian Optimization for TAsk Selection

Exhaustive search over a large subset space is the only way that guarantees an optimal $S^\star$ (Eq. 2), and Section 3 showed that this cost grows exponentially with the number of source tasks. To address this, we proposed a novel method called BOTAS, which treats subset selection as black-box optimization over the combinatorial space itself. It uses a Gaussian Process (GP) surrogate to decide which of the $4{,}083$ candidate mixtures to evaluate next. Algorithm 1 describes the full optimization process of BOTAS.

Reducing to a search over subsets. Evaluating a candidate mixture S against the objective in Eq. 2 still requires a choice of λ. Rather than treat (S, λ) as a joint search space, we optimize out λ using the same per-mixture coefficient sweep that produced the exhaustive ground truth (see Section 5.2):

$$f(S) = \max_{\lambda \in \Lambda} \ \mathrm{Acc}_g\big(\theta(S, \lambda)\big). \tag{5}$$

BOTAS searches over $S \subseteq \{1, \ldots, T\}$ only, and computes $f(S)$ by merging and evaluating a new model. In addition, the Gaussian process requires a fixed-dimensional input for each candidate S.

Algorithm 1 BOTAS

Input: candidate subsets $\{1, \ldots, 2^T - T - 1\}$, feature map $z(\cdot)$, initial budget n_0, budget N
Draw n_0 subsets uniformly at random; evaluate $f(S)$ for each via Eq. 5; add to $\mathcal{D}_0$
for $t = n_0, \ldots, N-1$ **do**
  Fit GP surrogate on $\{(z(S), f(S)) : S \in \mathcal{D}_t\}$
  $S_{t+1} \leftarrow \arg\max_{S \notin \mathcal{D}_t}$ ExpectedImprovement(S)
  Evaluate $f(S_{t+1})$; $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{S_{t+1}\}$
end for
Return: $\arg\max_{S \in \mathcal{D}_N} f(S)$

Feature representation. In the GP surrogate, we represent S by its binary subset indicator $z(S) \in \{0,1\}^T$, with $z(S)_i = 1$ if source $i \in S$. This is the simplest encoding that distinguishes every subset in the space, and it needs no access to the source task vectors. We compare it against alignment-based encodings that utilize source tasks, namely Subspace Alingment Ratio (SAR) (Marczak et al., 2025), $\text{sar}(S)$, which projects the target task vector onto the subspace spanned by the merged task vectors and scores the merged vectors by how much of the target vector energy survives the projection[2] together with an isotropic variant (iso-SAR) $\text{sar}_{\text{iso}}(S)$, each optionally concatenated onto $z(S)$ or normalized by $|S|$ to control for the tendency of an alignment score to grow with mixture size. Every representation is computable before any merged model is evaluated.

Choice of covariance function. We use the ARD RBF kernel (Neal, 2012) with the dimension-scaled log-normal lengthscale prior of Hvarfner et al. (2024). Due to binary subset indicators, we use a weighted Hamming kernel (Hutter et al., 2011). The Matérn family is usually preferred in continuous Bayesian optimization because the infinite differentiability of RBF sample paths is an unrealistic prior. That argument does not transfer to our setting, where the smallest move in the search space is the addition or removal of a single source task.

Surrogate and acquisition. We place a Gaussian process prior over f,

$$f(S) \sim \mathcal{GP}\big(\mu(z(S)),\ k(z(S), z(S'))\big), \tag{6}$$

fit as a `SingleTaskGP`, and select the next candidate to query by maximizing expected improvement (EI) over all unevaluated subsets:

$$S_{t+1} = \arg\max_{S \notin \mathcal{D}_t} \mathbb{E}\big[\max(f(S) - f_t^+, 0)\big], \tag{7}$$

where $\mathcal{D}_t$ is the set of subsets evaluated so far and f_t^+ is the best value observed within it.

For each (target, feature representation, seed) triple, we record whether BOTAS recovers the exact optimum $S^\star$ within budget N, and we record the number of evaluations needed to reach within 1% of $f(S^\star)$. Since $f(S)$ is looked up against the same evaluation set that we use to establish $S^\star$ itself (Section 3), recovery is measured against that set's own optimum.

5 RESULTS

5.1 EXPERIMENTAL SETUP

Our study covers 24 datasets across five task families, split into 12 source tasks and 12 disjoint held-out target tasks, most of which are standard GLUE (Wang et al., 2018), SuperGLUE (Wang et al., 2019), and commonsense reasoning benchmarks. Moreover, every task family should contain an equal number of source and target tasks, while larger datasets are primarily assigned as source tasks. We also follow the transfer-learning convention of Vu et al. (2022) and Asai et al. (2022), so that each target has related and unrelated source tasks and no conclusion rests on a single task family.

Table 2 lists the split within each family, together with zero-shot accuracy (pre-trained backbone) on the corresponding task, and full fine-tuning (FFT) accuracy (fine-tuned on the task's own training

[2]As implemented, SAR is the fraction of the total (Frobenius) energy of the task vector retained by the projection

Table 2: Full fine-tuning (FFT) and zero-shot performance of Llama-3.2-1B-Instruct on all 24 datasets, grouped by task family. FFT is fine-tuned on the task's own training set and reported as the mean ± standard deviation over seeds 42, 123, and 456. Zero-shot is evaluated with no training data and is therefore identical across seeds, so no deviation is reported. Values are an exact match, except STS-B, which reports Pearson correlation. Source tasks are shaded blue and held-out target tasks orange.

	Task	FFT	Zero-shot
Entailment / NLI	MNLI (Williams et al., 2018)	0.896 ± 0.002	0.001
	QNLI (Wang et al., 2018)	0.934 ± 0.005	0.005
	SNLI (Bowman et al., 2015)	0.905 ± 0.002	0.000
	ANLI-R1 (Nie et al., 2020)	0.587 ± 0.019	0.000
	RTE (Wang et al., 2018)	0.780 ± 0.011	0.256
	CB (De Marneffe et al., 2019)	0.607 ± 0.107	0.000
	SciTail (Khot et al., 2018)	0.958 ± 0.003	0.012
Paraphrase	QQP	0.907 ± 0.004	0.058
	PAWS (Zhang et al., 2019)	0.938 ± 0.004	0.094
	MRPC (Dolan & Brockett, 2005)	0.842 ± 0.009	0.005
	STS-B (Cer et al., 2017)	0.891 ± 0.004	0.265
Sentiment & acceptability	SST-2 (Socher et al., 2013)	0.951 ± 0.003	0.389
	IMDb (Maas et al., 2011)	0.962 ± 0.005	0.343
	CR (Ding et al., 2008)	0.935 ± 0.004	0.184
	RT (Pang & Lee, 2005)	0.915 ± 0.008	0.231
	CoLA (Warstadt et al., 2019)	0.829 ± 0.003	0.664
Question answering	ReCoRD (Zhang et al., 2018)	0.791 ± 0.004	0.005
	SQuAD v2 (Rajpurkar et al., 2018)	0.828 ± 0.002	0.419
	BoolQ (Clark et al., 2019)	0.829 ± 0.003	0.387
	MultiRC (Khashabi et al., 2018)	0.855 ± 0.002	0.440
Commonsense reasoning	HellaSwag (Zellers et al., 2019)	0.852 ± 0.000	0.000
	WinoGrande (Sakaguchi et al., 2021)	0.683 ± 0.009	0.000
	COPA (Roemmele et al., 2011)	0.730 ± 0.026	0.300
	PIQA (Bisk et al., 2020)	0.794 ± 0.004	0.368

set). To create the source and target task vectors, we fine-tune a representative and popular autoregressive model, Llama-3.2-1B-Instruct (Grattafiori et al., 2024), on each corresponding task using three different seeds (42, 123, 456). For fine-tuning of the source and target tasks, we utilize the PEFT-Factory framework (Belanec et al., 2026). For the Bayesian Optimization and surrogate fitting, we utilize the BoTorch framework (Balandat et al., 2020). In our experimental setup, BOTAS takes advantage of a precomputed exhaustive ground truth and uses it to look up the validation accuracy for each source-task combination. This makes the search *retrospective*, since no additional fine-tuning or merging occurs beyond what was already computed during the exhaustive search.

We evaluate source task subsets of size $k = 2, \ldots, 12$ from the 12 source tasks against each of the 12 held-out target tasks, at 40 values of λ, which gives $4{,}083$ subsets $\times$ 40 coefficients $\times$ 12 target tasks. We run BOTAS with an initial number of subsets of $n_0 = 3$ and a budget of $N = 63$, which is 1.5% of the $4{,}083$-subset space, independently for each of the 12 target tasks and across 10 random seeds that control the initial design. We set N to the smallest budget that gives results close to optimum (or directly the optimum in most of the cases), since BOTAS recovers optimum in 28% at 0.5% of the search space and in 83% at 1.5%. We set $n_0 = 3$ because it is the smallest initial design the surrogate can be fit from (for additional information, see Appendix F.1).

Training-free selection baselines. Alongside evaluating merged models, we compute a set of baselines that score every candidate mixture in closed form from the single-task checkpoints alone, with no merging and no evaluation pass. We evaluate SAR and SAR-iso (Marczak et al., 2025) as a closed-form baseline (defined in Section 4). In addition, we also evaluate Target Similarity,

Table 3: Best-performing source mixture per target over the full $4{,}083$-subset space, ordered by the accuracy reached. k is the mixture size, λ the scaling coefficient, and NAI is rescaled between the zero-shot and fine-tuned columns of Table 2.

Target	Best mixture	k	λ	NAI
CR	SST-2+ReCoRD+ANLI-R1	3	0.40	0.998
RT	SST-2+ReCoRD+IMDb	3	0.55	1.020
CB	MNLI+SST-2+SNLI	3	0.80	1.471
SciTail	MNLI+QQP+ANLI-R1+HellaSwag	4	0.30	0.901
RTE	MNLI+QNLI+QQP+SST-2+ANLI-R1+PAWS+SQuAD v2+WinoGrande	8	0.25	1.117
COPA	HellaSwag+WinoGrande	2	0.35	1.093
MultiRC	MNLI+QNLI+ANLI-R1+PAWS	4	0.20	0.764
PIQA	IMDb+HellaSwag+WinoGrande	3	0.35	0.829
STS-B	QNLI+PAWS+WinoGrande	3	0.45	0.703
BoolQ	QNLI+ReCoRD+PAWS+HellaSwag+WinoGrande	5	0.35	0.706
MRPC	ReCoRD+HellaSwag	2	0.80	0.812
CoLA	ReCoRD+IMDb+HellaSwag	3	0.30	0.058

which is defined as an average of normalized dot products between each individual source task in a specific subset and the target task, and Knowledge Consistency, defined as an average of a pairwise normalized dot product between all source tasks in a subset (both proposed by Zhang et al. (2025)). Lastly, we also evaluate Cosine Similarity and Dot Product, which are computed as each function over the combined source tasks and target task.

5.2 Exhaustive Ground Truth

Winning mixtures draw on the target's own task family. Table 3 reports, for each target, the best-performing source mixture over the full search space. Source tasks of the same task family as the target task are often included in the best-performing mixture. Commonsense target tasks select commonsense sources, CR and RT both include SST-2, and the NLI target tasks include NLI sources. The winning mixture contains at least one source task from the target task's task family for 10 of 12 target tasks, compared with the expected value of 6.8 for random mixtures of the same sizes (we calculate the expected value and statistical significance in Appendix B). The two exceptions are MRPC and MultiRC. Since family membership is never supplied to the search, this alignment reflects task structure rather than a property of the merging procedure.

Optimal mixtures are small, but not reliably so. Ten of the twelve target tasks are best served by a mixture of length $k \leq 4$, with a median of 3, and two of them by a single source task pair. RTE, however, selects an 8-source mixture, and merging all 12 sources is never optimal. No single source task dominates the winning mixtures either, as the most frequent member is HellaSwag with 6 of the 12. Every target nonetheless selects a *distinct* subset, which is the ground truth that a selection method must predict.

Four target tasks benefit more from model merging than from full fine-tuning. Four target tasks, namely CB, RTE, COPA, and RT, achieve NAI > 1, so the best merged model outperforms the model fine-tuned directly on the target. These four have the smallest training sets, where direct fine-tuning is prone to overfitting, whereas the merged sources contribute signal learned from much larger training sets elsewhere. At the other extreme, CoLA has a best achievable NAI of 0.057, because its zero-shot accuracy is already 0.664 against a 0.832 fine-tuned ceiling (Table 2), so the best mixture recovers only 6% of the fine-tuning gain. CoLA is also the only target task that addresses linguistic acceptability, a task type that none of the 12 source tasks cover.

Ties and scaling coefficients. On the smallest target tasks, the optimum is not a unique object, since 67 of the $4{,}083$ mixtures tie exactly at CB's maximum accuracy. We break ties toward the smaller mixture throughout. Appendix C gives the tie counts, and Appendix C.1 reports the distribution of the scaling coefficient, which concentrates sharply toward the low end of the swept grid.

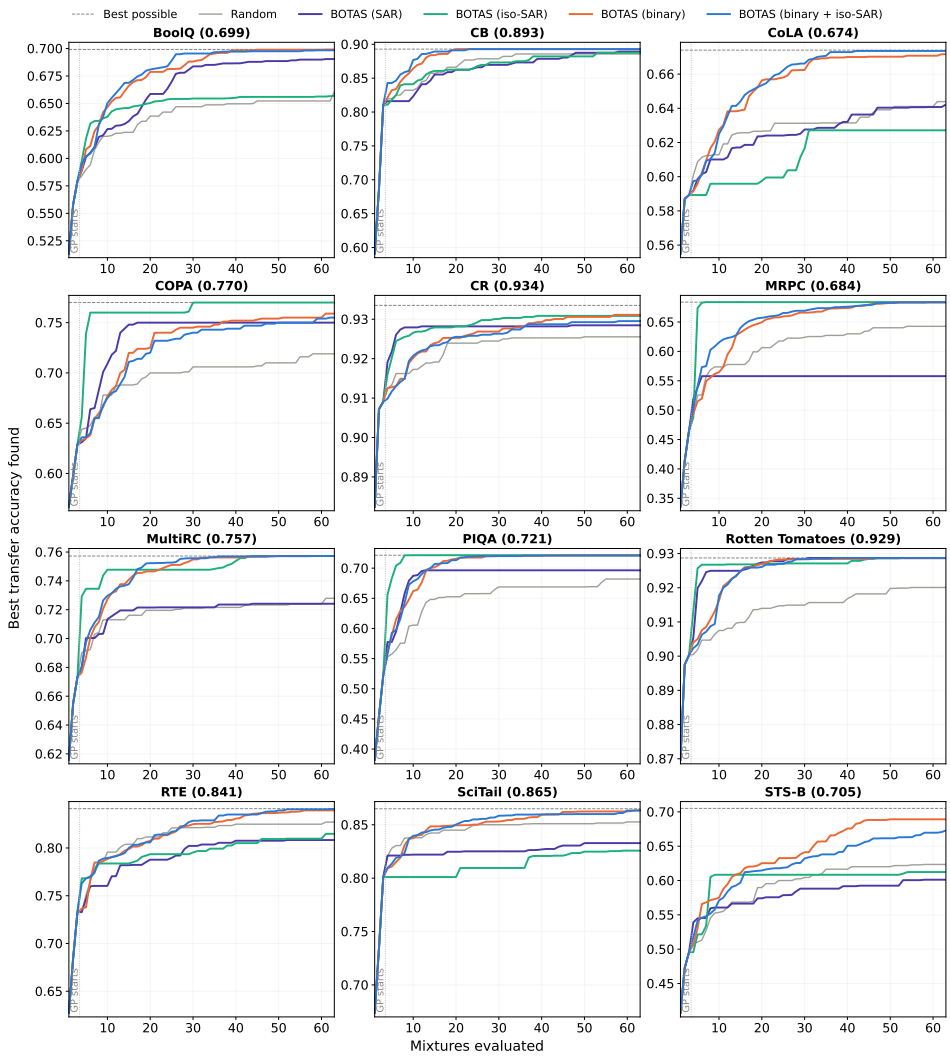


Figure 1: Best transfer accuracy found as a function of evaluations spent, per target task, averaged over 10 seeds. The value in each panel heading is the best accuracy available. The vertical dotted line marks where the GP surrogate begins guiding the search, after the $n_0 = 3$ initial random draws.

5.3 Recovering the Best Subset Using BOTAS

BOTAS recovers the optimum from 1.5% of the space. BOTAS recovers the optimum at least as often as random search at every budget between n_0 and N, and passes random search's final optimum recovery rate after 10 evaluations. Aggregated across all 12 target tasks and 10 seeds (Figure 1), the binary feature representation finds the exact optimum in 83% of runs, within an average of 29.1 of the 4,083 candidates, which is roughly 0.7% of the space, compared with 7% of runs for random search. Adding iso-SAR to the indicator changes nothing measurable, as the two reach 82% and 83%. Arity-normalizing the alignment features makes them *worse*, not better, reducing SAR from 15% to 12% and iso-SAR from 48% to 22%. Appendix D shows how each representation converges on each target task, and Appendix F reports per-target recovery rates, convergence speed, and final accuracy.

Subspace alignment does not identify the best mixture. If SAR predicted transfer on its own, selection could skip the combinatorial search entirely and rank mixtures by a closed-form geometric score. However, this is not the case. The per-target rank correlation between SAR and transfer accuracy is low in the median and inconsistent in sign, and it is *negative* on 3 of 12 target tasks. The

Table 4: Selection quality of BOTAS against the training-free geometric baselines. Δ is the difference to the optimum, ρ is the median per-target Spearman correlation with accuracy. "Found best" counts target tasks where the selected source task combination is optimal.

Method	mean Δ	med. Δ	med. ρ	found best
Knowledge Consistency (Zhang et al., 2025)	0.414	0.378	-0.14	0%
Dot Product	0.257	0.234	0.335	0%
SAR	0.181	0.142	0.174	0%
Cosine Similarity (Vu et al., 2022)	0.163	0.108	0.398	0%
iso-SAR	0.155	0.110	0.421	0%
Target Similarity (Zhang et al., 2025)	0.147	0.069	0.416	8%
BOTAS (SAR)	0.035	0.025	—	15%
Random Search	0.030	0.023	—	7%
BOTAS (iso-SAR)	0.021	0.003	—	48%
BOTAS (SAR, norm)	0.019	0.008	—	12%
BOTAS (iso-SAR, norm)	0.012	0.005	—	22%
BOTAS (binary + iso-SAR)	0.005	0.000	—	82%
BOTAS (binary)	**0.003**	**0.000**	—	**83%**

isotropic variant is somewhat stronger in the median, but it still has the wrong sign on MRPC, as reported in Appendix E. Correlation is, in any case, not a sufficient criterion for selection. Even on RT and CR, where SAR ranks mixtures well, the mixtures that SAR ranks in its top 5 overlap with the true top 5 by accuracy in close to 0 of 5 cases. Appendix D shows that SAR grows with mixture size, and that correcting for this does not rescue it.

Target similarity is the strongest training-free selection baseline, and still loses to Bayesian search. Table 4 summarizes the selection quality of the six training-free selection baselines. Target similarity is the strongest baseline, as it reduces the median Δ of iso-SAR from 0.110 to 0.069. However, it remains a poor selector in absolute terms, since it finds the optimum only on 8% of tasks, whereas the binary feature representations in GP find it in 83% of runs. Random search sharpens the comparison further. It spends only 63 merge-and-evaluate passes, and still reaches a mean Δ of 0.030 against 0.147 for Target Similarity (Appendix F). In addition, some of the training-free selection baselines (Target Similarity, Cosine Similarity, Dot Product, and both SAR variants) require the target tasks to compute the score, which requires full fine-tuning on the target training data, whereas BOTAS does not.

6 Conclusion

We asked whether Bayesian optimization can identify which subset of fine-tuned checkpoints to merge for an unseen target task, without access to target training data and without evaluating the entire subset space. Exhaustively evaluating $4{,}083$ subsets of 12 sources against 12 held-out target tasks showed that transfer is strongly aligned with task family, and that optimal mixtures are usually small, but not reliably so, as one target task selects 8 of the 12 sources. On four target tasks, the best merged model outperforms fine-tuning directly on the target itself.

BOTAS treats subset selection as black-box optimization over the discrete space of source subsets, with a Gaussian process surrogate over binary subset indicators. Using only a small target evaluation set and no labeled target training data, it recovers the exact optimal subset in 83% of runs while evaluating 63 of the $4{,}083$ candidates.Where it does not find the optimum, it falls short by a mean Δ of 0.003. Lastly, geometric task-vector features do not improve performance over binary feature representations, and in some variants, they make it measurably worse.

BOTAS is also highly applicable and scalable, since the budget in Bayesian Optimization can be defined by real-world computational constraints, and in our experiments, we show that a budget of 1.5% of the search space is sufficient to obtain the optimum (or close to the optimum).

We release the full $4{,}083 \times 40 \times 12$ evaluation grid as a reusable benchmark, so that future source-selection methods can be scored against exhaustive ground truth rather than a learned proxy.

AI USE STATEMENT

In this work, we used generative AI tools for spell-checking, paragraph reformulation, and summarizing results, which were then adapted into the final texts of this paper. We have not used generative AI tools to generate related work or references. Additionally, we used generative AI tools to assist with code writing and experiment preparation. We have reviewed all AI-assisted work. All LLM-generated code was verified and tested for correctness and security, and all LLM-generated texts and summaries were manually checked and rewritten by 2 authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The experiments in this paper were conducted with publicly available datasets MNLI, QNLI, SNLI, ANLI-R1, RTE, CB, SciTail, QQP, PAWS, MRPC, STS-B, SST-2, IMDb, Customer Reviews, Rotten Tomatoes, CoLA, ReCoRD, SQuADv2, BoolQ, MultiRC, HellaSwag, WinoGrande, COPA, and PIQA, citing the original authors. As we were unable to determine the licenses for all used datasets, we have opted to use them in the limited form possible, adhering to the terms of use of the GLUE and SuperGLUE benchmarks and, for the remaining datasets, to the terms set by their original authors. As the datasets are commonly used in related work and have been published in scientific works that underwent an established review process, we do not check for offensive content, as it has already been removed by the authors of these publicly available datasets. In addition, we do not use any personally identifiable information or offensive content, and we do not engage in crowdsourcing for data annotation. To our knowledge, we are not aware of any potential ethical harms or negative societal impacts of our work, apart from those related to the field of Machine Learning (i.e., the use of energy-consuming computational resources that produce heat and indirect CO2 emissions). We follow the license terms for the Llama-3.2-1B-Instruct model we use – all models and datasets allow their use as part of the research. As we score classification tasks by exact match against a fixed label set, we, in most cases, minimize the risk of generating offensive or biased content, although extractive question-answering tasks produce free-form spans.

The experiments in this paper require a significant amount of GPU computing resources. We fine-tune 1 model on each of 24 datasets, and then evaluate each of the $4{,}083$ source subsets against each of the 12 held-out targets at 40 scaling coefficients, which amounts to $1{,}959{,}840$ merge-and-evaluate passes. The selection experiments themselves add no GPU cost, since they query this precomputed grid by lookup. Overall, the experiments, including evaluations and preliminary experiments (which are not within the scope of our work), were conducted on a private infrastructure with a carbon efficiency of 0.432 kgCO_2eq/kWh. Approximately 8000 hours of computation were performed on NVIDIA H200 hardware (TDP 700W). Total emissions are estimated to be 2,419 kgCO_2eq, of which 0 percent were directly offset. Whenever possible, we tried to reduce the computational costs.

REPRODUCIBILITY STATEMENT

Section 5.1 lists the 24 datasets, the source and target split, the model, and the seed, and states the size of the evaluation grid, while Section 4 and Algorithm 1 gives the surrogate, the acquisition function, and the feature representations under test, and the evaluation budget, together with the evidence that our conclusions do not depend on it. The appendices hold what did not fit in the main text due to page limit, namely, the per-target recovery, convergence speed, and difference to the optimum in Appendix F, the distribution of the scaling coefficient in Appendix C.1, the tie structure that makes the optimum non-unique on the smallest targets in Appendix C, the null model behind the family-alignment test in Appendix B, the training-free selection baselines, and their per-target correlations in Appendix E. We release an anonymized repository at `https://anonymous.4open.science/r/bayesian-task-selection-C131` containing the code for every experiment and analysis reported here, together with the full $4{,}083 \times 40 \times 12$ evaluation grid as a single table of per-mixture, per-target results. Because that grid is released, every number, figure, and appendix table in this paper can be regenerated from the committed artifacts without repeating the sweep, and our selection experiments are themselves lookups against it rather than new merges. The dataset path in the released code is redacted for review, so the data-loading code does not run as-is. The repository README lists each placeholder and what it represents.

REFERENCES

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6430–6439, 2019.

Akari Asai, Mohammadreza Salehi, Matthew Peters, and Hannaneh Hajishirzi. ATTEMPT: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6655–6672, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.446. URL https://aclanthology.org/2022.emnlp-main.446/.

Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538, 2020.

Robert Belanec, Ivan Srba, and Maria Bielikova. Peft-factory: Unified parameter-efficient finetuning of autoregressive large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 188–202, 2026.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Oliver Bolton, Aakanksha, Arash Ahmadian, Sara Hooker, Marzieh Fadaee, and Beyza Ermis. Simmerge: Learning to select merge operators from similarity signals. In *Third Workshop on Test-Time Updates (Main Track)*, 2026. URL https://openreview.net/forum?id=7cQDm31mKF.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, and Jian Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075/.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens (eds.), *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14, Vancouver, Canada, August 2017. ACL. doi: 10.18653/v1/S17-2001.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pp. 107–124, 2019.

Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pp. 231–240, 2008.

Zhuojun Ding, Wei Wei, and Chenghao Fan. Selecting and merging: Towards adaptable and scalable named entity recognition with large language models. In Wanxiang Che, Joyce Nabende,

Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9869–9886, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.487. URL https://aclanthology.org/2025.acl-long.487/.

William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the International Workshop on Paraphrasing*, 2005.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodola. Task singular vectors: Reducing task interference in model merging. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18695–18705. IEEE, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Xianda Guo, Pinhan Fu, Ruilin Wang, Wenke Huang, Mang Ye, and Qin Zou. Stereofactory: A unified merging framework for robust stereo matching. *arXiv preprint arXiv:2606.17475*, 2026.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic loRA composition. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=TrloAXEJ2B.

Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the 5th International Conference on Learning and Intelligent Optimization*, LION'05, pp. 507–523, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 9783642255656. doi: 10.1007/978-3-642-25566-3_40. URL https://doi.org/10.1007/978-3-642-25566-3_40.

Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla Bayesian optimization performs great in high dimensions. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20793–20817. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/hvarfner24a.html.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 252–262, 2018.

Tushar Khot, Ashish Sabharwal, and Peter Clark. SciTail: A textual entailment dataset from science question answering. In *AAAI*, 2018.

Sanwoo Lee, Jiahao Liu, Qifan Wang, Jingang Wang, Xunliang Cai, and Yunfang Wu. Dynamic fisher-weighted model merging via Bayesian optimization. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4923–4935, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.254. URL https://aclanthology.org/2025.naacl-long.254/.

Kaiyang Li, Shaobo Han, Qing Su, and Shihao Ji. Bayesian model merging. *arXiv preprint arXiv:2605.12843*, 2026.

Pin-Jie Lin, Miaoran Zhang, Marius Mosbach, and Dietrich Klakow. Exploring the effectiveness and consistency of task selection in intermediate-task transfer learning. In Xiyan Fu and Eve Fleisig (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 170–185, Bangkok, Thailand, August 2024. Association for Computational Linguistics. ISBN 979-8-89176-097-4. doi: 10.18653/v1/2024.acl-srw.24. URL https://aclanthology.org/2024.acl-srw.24/.

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, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=RBZpAa27ls.

Radford M Neal. *Bayesian learning for neural networks*. Springer Science & Business Media, 2012.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.441. URL https://aclanthology.org/2020.acl-main.441/.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36:66727–66754, 2023.

Marcin Osial, Bartosz WĂłjcik, Sebastian Cygert, et al. Efficient multi-source knowledge transfer by model merging. *arXiv preprint arXiv:2508.19353*, 2025.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, 2005.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. AdapterHub: A framework for adapting transformers. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 46–54, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.7. URL https://aclanthology.org/2020.emnlp-demos.7/.

Jason Phang, Thibault Févry, and Samuel R Bowman. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*, 2018.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. What to pre-train on? Efficient intermediate task selection. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10585–10605, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.827. URL https://aclanthology.org/2021.emnlp-main.827/.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. Intermediate-task transfer learning with pretrained language models: When and why does it work? In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5231–5247, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.467. URL https://aclanthology.org/2020.acl-main.467/.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pp. 90–95, 2011.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, August 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

David Schulte, Felix Hamborg, and Alan Akbik. Less is more: Parameter-efficient selection of intermediate tasks for transfer learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9431–9442, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.529. URL https://aclanthology.org/2024.emnlp-main.529/.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on EMNLP*, pp. 1631–1642, 2013.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. In *International Conference on Learning Representations*, volume 2025, pp. 4501–4519, 2025.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. Exploring and predicting transferability across NLP tasks. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7882–7926, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.635. URL https://aclanthology.org/2020.emnlp-main.635/.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. SPoT: Better frozen model adaptation through soft prompt transfer. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5039–5059, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.346. URL https://aclanthology.org/2022.acl-long.346/.

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. *arXiv preprint arXiv:1804.07461*, 2018.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jiménez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the ACL*, 7:625–641, 2019. doi: 10.1162/tacl_a_00290.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the ACL: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. ACL. doi: 10.18653/v1/N18-1101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. Pmlr, 2022.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

Kotaro Yoshida, Yuji Naraki, Takafumi Horie, Ryotaro Shimizu, Ioannis Mitliagkas, and Hiroki Naganuma. Distac: Conditioning task vectors via distillation for robust model merging. In *International Conference on Learning Representations*, volume 2026, pp. 118509–118534, 2026.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Peiyi Zhang, Richong Zhang, Zhijie Nie, and Ziqiao Wang. Dynamic task vector grouping for efficient multi-task prompt tuning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 26805–26821, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1374. URL https://aclanthology.org/2025.findings-acl.1374/.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*, 2018.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*, 2019.

A LIMITATIONS

Due to the computational requirements of extensive ground-truth search, we evaluate BOTAS on a single popular representative model, Llama-3.2-1B-Instruct (Grattafiori et al., 2024). Similarly, it is also computationally expensive to empirically evaluate scalability to a larger number of source tasks. However, we evaluate multiple seeds of pre-trained source tasks and Bayesian Optimization, which suggests that our findings will hold across different language model families or sizes, as well as larger source task pools.

We use the same evaluation set both to search over S and to report the final accuracy of the selected subset. The scores we report are therefore subject to selection-induced optimism, also known as the "winner's curse", since searching through many candidates and reporting the best-scoring candidate's own score yields a biased upward estimate of that candidate's true performance. This effect shrinks with evaluation-set size, and it is most relevant for our smallest target tasks, namely CB with $n = 56$ and COPA with $n = 100$, relative to our largest target tasks, namely MultiRC with $n = 4{,}848$ and BoolQ with $n = 3{,}270$.

Table 5: Distribution of the winning scaling coefficient λ across $n = 48{,}996$ (mixture, target) pairs.

λ interval	share	cumulative
$[0.05, 0.10)$	35.6%	35.6%
$[0.10, 0.20)$	43.9%	79.5%
$[0.20, 0.30)$	10.1%	89.6%
$[0.30, 0.50)$	4.8%	94.4%
$[0.50, 0.75)$	3.4%	97.8%
$[0.75, 1.00)$	1.6%	99.4%
$[1.00, 1.50)$	0.6%	100.0%
$[1.50, 2.00]$	0.0%	100.0%

Lastly, the computational cost of calculating Expected Improvement can increase with the number of source tasks. However, the increased computational cost of BOTAS is offset by the reduced need for training data, which can be more expensive to acquire. In addition, this problem can be potentially solved by sampling, local neighborhoods, structured kernels, or heuristic candidate sets.

B Family Alignment Test

Section 5.2 reports that winning mixtures tend to draw on the target's own task family. Because we obtain that observation by taking an argmax over $4{,}083$ candidates per target, selective reading of the winners could produce it by chance, so we test it rather than argue from examples.

We assign each of the 24 datasets to one of the five families of Section 5.1 and score each target on whether its accuracy-optimal mixture contains at least one source from the target's own family, breaking ties toward the smaller mixture as elsewhere. The null draws a uniformly random mixture of the same size from the 12 sources, which holds mixture size fixed so that the statistic cannot be driven by the size distribution alone. Over $200{,}000$ draws, the observed count of 10 of 12 aligned target tasks compares with an expected 6.8, yielding $p = 0.038$.

The effect is therefore real but modest across 12 target tasks, and uneven among them. COPA's winner is entirely the same family, and RT's is 0.67 same-family against a pool share of 0.17, whereas BoolQ's is 0.20 against the same 0.17 pool share, which is close to the pool rate. MRPC and MultiRC contain no same-family source at all. We therefore read family alignment as a tendency in the landscape rather than a rule that a selection method could rely on.

C Exhaustive Ground Truth Details

At CB's maximum accuracy of 0.893, 67 of the $4{,}083$ mixtures tie exactly, because with only 56 evaluation examples, accuracy is quantized to steps of $1/56 \approx 1.8\%$. MRPC and SciTail each have 2 mixtures tied at their maximums, and the remaining 9 target tasks have unique optima. We break ties toward the smaller mixture throughout, so any recovery rate computed against CB's optimum should be read with this in mind.

C.1 Distribution of the Scaling Coefficient

Across all $48{,}996$ (mixture, target) winning pairs, λ concentrates sharply toward the low end of the $\{0.05, \ldots, 2.00\}$ grid, as reported in Table 5. In total, 79% of winners fall below $\lambda = 0.2$, and the highest winning value observed anywhere in the sweep is 1.80. The top half of the grid, which accounts for 20 of 40 points and half of the compute spent per evaluated mixture, yields essentially no winners. This is not an artifact of clipping. When we restrict the analysis to each target's top-decile mixtures, only 9.3% sit at the grid minimum, compared to 36% overall, indicating that low-λ pinning is concentrated among mixtures that do not transfer well in the first place. Per-target behavior differs sharply, since BoolQ pins 90% of its mixtures at the grid minimum while CB pins under 1%. A single shared λ range therefore does not suit every target equally well, and a narrower grid would recover the same winners at a fraction of the cost.

Table 6: Recovery rate and speed by GP feature representation, aggregated over 12 target tasks and 10 seeds (63-evaluation budget).

GP features	found true best	evals to within 1%
Binary subset vector	**83%**	**29.1**
Binary + iso-SAR	82%	24.1
iso-SAR	48%	31.4
iso-SAR (normalized)	22%	32.4
SAR	15%	47.9
SAR (normalized)	12%	38.8
Random search	7%	51.1

D Arity Dependence of Subspace Alignment

SAR increases with mixture size because a larger summed task vector spans a wider subspace and is more likely to capture the directions of any given target vector through sheer coverage, regardless of whether that target actually benefits from those sources. The argmax of SAR therefore tends toward the full 12-source set, which Table 3 shows is never optimal. However, normalizing by mixture size does not rescue it, as Table 6 shows that arity-normalized features recover the optimum less often than unnormalized ones. This suggests that removing the arity confound also removes the signal, rather than isolating a cleaner one.

We also report only per-target correlations, and never a pooled correlation across target tasks. Pooling all 12 target tasks together gives $\rho = 0.53$ for SAR and 0.59 for iso-SAR, which is roughly triple the per-target median. This is an artifact of target tasks differing in overall accuracy level, as mean accuracy spans 0.31 to 0.86 across target tasks. Such a difference in manufacturing between-target correlation says nothing about ranking mixtures *within* a target, and ranking within a target is the only operation that selection performs.

E Per-Target Correlation of the Training-Free Selection Baselines

Table 7 reports the Spearman correlation between each training-free selection baseline in Section 5.1 and transfer accuracy, computed for each target across all $4{,}083$ mixtures. The last column is the median over target tasks, as reported in Table 4. Section 5.3 describes these correlations only in general terms, and this table reports the individual values.

The variation within each row is what matters here. Every selection baseline score is negative on at least one target task. SAR varies the most, from -0.38 on CoLA to 0.94 on RT, so its median of 0.17 does not describe any of its individual target tasks well. The Knowledge Consistency baselines remain near zero in the median, as a sanity-check baseline should, while still reaching 0.72 on CB. This shows that a target-independent score can correlate strongly on a single target task by chance.

Table 7: Per-target Spearman correlation ρ between each training-free selection baseline and transfer accuracy, over all $4{,}083$ mixtures within a target. The final column is the median over the 12 target tasks and reproduces the "med. ρ" column of Table 4. We report exact match except on STS-B, which uses Pearson correlation.

Basline	CR	RT	CB	SciTail	RTE	COPA	MultiRC	PIQA	STS-B	BoolQ	MRPC	CoLA	median
Target Similarity	0.52	0.70	0.75	0.39	0.08	0.63	0.54	0.44	0.05	0.14	−0.53	0.34	0.42
iso-SAR	0.72	0.83	0.45	0.26	0.13	0.55	0.66	0.55	0.16	0.39	−0.51	−0.19	0.42
Cosine Similarity	0.77	0.88	0.47	0.28	0.19	0.63	0.64	0.69	0.32	0.23	−0.39	0.33	0.40
SAR	0.89	0.94	−0.10	0.11	0.23	0.12	0.39	0.35	0.06	0.44	−0.16	−0.38	0.17
Dot Product	0.43	0.60	0.69	0.38	0.18	0.29	0.42	0.41	0.05	−0.22	−0.49	0.20	0.34
Knowledge Consistency	−0.65	−0.54	0.72	0.34	0.03	−0.34	0.09	−0.31	−0.08	−0.35	−0.20	0.19	−0.14

F BOTAS: DETAILED RESULTS AND SPECIFICATION

Table 6 reports recovery rate and convergence speed aggregated over all target tasks and seeds. Tables 8, 9, 10 and 11 break those numbers down by target, the last of them reporting the difference to the optimum, and Figure 2 shows the best accuracy each feature representation reaches within budget.

The binary subset indicator is the only representation that performs well on every target. It recovers the optimum on at least 8 of 10 seeds for 9 of the 12 target tasks, and its weakest target is CR at 20%, where the accuracy it reaches is nonetheless within 0.003 of the optimum. The alignment features are far more uneven. iso-SAR recovers the optimum on all 10 seeds for 5 target tasks, yet never recovers it on CR, SciTail, RTE, STS-B, BoolQ or CoLA, and SAR never recovers it on 9 of the 12 target tasks. Random search is competitive only on CB, where 67 mixtures tie at the optimum (Section 5.2), and on which it succeeds in 60% of runs.

Table 8: Per-target recovery rate, as the percentage of 10 seeds in which the exact optimum $S^\star$ was found within the 63-evaluation budget.

	CR	RT	CB	SciTail	RTE	COPA	MultiRC	PIQA	STS-B	BoolQ	MRPC	CoLA
Binary subset vector	20	90	100	80	80	60	100	100	80	100	90	90
Binary + iso-SAR	20	100	100	80	90	70	100	90	50	90	100	90
iso-SAR	0	100	70	0	0	100	100	100	0	0	100	0
iso-SAR (norm.)	0	0	0	0	0	100	0	70	0	0	90	0
SAR	0	100	80	0	0	0	0	0	0	0	0	0
SAR (norm.)	0	0	30	0	0	10	0	0	0	0	100	0
Random search	0	0	60	0	0	10	0	0	0	0	10	0

Table 9: Per-target number of evaluations needed to reach within 1% of $f(S^\star)$, averaged over 10 seeds. A value of 63.0 means the budget was exhausted without reaching that threshold.

	CR	RT	CB	SciTail	RTE	COPA	MultiRC	PIQA	STS-B	BoolQ	MRPC	CoLA
Binary subset vector	13.5	8.8	12.2	28.5	33.6	33.6	17.8	16.2	36.6	21.5	32.7	27.9
Binary + iso-SAR	15.9	10.5	10.7	26.5	29.6	40.3	16.7	15.7	48.0	20.5	30.0	24.7
iso-SAR	5.6	3.9	36.0	57.0	60.5	5.7	9.3	4.8	63.0	63.0	5.1	63.0
iso-SAR (norm.)	8.8	4.7	63.0	57.0	63.0	5.2	13.0	5.3	63.0	60.9	23.6	21.2
SAR	5.0	4.5	34.3	57.0	63.0	63.0	63.0	63.0	63.0	40.5	63.0	55.9
SAR (norm.)	6.2	4.9	53.1	51.5	63.0	53.1	47.5	17.1	63.0	31.1	12.4	63.0
Random search	31.1	35.4	36.8	42.2	53.3	62.2	58.6	54.8	63.0	57.5	59.0	59.3

Table 10: Per-target best transfer accuracy found within the 63-evaluation budget, averaged over 10 seeds, against the true optimum from the exhaustive sweep.

	CR	RT	CB	SciTail	RTE	COPA	MultiRC	PIQA	STS-B	BoolQ	MRPC	CoLA
Binary subset vector	0.931	0.929	0.893	0.863	0.839	0.759	0.757	0.721	0.689	0.699	0.683	0.672
Binary + iso-SAR	0.930	0.929	0.893	0.863	0.841	0.755	0.757	0.721	0.673	0.698	0.684	0.674
iso-SAR	0.931	0.929	0.887	0.826	0.815	0.770	0.757	0.721	0.612	0.657	0.684	0.627
iso-SAR (norm.)	0.929	0.927	0.859	0.840	0.824	0.770	0.756	0.721	0.675	0.681	0.678	0.668
SAR	0.928	0.929	0.889	0.833	0.808	0.750	0.724	0.696	0.601	0.690	0.558	0.642
SAR (norm.)	0.927	0.927	0.868	0.842	0.824	0.729	0.750	0.720	0.642	0.692	0.684	0.634
Random search	0.926	0.920	0.886	0.853	0.827	0.719	0.728	0.682	0.623	0.660	0.643	0.644
True optimum	0.934	0.929	0.893	0.865	0.841	0.770	0.757	0.721	0.705	0.699	0.684	0.674

F.1 BO BUDGET JUSTIFICATION

The budget is the cost of the method, since every evaluation is one merge followed by one pass over the target evaluation set, so we set N to the smallest value at which the result of this paper is already established rather than to the largest value we could afford. Table 12 reports how often the optimum is recovered as the budget grows. At 20 evaluations, which is 0.5% of the search space, the binary indicator recovers the optimum on 28% of runs, and at 40 evaluations on 59%, so half a percent of the space is not sufficient for the surrogate to have converged on most target tasks. At the cap the curve is still rising, from 78% at 50 evaluations to 83% at 63, so a larger budget would only strengthen the reported figure and we therefore prefer the smaller one. Matching that 83% by uniform sampling instead requires $3{,}176$ of the $4{,}083$ candidates, roughly 50 times the budget. We

Table 11: Difference to the optimum, that is, each target's best available score minus the score actually reached within the 63-evaluation budget, averaged over 10 seeds. This is Δ of Eq. 3, the same quantity Table 4 reports for the training-free selection baselines, so the two are directly comparable. A difference of 0.000 means the optimum was reached on every seed. The lower block adds the training-free selection baselines of Section 5.3, which score all $4{,}083$ mixtures rather than 63, so every method in the table is on one scale. The last column averages over the 12 target tasks, and we report exact match except on STS-B, which uses Pearson correlation.

	CR	RT	CB	SciTail	RTE	COPA	MultiRC	PIQA	STS-B	BoolQ	MRPC	CoLA	mean
Bayesian optimization, 63 evaluations													
Binary subset vector	0.002	0.000	0.000	0.002	0.002	0.011	0.000	0.000	0.016	0.000	0.001	0.002	0.003
Binary + iso-SAR	0.004	0.000	0.000	0.002	0.000	0.015	0.000	0.001	0.032	0.001	0.000	0.000	0.005
iso-SAR	0.003	0.000	0.005	0.039	0.026	0.000	0.000	0.000	0.093	0.042	0.000	0.047	0.021
iso-SAR (norm.)	0.004	0.002	0.034	0.025	0.017	0.000	0.001	0.000	0.030	0.018	0.006	0.006	0.012
SAR	0.005	0.000	0.004	0.032	0.033	0.020	0.033	0.025	0.104	0.009	0.126	0.032	0.035
SAR (norm.)	0.007	0.002	0.025	0.023	0.017	0.041	0.007	0.002	0.063	0.007	0.000	0.040	0.019
Random search	0.008	0.009	0.007	0.012	0.014	0.051	0.029	0.039	0.082	0.039	0.041	0.030	0.030
Training-free selection baselines, all $4{,}083$ mixtures scored													
Target Similarity	0.013	0.002	0.089	0.064	0.314	0.000	0.004	0.002	0.372	0.073	0.368	0.469	0.147
iso-SAR	0.013	0.002	0.554	0.278	0.112	0.040	0.023	0.002	0.226	0.109	0.368	0.132	0.155
Cosine Similarity	0.013	0.002	0.554	0.261	0.159	0.010	0.023	0.002	0.354	0.109	0.363	0.107	0.163
SAR	0.008	0.005	0.286	0.269	0.144	0.120	0.057	0.188	0.506	0.092	0.360	0.139	0.181
Dot Product	0.051	0.057	0.089	0.109	0.260	0.250	0.218	0.589	0.354	0.592	0.368	0.142	0.257
Knowledge Consistency	0.681	0.895	0.089	0.064	0.314	0.760	0.086	0.721	0.417	0.137	0.338	0.469	0.414

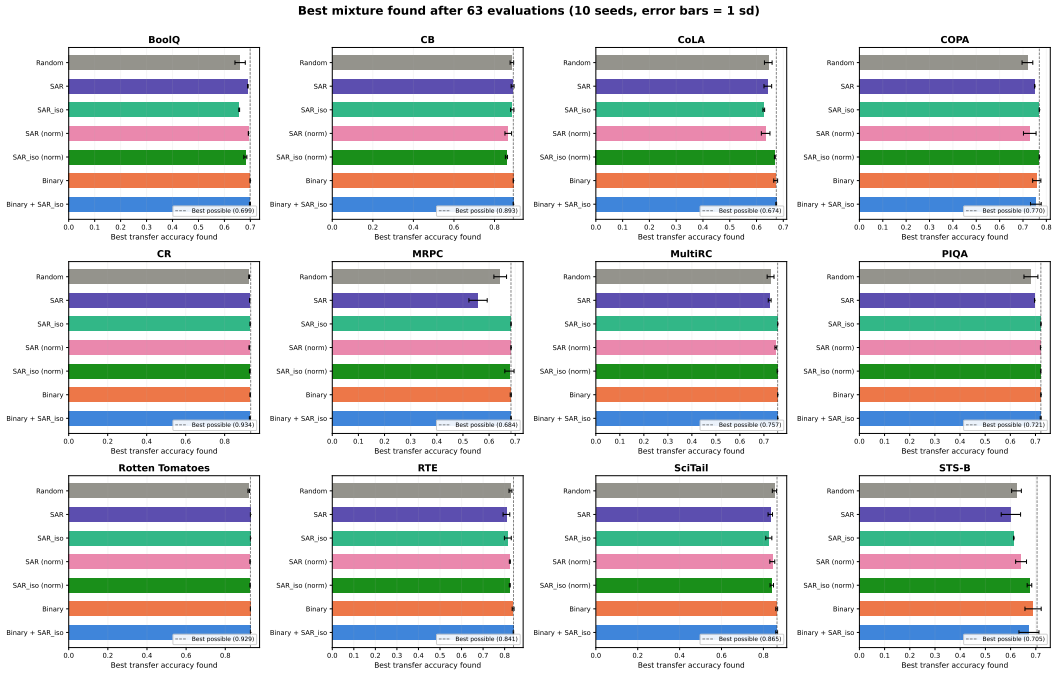


Figure 2: Best transfer accuracy found within a 63-evaluation budget, per target and feature representation, averaged over 10 seeds. Error bars show one standard deviation. The dashed line marks each target's true optimum from Section 5.2.

set $n_0 = 3$ because it is the smallest initial design the surrogate can be fit from, which leaves 60 of the 63 evaluations under its control, and because the paired random baseline replays the same n_0 draws, so a larger initial design would spend more of the budget on a prefix that the two methods share. Neither method recovers the optimum within the initial design on any target task, so every recovery we report is attributable to the guided phase of the search.

Table 12: Recovery of the accuracy-optimal mixture as a function of the evaluation budget N, aggregated over 12 target tasks and 10 seeds. Share is the fraction of the $4{,}083$ candidate mixtures scored before choosing. $N = 3$ is the initial design, evaluated before the surrogate guides the search.

N	Share of space	BOTAS	Random search
3	0.07%	0%	0%
5	0.12%	2%	0%
10	0.24%	8%	1%
20	0.49%	28%	3%
30	0.73%	45%	5%
40	0.98%	59%	6%
50	1.22%	78%	6%
63	1.54%	83%	7%