DOI: https://doi.org/10.15276/aait.09.2026.38
UDC 004.8:004.89:378.147

Compatibility-based source-pair screening for language model merging: An empirical evaluation

Tran Nhat Linh Pham[1)]
ORCID: https://orcid.org/0009-0008-9245-3387; linhptn9746@ut.edu.vn. Scopus Author ID: 60219414200
Thi Khanh Tien Nguyen[1)*]
ORCID: https://orcid.org/0000-0001-5379-7226; tienntk@ut.edu.vn. Scopus Author ID: 58735109800
The Vinh Tran[1)]
ORCID: https://orcid.org/0000-0002-4241-1065; vinhtt@ut.edu.vn. Scopus Author ID: 56835334700
[1] Ho Chi Minh City University of Transport, 2, Vo Oanh Str., Ho Chi Minh City, 700000, Vietnam
*Corresponding author. E-mail: tienntk@ut.edu.vn

ABSTRACT

Relevance. Expanding repositories of specialized language model adapters make exhaustive source-pair evaluation costly. **Purpose.** This study evaluates whether pre-merge compatibility descriptors provide predictive and selection value beyond domain composition and direct feature ranking. **Methods.** Twelve low-rank adapters sharing a pretrained backbone yield sixty-six conversational and mathematical candidate pairs. Task-vector direction, sign interference and centered kernel alignment predict merged-model benchmark scores. Evaluation includes training-only cross-validation, sixteen held-out pairs, feature ablation, same-domain prediction and adapter-, configuration- and recipe-disjoint holdouts. **Results.** Linear regression achieves held-out correlations of zero point eight five eight for conversation and zero point eight zero one for mathematics. However, a pair-type baseline reduces mean absolute error from zero point zero two four eight three to zero point zero zero seven eight one for conversation, and from zero point zero one nine zero eight to zero point zero one zero zero eight for mathematics. Adding compatibility to pair type modestly improves mathematical error but worsens conversational error. Both the predictor and representation similarity recover the best held-out pair with a single selection. Grouped results reveal unstable calibration and no consistent selection advantage; same-domain correlations are negative. **Conclusions.** Domain composition explains much of aggregate predictability in this repository. Compatibility-based screening has repository-specific ranking value, but supervised calibration does not establish superiority over simpler baselines. Cost reductions remain conditional projections because successful evaluation-runtime measurements are unavailable. Candidate growth remains quadratic, and calibration, caching and transfer determine practical savings.

Keywords: language model merging; source-pair screening; compatibility prediction; candidate ranking; adapter screening; language models

***For citation*: Pham Tran Nhat Linh, Nguyen Thi Khanh Tien*, Tran The Vinh. "Compatibility-based source-pair screening for language model merging: An empirical evaluation". *Applied Aspects of Information Technology.* 2026; Vol.9 No.4: 585–600. DOI: https://doi.org/10.15276/aait.09.2026.38**

INTRODUCTION

Large Language Models (LLMs) have achieved remarkable success across a wide range of natural language processing tasks and have been successfully deployed in diverse downstream applications, including educational assistants [1], strategic reasoning systems [2], intelligent recommendation systems [3], and other domain-specific AI applications. However, adapting a foundation model independently to each downstream application remains computationally expensive and storage-intensive. Parameter-efficient fine-tuning (PEFT) methods, particularly Low-Rank Adaptation (LoRA), address this challenge by learning only a small set of task-specific parameters while keeping the pretrained backbone unchanged [4]. As a result, organizations and open-source communities increasingly maintain large repositories of LoRA adapters specialized for different domains, tasks, and training configurations. To exploit these independently trained adapters, language model merging has emerged as an attractive alternative to repeated multi-task fine-tuning. Rather than retraining a single model on multiple tasks, model merging combines parameter updates from independently fine-tuned models to construct a unified model capable of inheriting complementary capabilities [5]. Representative approaches include Task Arithmetic [6], TIES-Merging [7], DARE [8], Model Stock [9], and adaptive merging strategies [10]. They address parameter interpolation, conflict resolution, sparsification or geometric aggregation. Adaptive and evolutionary approaches also optimize merging choices using task feedback; source selection is therefore not universally absent from prior work [10], [11].

© Pham L., Nguyen T*., Tran V. 2026

This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/deed.uk)

In practice, large adapter repositories may contain hundreds or even thousands of candidate models. For a repository containing N adapters, exhaustive pairwise evaluation requires $N(N-1)/2$ candidate merges, causing computational cost to grow quadratically with repository size. Consequently, source-pair screening should be viewed as an independent optimization problem that precedes model merging rather than as an implicit preprocessing step. The scientific question is whether measured compatibility contains information about downstream quality beyond task-domain membership; the engineering question is whether that information justifies screening and calibration costs.

Three compatibility metrics are investigated: task-vector cosine similarity, which measures directional agreement between parameter updates; interference rate, which quantifies sign conflicts among important parameters; and Centered Kernel Alignment (CKA), which measures representational similarity between independently fine-tuned models. These compatibility features are used to predict expected merge quality, rank candidate source-model pairs, and select candidates for computationally expensive merging and downstream evaluation. Their complementarity and selection value are hypotheses tested by ablation and baseline comparison, rather than assumptions.

The framework is evaluated using twelve LoRA adapters independently fine-tuned from Gemma-2-2B-IT for Vietnamese conversational generation and mathematical reasoning. Comprehensive experiments include regression model comparison, held-out pair evaluation, compatibility feature analysis, Top-k source-pair screening, comparison with baseline strategies, and computational cost analysis. Additional controls test domain composition, same-domain discrimination, new adapter endpoints, configuration separation and coefficient sensitivity.

The main contributions are: (i) a source-pair prediction and ranking formulation that separates screening from a fixed merging operator on a frozen, fully labelled pair repository; (ii) a controlled empirical comparison that separates aggregate prediction, same-domain discrimination, grouped holdouts and shortlist utility, showing that domain composition explains much of aggregate predictability while fine-grained and configuration-disjoint selection remain limited; and (iii) an analysis of feature sensitivity and conditional computational cost under declared runtime assumptions. The contribution is an assessment of screening capabilities and limits on this repository, not a demonstrated universally superior selection algorithm.

LITERATURE REVIEW AND PROBLEM STATEMENT

Language model merging. Language model merging combines independently fine-tuned models in parameter space to construct a multi-capability model without repeating full multi-task training [5]. Task Arithmetic represents fine-tuned models as task vectors relative to a shared pretrained model and combines these vectors through linear operations [6]. TIES-Merging improves this process by trimming low-magnitude updates, resolving sign disagreement, and merging consistent parameters [7]. DARE further reduces redundancy through random pruning and rescaling [8], whereas Model Stock formulates a geometric weight-averaging strategy [9].

Model soups and Fisher-weighted averaging combine fine-tuned parameters using averaging and parameter importance [12], [13]. Git Re-Basin addresses permutation symmetries [14]. Tangent-space arithmetic and singular-value-based merging investigate alternative update structure [15], [16]. Magnitude-based sampling, task-subspace matching and sparse masks further extend this design space [17], [18], [19]. Dataless weight fusion, rewarded soups, model-exclusive arithmetic and Localize-and-Stitch address knowledge combination, reward trade-offs or localized updates [20], [21], [22], [23].

These methods provide aggregation rules or search procedures, but do not by them establish the predictive value of the three descriptors evaluated here. Adaptive and evolutionary methods can use evaluation feedback to optimize merging choices [10], [11]. The narrower question is whether inexpensive pre-merge measurements can rank pairs for a fixed operator without evaluating every candidate at selection time. Supervised screening still needs labelled calibration examples, and no benefit is assumed for an entirely unlabelled repository.

Compatibility analysis and research gap. Relationships between neural models can be examined in both parameter and representation spaces. Task-vector cosine similarity measures whether fine-tuning updates point in similar directions and thus provides a global indication of parameter alignment [6]. Interference measures derived from sign disagreement quantify conflicts among influential parameter updates, a mechanism directly related to the conflict-resolution stage of

TIES-Merging [7]. Centered Kernel Alignment (CKA), by contrast, compares hidden representations and is invariant to isotropic scaling and orthogonal transformations [24].

Existing compatibility measures are primarily used to characterize model relationships or explain merging behavior. However, descriptive similarity does not necessarily translate into an effective source-pair screening strategy. A practical source-pair screening method must determine whether lightweight pre-merge compatibility measures can accurately predict downstream merge quality, whether multiple measures provide complementary information, and whether the computational cost of compatibility screening is substantially lower than exhaustive merge-and-evaluation. Domain composition is an essential control because descriptors may distinguish task families without resolving differences among adapters in the same family.

Problem statement. Consider a repository containing N LoRA adapters derived from a shared pretrained model. Exhaustive source-pair screening requires evaluating $M = N(N-1)/2$ candidate adapter pairs, where each candidate normally undergoes model merging followed by downstream benchmark evaluation. This work formulates source-pair screening as a compatibility prediction and ranking problem. The objective is to estimate task-specific merge quality, rank candidate pairs and forward a Top-k shortlist to merging and evaluation. The central question is whether supervised compatibility prediction improves prediction or selection beyond pair type and direct feature ranking, and under which calibration and reuse assumptions it can reduce cost.

RESEARCH AIM AND OBJECTIVES

The aim is to evaluate compatibility-based source-pair screening before language model merging. Method-development objectives are to formulate task-specific prediction and ranking and to define parameter- and representation-space descriptors. Experimental-validation objectives are to compare predictors and baselines, test feature combinations and same-domain discrimination, examine held-out identities and configurations, and separate measured selection outcomes from conditional cost projections.

MATERIALS AND METHODS

Screening workflow. The framework consists of six sequential stages: source-pair construction; compatibility feature extraction; compatibility prediction; candidate ranking and Top-k source-pair screening; model merging; and downstream evaluation. Figure 1 separates calibration from subsequent screening. During benchmark construction, all pairs are merged and evaluated to provide targets; shortlist-only evaluation describes prospective use after calibration, not how the retrospective benchmark labels were obtained.

Source models and candidate pairs. Since compatibility prediction operates on pairs of source models, the benchmark must first define the source adapter repository and then enumerate all candidate pairs for subsequent compatibility analysis and merge evaluation.

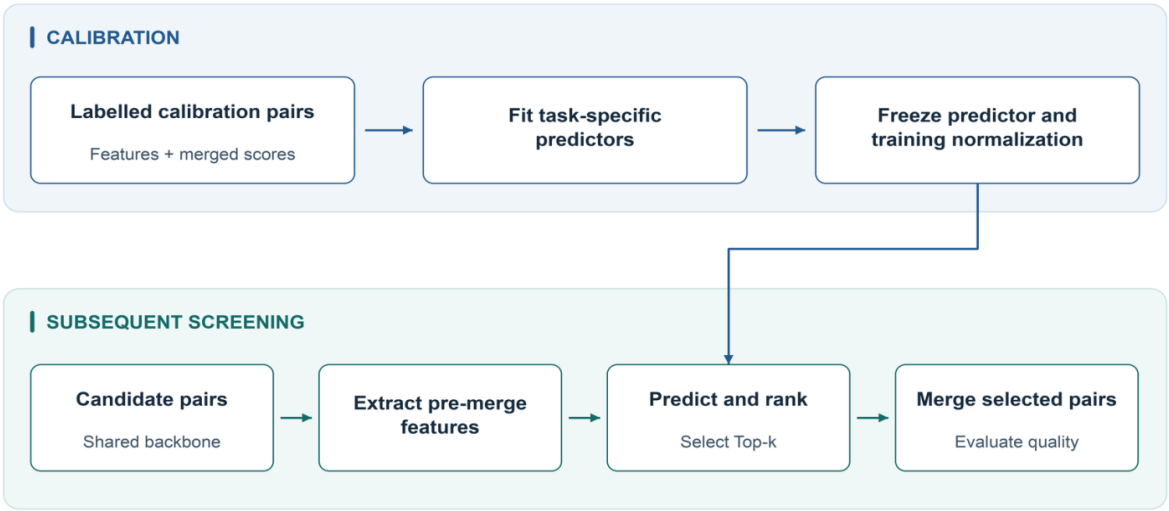


Fig. 1. **Calibration requires labelled merges; prospective screening ranks candidates before selected merges**
Source: **compiled by the authors**

Twelve LoRA adapters were independently fine-tuned from the base model, comprising six adapters for Vietnamese conversational generation and six adapters for Vietnamese mathematical reasoning. Gemma-2-2B-IT is the shared backbone [25]. The chat training data are vi_alpaca and the mathematical training data are vi_gsm8k. To construct a heterogeneous repository of source models, three commonly used LoRA fine-tuning hyperparameters were intentionally varied: training epochs, learning rate and rank. Each configuration in Table 1 is applied to both domains.

Table 1. **Training configurations used to construct the source LoRA adapter repository**

Config	Training Epochs	Learning Rate	LoRA Rank	Generated Adapters
C1	1	2×10^{-4}	8	Chat, Math
C2	1	2×10^{-4}	32	Chat, Math
C3	2	1×10^{-4}	8	Chat, Math
C4	2	1×10^{-4}	32	Chat, Math
C5	2	2×10^{-4}	16	Chat, Math
C6	3	1×10^{-4}	16	Chat, Math

Source: **compiled by the authors**

The repository yields 66 unordered pairs: 15 chat-chat, 36 chat–math and 15 math–math. Each merged model is evaluated on Vietnamese conversational generation and mathematical reasoning; benchmark statistics are summarized in Table 2.

Compatibility feature extraction. Each candidate source model pair is represented by three lightweight compatibility metrics computed prior to model merging. These metrics characterize aspects of similarity in both parameter space and representation space and together form the feature vector used for compatibility prediction

$$\mathbf{x}_{ij} = \left[s_{ij}^{\cos}, r_{ij}^{\text{int}}, s_{ij}^{\text{cka}}\right]^{\mathsf{T}} \quad (1)$$

Here $s_{ij}^{\cos}$ is task-vector cosine similarity, r_{ij}^{int} is interference rate, and s_{ij}^{cka} is representational similarity measured by CKA. Table 3 summarizes their intended interpretations. Cosine may conceal localized conflicts; interference discards magnitudes after masking; CKA depends on inputs, layer and pooling. Their predictive utility and redundancy are tested empirically.

Task vectors are defined as parameter differences between a fine-tuned model and the common pretrained model [6]. Cosine similarity measures the directional agreement between the task vectors of two source models:

$$s_{ij}^{\cos} = \frac{\Delta W_i \cdot \Delta W_j}{\|\Delta W_i\| \|\Delta W_j\|}. \quad (2)$$

Table 2. **Benchmark dataset summary**

Item	Value
Base model	Gemma-2-2B-IT
LoRA adapters	12
Candidate source pairs	66
Chat–Chat pairs	15
Chat–Math pairs	36
Math–Math pairs	15
Merge algorithm	TIES
Chat benchmark	ROUGE-L
Math benchmark	Exact Match

Source: **compiled by the authors**

Table 3. **Compatibility features**

Feature	Space	Interpretation
Cosine similarity	Parameter	Directional alignment of task vectors
Interference rate	Parameter	Sign conflicts in important parameter updates
CKA	Representation	Functional similarity of hidden representations

Source: **compiled by the authors**

The recorded extraction reconstructs dense float32 updates as $(\alpha/r)BA$ for q_proj, k_proj, v_proj and o_proj, aligns common layer names and flattens updates. Thus different LoRA ranks are compared in the same dense parameter coordinates, rather than by comparing factor matrices of different dimensions. Global cosine uses the aligned dense updates. Adapter-specific scaling values must be read from the corresponding adapter configurations; the available study summary does not enumerate them. Higher cosine similarity indicates stronger alignment between parameter updates; it does not itself establish higher merged quality. Although two task vectors may exhibit similar global directions, destructive interference can still occur at individual parameters.

Inspired by TIES-Merging [7], the interference rate measures sign disagreement between important parameter updates. For each shared layer, entries strictly above the $(1-\rho)$ magnitude quantile are retained in each update, with $\rho = 0.5$. Let Ω be the union, across aligned layers, of the intersections of these masks. Quantile ties can retain fewer than half the entries.

The pooled count implements overlap-weighted aggregation:

$$r_{ij}^{\text{int}} = \begin{cases} \frac{1}{|\Omega|}\sum_{p\in\Omega} \mathbb{I}\left[\text{sign}\left(\Delta W_i^{(p)}\right) \neq \text{sign}\left(\Delta W_j^{(p)}\right)\right], & |\Omega| > 0, \\ 0, & |\Omega| = 0 \end{cases}. \quad (3)$$

A lower value indicates fewer conflicting updates among jointly retained coordinates. The zero-overlap value is an implementation convention and should not be interpreted as evidence of compatibility.

CKA measures the similarity between hidden representations produced by two source models on the same input samples [24]. Given centered activation matrices X and Y, linear CKA is computed as:

$$s_{ij}^{\text{cka}} = \frac{\|X^\top Y\|_F^2}{\|X^\top X\|_F \, \|Y^\top Y\|_F}. \quad (4)$$

Here $\|\cdot\|_F$ denotes the Frobenius norm. Higher CKA values indicate greater representational similarity between the source models. Unlike parameter-space metrics, CKA measures representational similarity and can identify functionally similar models even when their parameter updates differ substantially.

The extraction notebook call uses the final hidden layer, mean pooling over unmasked tokens, batch size one, a maximum input length of 256 tokens and bfloat16 model weights, returning float32 pooled arrays. A currently reconstructed set contains 106 chat and 94 math prompts, including 127 evaluation prompts. This reconstruction does not verify the identity of the historical activation cache that generated the frozen CKA features. The analyses reuse stored features; they do not re-extract historical representations. Benchmark-input reuse is transductive even without target-score access, and a separate calibration-prompt validation remains necessary.

Compatibility prediction models. Given the compatibility feature vector $\mathbf{x}_{ij}$, the objective is to learn a task-specific regression function:

$$f_t: \mathbb{R}^d \to \mathbb{R}, \qquad \hat{y}_{ij}^{(t)} = f_t(\boldsymbol{x}_{ij}), \quad t \in \{chat, math\}. \quad (5)$$

The full representation has $d = 3$. Separate targets are defined for conversational generation and mathematical reasoning, enabling task-specific quality to be estimated independently before aggregation for selection. Five representative regression algorithms are investigated: Linear Regression, Ridge Regression, Random Forest, Extra Trees and XGBoost [26], [27]. Linear Regression provides an interpretable baseline; Ridge regularizes coefficients; the tree ensembles model nonlinear relationships. Their roles are summarized in Table 4.

Table 4. **Compatibility prediction models**

Model	Characteristics
Linear Regression	Linear baseline
Ridge Regression	L2 regularization
Random Forest	Bagging ensemble
Extra Trees	Extremely randomized trees
XGBoost	Gradient boosting

Source: **compiled by the authors**

Model merging and ground-truth generation. TIES-Merging is a task-vector-based method that mitigates parameter interference through three operations: task-vector trimming, sign-conflict resolution, and parameter averaging [7]. It is adopted as the reference merging algorithm throughout this study, with historical density 0.5. The frozen merge outputs supply supervised targets. The reported evidence is conditional on those outputs; it does not establish transfer to a different merging operator.

Each merged model is subsequently evaluated on two representative Vietnamese downstream tasks. Conversational generation is assessed on the vi_alpaca benchmark using ROUGE-L, while mathematical reasoning is evaluated on vi_gsm8k using Exact Match (EM). Both benchmarks contain 200 evaluation samples.

The recorded ROUGE-L implementation computes longest-common-subsequence F1 on lowercased whitespace tokens [28]. Mathematical evaluation removes commas and compares the final strings matched by a signed-integer/decimal regular expression in prediction and reference; missing numbers fail. Arithmetic equivalence is not applied, so this is a numeric-string matching target rather than a general mathematical correctness measure. Frozen scores are preserved without regenerating outputs.

Current dataset snapshots were checked for exact-text and user-prompt overlap between selected evaluation records and corresponding current training files, with no overlap reported. This does not establish absence of overlap in unavailable historical snapshots. Immutable historical dataset/tokenizer revisions, a complete decoding specification and all merge-weight details are not established by the available record. The reproducible scope is the statistical reanalysis of frozen features and targets; end-to-end regeneration of historical benchmarks remains a limitation.

EXPERIMENTAL DESIGN

Experimental setup. The repository and merging operator are fixed throughout prediction analysis. Linear and Ridge Regression use fold-local standardization; Ridge has a regularization coefficient of 1.0. Random Forest and Extra Trees each use 200 trees. XGBoost uses 100 trees, maximum depth three, learning rate 0.05, and row/column subsampling of 0.8. Stochastic estimators use seed 42 and one CPU worker. These frozen configurations are evaluated without held-out tuning [26], [27].

The historical merge/evaluation workstation is reported as an NVIDIA GeForce RTX 5070 Ti system using Python, PyTorch, Transformers and PEFT. Statistical reanalysis uses the CPU with scikit-learn 1.6.1 and XGBoost 3.0.2. Separate API smoke checks used synthetic merges and four-sample generation; they do not validate the full historical benchmark or provide successful benchmark-runtime measurements. Table 5 summarizes this historical experimental configuration and software stack.

Prediction protocols and domain-composition controls. The historical notebook benchmark used LOOCV on all 66 pairs, despite a training-only label. It is retained only as a reproduction result. The revision protocol uses the frozen 50 training pairs for LOOCV and fits preprocessing within each fold. The fixed Linear Regression predictor is fitted on those 50 pairs and tested on 16 held-out pairs without test-based hyperparameter or feature selection. All 11 test adapters and all six configuration families appear in training; the split evaluates new pairs, not new source models. The seed is 42; train pair-type counts are 11/29/10 and test counts are 4/7/5.

Table 5. **Experimental configuration**

Component	Configuration
Base model	Gemma-2-2B-IT
Source adapters	12 LoRA adapters
Candidate pairs	66
Merge algorithm	TIES-Merging
Prediction models	LR, Ridge, RF, ET, XGB
GPU	NVIDIA RTX 5070 Ti
Framework	PyTorch + Transformers + PEFT

Source: **compiled by the authors**

Protocol A compares regressors using training-only leave-one-out cross-validation (LOOCV). Protocol B fits the fixed predictor on 50 training pairs and evaluates 16 held-out pairs. Protocol C evaluates ranking within explicitly defined candidate pools. Pearson correlation, Spearman correlation, MAE and R^2 describe different aspects of prediction; a positive correlation does not guarantee accurate score calibration or useful candidate selection.

Domain-composition controls compare four exported prediction models: training-mean, pair-type-only, compatibility-only and pair type plus compatibility. Pair type denotes chat–chat, chat–math or math–math. The reported protocol uses training-local fitted statistics and preprocessing. Constant training-mean predictions have undefined Pearson and Spearman correlation. The available summary identifies these feature sets and exported scores, but does not fully specify the categorical design and reference coding of the combined regression. This limits independent implementation-level reproduction of that control and is not resolved by the numerical tables alone.

All seven nonempty compatibility-feature combinations use a fixed Linear Regression predictor. Same-domain LOOCV is performed separately on 15 chat–chat pairs for chat and 15 math–math pairs for math. Conditional uncertainty uses 10,000 paired-row percentile bootstrap replicates on held-out predictions with seed 42. Zero-variance replicates are excluded and counted. These intervals exclude model-refitting uncertainty, and pairs sharing adapters are not independent observations.

Grouped evaluation. Leave-one-adapter-out excludes every training pair containing the held-out adapter (55 train and 11 test pairs). Adapter-disjoint evaluation uses all 225 balanced four-adapter holdouts (two chat and two math); train uses the remaining eight adapters, test uses only held-out endpoints, and cross-boundary pairs are excluded (28/6/32 pairs). The exported fold definitions are fixed for this revision analysis. Repeated folds overlap; fold dispersion is descriptive rather than a confidence interval. Identity-disjoint does not imply configuration-disjoint. Additional configuration-disjoint evaluation holds out all fifteen two-family combinations from the six recorded (epochs, learning rate, rank) families. Each fold has twenty-eight training, six test and thirty-two excluded boundary pairs. A stricter recipe-disjoint analysis groups ranks sharing epochs and learning rate into four recipes and exhaustively holds out two recipes in six folds. Training and test sizes range from six to twenty-eight pairs; thirty-two or thirty-six boundary pairs are excluded. Both protocols exclude shared adapter identities and configurations, and recipe holdouts also exclude the declared shared training

recipes. Standardization is fitted on training pairs only; the fixed Linear Regression predictor is not tuned. Fold means and standard deviations are descriptive because folds overlap. These discrete groups do not guarantee separation under every continuous hyperparameter-distance definition.

Leave-one-adapter-out evaluates a pair with one unseen endpoint and one previously represented endpoint. The identity-, configuration- and recipe-disjoint tests use held-out endpoints only. Their overlapping folds are diagnostic repeated partitions of a small fixed repository, not independent replications or population-level confidence intervals.

Ranking objectives and baselines. The predicted compatibility scores are normalized and aggregated according to the multi-task selection objective. For task t, define the positive normalization constant $a_t = \max_{p\in\text{train}} y_p^{(t)}$. Predicted and observed joint scores use the same training-derived constants:

$$\hat{q}_p = \frac{1}{2}\sum_{t\in\{chat,math\}} \frac{\hat{y}_p^{(t)}}{a_t}, \qquad q_p = \frac{1}{2}\sum_{t\in\{chat,math\}} \frac{y_p^{(t)}}{a_t}. \quad (6)$$

Candidate adapter pairs are ranked in descending order of predicted joint score. Only the Top-k ranked pairs are forwarded to model merging and downstream evaluation. For a candidate pool P and selected set $S_k(P)$, define:

$$q_P^* = \max_{p\in P} q_p,\ q_k^* = \max_{p\in S_k(P)} q_p, \quad RP_P(k) = \frac{q_k^*}{q_P^*} \quad . \quad (7)$$

$$\text{Regret}_P(k) = q_P^* - q_k^*, \quad H_P(k) = \mathbb{I}[q_k^* = q_P^*],$$
$$SRR_P(k) = 1 - \frac{k}{|P|}. \quad (8)$$

Retained performance assumes $q_p^* > 0$; it is reported as a ratio or percentage, as labelled. Regret is an absolute difference on the normalized joint-score scale. A hit means that at least one exact optimum is selected; all exact optimum ties count as hits. Marginal chat and math optima may belong to different pairs, so full joint retention does not imply recovery of each task optimum.

The main pool contains 16 held-out pairs. Predictor, cosine, interference, CKA and random selection use identical pools and budgets k = 1,2,3,5. Direct-feature directions are chosen from training data; deterministic ties use ascending pair ID. Random-subset expectations are computed exactly without replacement. With one optimum, the random hit probability is $k/|P|$. Grouped Top-1 results use fold-local normalization and direction choices. Tables do not establish a statistically significant difference between ranking methods. Pair-type-only and combined-model ranking were not reported in the available results; their selection utility and sensitivity to tie-breaking remain untested.

Historical all-66 ranking uses raw predicted-score averaging and global normalization of the observed objective. It mixes training and test candidates and is retained only as a distinct reproduction result, not as the held-out screening experiment.

Conditional computational cost. Computational cost is expressed as serial pipeline time rather than GPU-kernel time. For $M = N(N-1)/2$ pairs, feature time T_f, merge/evaluation time T_{me} and a predictor calibrated outside the current pool:

$$T_{\text{exhaustive}} = MT_{me}, \qquad T_{\text{screened}} = MT_f + kT_{me},$$
$$D = (M-k)T_{me} - MT_f \quad . \quad (9)$$

Here D is conditional per-pool saving before calibration. The scenario uses T_f = 1.67 min/pair from a feature timing record and $T_{me} = 733/66$ min/pair as an unverified historical runtime assumption. Successful full benchmark evaluation-time measurements are unavailable. Both workflows retain quadratic growth in N; fixed-k asymptotic speedup is bounded by T_{me}/T_f. The scenario does not establish measured GPU savings or retained quality at larger repository sizes.

For R distinct future candidate pools of equal size, external calibration cost C_{cal} and a predictor reused without refitting, incremental savings are $RD - C_{cal}$. A strict saving requires $D > 0$ and $R > C_{cal}/D$. This comparison assumes each new pool requires new baseline evaluations. Repeated execution on identical models, prompts and settings must instead permit both workflows to cache prior results.

If calibration labels are acquired from the pool being searched, let L be the evaluated pairs and u_k the number of unlabelled pairs in the shortlist. With cached features for L, sequential screening costs are:

$$T_{\text{same pool}} = MT_f + (|L| + u_k)T_{me} + T_{fit}. \quad (10)$$

This counts feature extraction once, reuses labelled scores and adds only previously unevaluated shortlisted pairs. The best already labelled pair should remain eligible for final selection. It is a cost model, not a newly evaluated

selection protocol. Calibration from the current pool and calibration on a separate repository must not be combined into a single claimed saving.

RESEARCH RESULTS

Training-only and held-out prediction. Historical all-66 LOOCV correlations of 0.852 for chat and 0.608 for math are reproduced separately. Table 6 presents training-only LOOCV on 50 pairs, with preprocessing fitted within each fold. Linear Regression has the highest Pearson correlation and R^2 for both targets; it does not have the lowest MAE for every target, as XGBoost has slightly lower chat MAE.

After fitting the fixed Linear Regression model on the complete 50-pair training set, held-out correlations are 0.858 for chat and 0.801 for math (Table 7). These are new pairs of previously represented adapters. The larger math correlation than in training-only LOOCV can reflect split composition and sampling variability; it is not evidence that new adapters are easier to predict.

Table 8 gives conditional bootstrap intervals. They describe the uncertainty of fixed held-out predictions under row resampling and should not be interpreted as uncertainty over new adapter populations. No invalid correlation replicate occurred.

Domain-composition controls. Table 9 compares the same held-out targets with domain-composition baselines. Pair type alone gives MAE 0.00781 for chat and 0.01008 for math, compared with 0.02483 and 0.01908 for compatibility-only regression. These are approximately 68.5 % and 47.2 % lower errors, respectively. Thus high aggregate compatibility correlation does not establish information beyond domain membership.

Adding compatibility to pair type changes chat MAE from 0.00781 to 0.00828 and math MAE from 0.01008 to 0.00938. This is a small numerical improvement in math MAE and R^2 on this split, with worse chat MAE. Spearman correlation declines for both tasks, from 0.91732 to 0.70588 for chat and from 0.93878 to 0.92468 for math. Improved score error therefore cannot be generalized to improved ranking. No uncertainty estimate for these paired differences is reported. Figure 2 visualizes the held-out mean absolute error of the four prediction controls in Table 9.

Table 6. **Training-only LOOCV on 50 pairs**

Model	Target	Pearson r	MAE	R^2
Linear	Chat ROUGE-L	0.847	0.022	0.713
Linear	Math exact match	0.553	0.019	0.293
Ridge	Chat ROUGE-L	0.631	0.032	0.393
Ridge	Math exact match	0.155	0.023	0.023
Random Forest	Chat ROUGE-L	0.698	0.023	0.476
Random Forest	Math exact match	0.311	0.021	0.024
Extra Trees	Chat ROUGE-L	0.661	0.024	0.417
Extra Trees	Math exact match	0.161	0.023	-0.192
XGBoost	Chat ROUGE-L	0.730	0.021	0.528
XGBoost	Math exact match	0.374	0.020	0.109

Source: **compiled by the authors**

Table 7. **Prediction on 16 held-out pairs**

Model	Target	Pearson r	MAE	R^2
Linear	Chat ROUGE-L	0.858	0.025	0.727
Linear	Math exact match	0.801	0.019	0.493
Ridge	Chat ROUGE-L	0.656	0.037	0.415
Ridge	Math exact match	0.641	0.025	0.127
Random Forest	Chat ROUGE-L	0.561	0.030	0.263
Random Forest	Math exact match	0.572	0.020	0.275
Extra Trees	Chat ROUGE-L	0.551	0.030	0.254
Extra Trees	Math exact match	0.542	0.020	0.268
XGBoost	Chat ROUGE-L	0.548	0.031	0.211
XGBoost	Math exact match	0.608	0.020	0.330

Source: **compiled by the authors**

Table 8. **Conditional 95 % percentile bootstrap intervals**

Target	Pearson r interval	MAE interval
Chat ROUGE-L	0.683-0.934	0.0170-0.0327
Math exact match	0.490-0.957	0.0135-0.0245

Source: **compiled by the authors**

Table 9. **Held-out domain-composition controls**

Model	Target	Pearson r	Spearman rho	MAE	R^2
Training mean	Chat ROUGE-L	undefined	undefined	0.0471	-0.0490
Training mean	Math numeric match	undefined	undefined	0.0267	-0.0547
Pair type	Chat ROUGE-L	0.9894	0.9173	0.0078	0.9699
Pair type	Math numeric match	0.9353	0.9388	0.0101	0.8526
Compatibility	Chat ROUGE-L	0.8581	0.4441	0.0248	0.7266
Compatibility	Math numeric match	0.8005	0.7105	0.0191	0.4928
Combined	Chat ROUGE-L	0.9875	0.7059	0.0083	0.9678
Combined	Math numeric match	0.9430	0.9247	0.0094	0.8799

Source: **compiled by the authors**

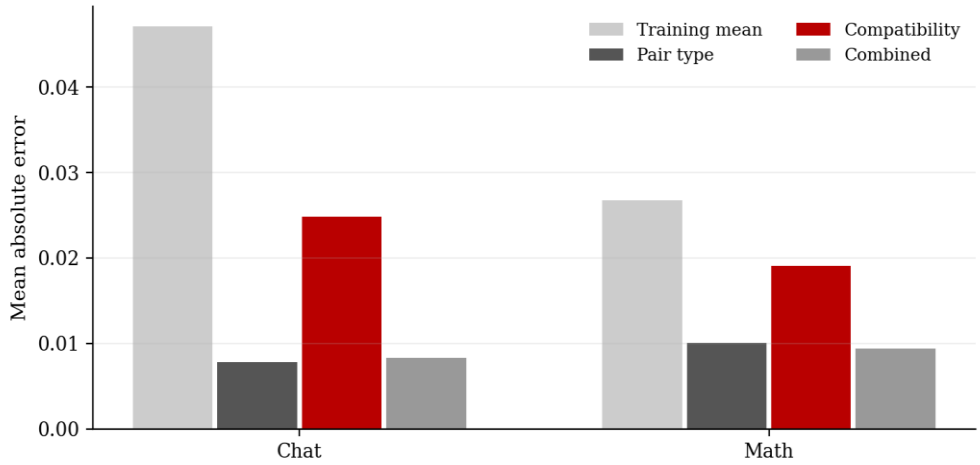


Fig. 2. **Held-out MAE for the four prediction controls**
Source: **compiled by the authors**

Feature ablation and same-domain discrimination. Centered Kernel Alignment (CKA) is also strongly correlated with cosine similarity (r = 0.900) and interference (r = −0.896), although less strongly than the two parameter-space metrics are correlated with each other. This pattern reflects both the shared pretrained backbone and the separation between same-domain and cross-domain pairs. These are descriptive full-repository associations rather than held-out performance.

Task-vector cosine similarity and interference rate exhibit an almost perfect negative correlation (r = −0.995), confirming that they encode nearly the same parameter-space structure with opposite orientation. Treating their regression coefficients or feature importances as independent evidence would therefore be misleading. Seven-way ablation in Table 10 shows that feature combinations are not uniformly beneficial.

Table 10. **Seven-way fixed-regressor ablation; held-out Pearson correlation**

Features	Chat ROUGE-L	Math exact match
CKA	0.507	0.086
Cosine	0.270	-0.003
Cosine + CKA	0.494	0.030
Cosine + interference	0.787	0.818
Cosine + interference + CKA	0.858	0.801
Interference	0.192	-0.086
Interference + CKA	0.608	0.282

Source: **compiled by the authors**

Cosine plus interference yields chat/math correlations of 0.787/0.818, compared with 0.858/0.801 for all three features. CKA improves the chat correlation but slightly reduces the math correlation in this split. These results do not support a claim that all three metrics are always necessary. The standardized two-feature design has a reported condition number of 19.41 for the held-out experiment, with grouped maxima of 41.09; correlated coefficients require cautious interpretation. The available perturbation summary reports unchanged held-out Top-1 after rounding raw features to four or more decimals and adding Gaussian noise at 0.01 % of training feature standard deviation. At 1 % noise, Top-1 changes in six of ten repeats. The summary does not establish whether these Top-1 checks use the two-feature or full three-feature predictor, where perturbations are applied, or whether fitting is repeated. These observations are exploratory sensitivity evidence and do not validate robustness of the main ranking model.

Same-domain LOOCV produces negative Pearson, Spearman and R^2 for chat–chat prediction of chat quality and math–math prediction of math quality (Table 11). Aggregate predictive association therefore does not establish reliable fine-grained discrimination among homogeneous adapters.

New-adapter and configuration robustness. Table 12 and Table 13 summarize grouped evaluation. Positive mean correlations coexist with negative mean R^2 in most settings. Leave-one-adapter-out chat R^2 is especially unstable. A negative test R^2 indicates error exceeding that of a constant predictor equal to that test fold's mean; it does not alone establish failure of ranking, nor is that test mean a deployable training-only baseline. The analysis summary reports lower pair-type-only MAE than compatibility-only across grouped protocols and notes positive median R^2 alongside very negative means for some combined-model settings. Numerical medians, quartiles and full per-fold coefficient exports are not included in the available table summaries, so they are not reconstructed here. The displayed means do not resolve outlier-fold behavior. Small test pools and shared observations across folds limit inference about generalization.

Held-out and grouped source-pair selection. Table 14a-c compares methods on the same 16-pair pool. Predictor and CKA both recover the optimum at Top-1; cosine first recovers it at Top-2 and interference at Top-3. At Top-5 all deterministic methods attain full retention. Exact random Top-5 retention is 95.447 % and optimum-hit probability is 31.25 %. Near-optimal score retention and exact optimum recovery are distinct outcomes.

Grouped Top-1 means are shown in Table 15a-c. CKA has higher mean retention and lower mean regret than compatibility in each protocol. This does not imply consistently higher hit probability: in configuration-disjoint folds, compatibility hits 13.33% of optima and CKA 6.67 %, both below the random expectation of 16.67 %. In recipe-disjoint folds both methods have zero hits despite retaining near-optimal scores on average. All fold fractions are descriptive.

Full Top-2/3 grouped values and ranking results for pair-type-only or combined models are not supplied in the reported summaries. No missing outcomes are inferred. The evidence supports repository-specific near-optimal ranking by some descriptors, while exact-best recovery is unreliable under stricter holdouts. It does not demonstrate an advantage of supervised compatibility over CKA.

Table 11. **Same-domain LOOCV; 15 pairs per subset**

Protocol	Target	Pearson r	Spearman rho	MAE	R^2
Chat-chat	Chat ROUGE-L	-0.373	-0.407	0.006	-0.476
Math-math	Math exact match	-0.574	-0.700	0.012	-0.498

Source: **compiled by the authors**

Table 12. **Adapter-grouped prediction; means and descriptive fold dispersion**

Protocol	Target	Folds	Valid folds	Mean r	SD r	Mean MAE	Mean R^2
Adapter-disjoint	Chat ROUGE-L	225	225	0.878	0.079	0.029	0.192
Adapter-disjoint	Math exact match	225	225	0.678	0.227	0.024	-0.866
Leave one adapter out	Chat ROUGE-L	12	12	0.372	0.650	0.026	-7.485
Leave one adapter out	Math exact match	12	12	0.493	0.383	0.021	-0.515

Source: **compiled by the authors**

Table 13. **Configuration- and recipe-disjoint prediction; descriptive fold summaries**

Protocol	Target	Folds	Valid folds	Mean r	SD r	Mean MAE	Mean R^2
Configuration-disjoint	Chat ROUGE-L	15	15	0.848	0.099	0.037	-0.328
Configuration-disjoint	Math exact match	15	15	0.673	0.259	0.026	-2.141
Recipe-disjoint	Chat ROUGE-L	6	6	0.831	0.056	0.036	-0.234
Recipe-disjoint	Math exact match	6	6	0.531	0.172	0.036	-5.713

Source: **compiled by the authors**

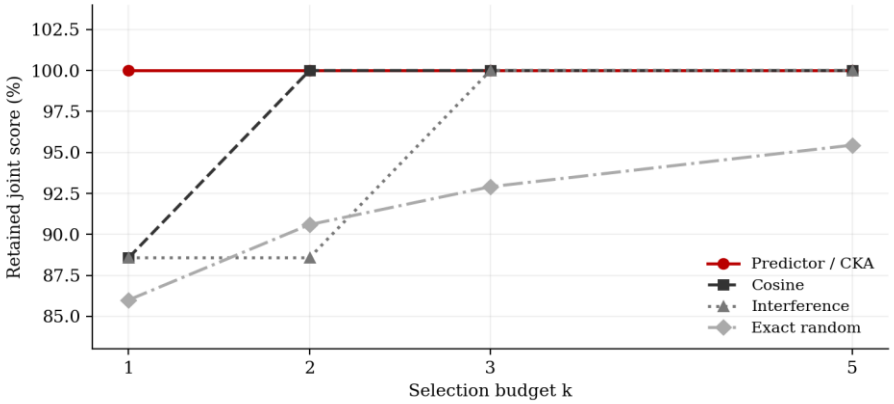


Fig. 3. **Held-out retained performance at budgets one, two, three and five. Predictor and CKA overlap at full retention**

***Source*: compiled by the authors from Table 14a-c**

Table 14a. **Retained performance (%) across selection budgets**

Method	k = 1	k = 2	k = 3	k = 5
Compatibility	100.00	100.00	100.00	100.00
Cosine	88.57	100.00	100.00	100.00
Interference	88.57	88.57	100.00	100.00
CKA	100.00	100.00	100.00	100.00
Exact random	86.00	90.59	92.91	95.45

Source: **compiled by the authors**

Table 14b. **Selection regret across selection budgets**

Method	k = 1	k = 2	k = 3	k = 5
Compatibility	0.0000	0.0000	0.0000	0.0000
Cosine	0.0941	0.0000	0.0000	0.0000
Interference	0.0941	0.0941	0.0000	0.0000
CKA	0.0000	0.0000	0.0000	0.0000
Exact random	0.1153	0.0775	0.0584	0.0375

Source: **compiled by the authors**

Table 14c. **Best-pair hit (%) across selection budgets**

Method	k = 1	k = 2	k = 3	k = 5
Compatibility	100.00	100.00	100.00	100.00
Cosine	0.00	100.00	100.00	100.00
Interference	0.00	0.00	100.00	100.00
CKA	100.00	100.00	100.00	100.00
Exact random	6.25	12.50	18.75	31.25

Source: **compiled by the authors**

Table 15a. **Mean retained performance (%) for grouped Top-1 selection**

Protocol	CKA	Compa-tibility	Exact random
Adapter-disjoint	96.40	93.37	89.11
Configuration-disjoint	94.82	94.32	90.83
Leave-one-adapter-out	96.93	93.73	85.77
Recipe-disjoint	93.72	89.28	85.87

Source: **compiled by the authors**

Table 15b. **Mean regret for grouped Top-1 selection**

Protocol	CKA	Compa-tibility	Exact random
Adapter-disjoint	0.0302	0.0546	0.0909
Configuration-disjoint	0.0412	0.0458	0.0754
Leave-one-adapter-out	0.0254	0.0518	0.1179
Recipe-disjoint	0.0532	0.0940	0.1229

Source: **compiled by the authors**

Table 15c. **Mean best-pair hit (%) for grouped Top-1 selection**

Protocol	CKA	Compa-tibility	Exact random
Adapter-disjoint	40.44	22.67	16.67
Configuration-disjoint	6.67	13.33	16.67
Leave-one-adapter-out	58.33	33.33	9.09
Recipe-disjoint	0.00	0.00	7.82

Source: **compiled by the authors**

Historical mixed-pool result and cost scenario. For continuity with the original benchmark, Table 16 retains the historical all-66 ranking result. Historical Top-5 retains 93.60 % and Top-20 recovers the joint optimum. This mixed-pool objective and ranking differ from the held-out protocol and must not be combined with its retention results or treated as new-pair generalization.

Table 16. **Historical all-66 ranking under its original objective**

Budget k	Retained score (%)
1	93.07
3	93.60
5	93.60
10	93.60
20	100.00
66	100.00

Source: **compiled by the authors**

Table 17 retains the analytical large-N scenario for a previously calibrated predictor, fixed k = 5, constant per-pair time and unchanged serial execution. At N = 12, assumed exhaustive cost is 733 min and screening cost is 165.75 min, a projected 77.39 % reduction before calibration. No successful benchmark evaluation-time measurement validates the assumed denominator, and no large-N experiment establishes quality retention equivalent future pools would exceed that acquisition cost only if fit cost and other overheads remain below the resulting margin and the calibrated predictor transfers. This is not a saving from repeating an identical cached experiment. Within the original 66-pair pool, fifty labeled pairs leave only sixteen unseen candidates, so the applicable incremental comparison must reuse those existing results within the same pool rather than treating them as new external calibration cost.

DISCUSSION OF RESULTS

What the compatibility signals explain. Rather than replacing existing merging algorithms, the framework complements them by estimating source-pair quality before computationally expensive merge-and-evaluation. The present controls show, however, that much of the aggregate prediction is explained more accurately by pair type. Negative same-domain correlations further restrict claims about compatibility among homogeneous adapters. The scientific contribution is this distinction between domain separation and fine-grained selection, rather than a universal compatibility predictor.

Linear Regression performs well relative to the other investigated compatibility regressors in a small, low-dimensional benchmark. This does not establish that compatibility is inherently linear or that nonlinear models universally overfit. The near-collinearity of cosine and interference, their combined predictive gain and perturbation sensitivity motivate analysis of residual information and coefficient stability. More informative features and broader independent data are more defensible next steps than increasing predictor complexity alone.

Prediction versus selection. Prediction quality is assessed along three complementary axes: (i) absolute score calibration, (ii) stability across evaluation protocols, and (iii) ranking utility for Top-k selection. Pair type outperforms compatibility-only in held-out error, while the combined model has task- and metric-dependent changes. CKA matches supervised ranking on the held-out optimum and has lower mean grouped regret. These findings provide no consistent justification for paying supervised calibration cost over direct CKA ranking in this repository. Comparing pair-type and combined ranking, including randomized tie-breaking, remains necessary to determine whether either improves source selection.

Retained performance and optimum-hit probability answer different questions. A method can retain most attainable score while never finding an exact optimum in a small set of difficult holdouts. Both must be reported when interpreting a source-selection claim. Correlations, mean R^2 and a single Top-5 retention number are insufficient substitutes for this decision-level comparison.

Using the same timing assumptions, generating fifty external labels plus their features costs approximately 638.80 min before fitting. The per-new-pool saving D is 567.25 min.

Table 17. **Conditional serial cost projection; external calibration excluded**

N.	Pairs	Exhaustive (h)	Screened (h)	Reduction (%)	Speedup
12	66	12.22	2.76	77.39	4.42
100	4950	916.25	138.70	84.86	6.61
500	124750	23091.35	3473.13	84.96	6.65
1000	499500	92457.95	13903.68	84.96	6.65

Source: **compiled by the authors**

Practical cost and limitations. Practical savings depend on acquisition cost, cached results, descriptor cost and predictor transfer. Direct-feature and domain-metadata baselines have different information requirements and should not automatically be assigned the full three-feature cost. Reusing activations or dense updates across pairs could also change feature cost at larger N. The constant per-pair model is therefore a declared scenario, not a measured scaling law.

Several limitations should be considered when interpreting the present results. First, the experiments were conducted on a relatively small repository consisting of twelve LoRA adapters derived from a single Gemma-2-2B-IT backbone for Vietnamese downstream tasks. Although this controlled setting enables systematic analysis, larger repositories spanning multiple model families, languages, and training objectives are needed to assess the generality of the framework.

Second, TIES-Merging was used to generate ground-truth merge quality. Although screening is conceptually separable from merging, predictor targets depend on the operator; effectiveness with DARE, DELLA, Fisher-weighted averaging or linear interpolation remains to be validated. Pairwise findings do not establish compatibility for merging three or more models simultaneously.

Third, CKA depends on input selection and representation extraction. Historical cache and dataset identities are incompletely verified, the reconstructed prompts overlap evaluation inputs, and numeric-string matching has restricted mathematical interpretation. Conditional bootstrap intervals omit refitting uncertainty, grouped folds overlap, and full grouped ranking controls remain incomplete. These limitations restrict both end-to-end reproduction and claims about unseen repositories. Successful benchmark-runtime measurements are still required before reporting measured GPU savings.

CONCLUSIONS

This study investigated pre-merge compatibility prediction as an approach to source-pair screening for language model merging. Using twelve Vietnamese LoRA adapters derived from Gemma-2-2B-IT, we evaluated whether task-vector cosine similarity, interference rate and CKA predict TIES merge quality and support candidate ranking.

Linear Regression achieves held-out Pearson correlations of 0.858 for chat and 0.801 for math, but pair type predicts scores substantially more accurately. Adding compatibility to pair type yields a small numerical math-error reduction and worse chat error, with lower Spearman correlation for both tasks. Same-domain prediction is unsuccessful. Predictor and CKA both recover the held-out optimum at Top-1; grouped results establish no consistent supervised selection advantage and reveal limited exact-optimum recovery under configuration and recipe holdouts.

Conditional cost projections illustrate when screening could reduce merge/evaluation work, but measured GPU savings, transfer to larger repositories and calibration amortization are not established. Future work should evaluate pair-type and combined ranking, independent calibration prompts, complete benchmark provenance, additional merging operators and larger disjoint repositories. The evidence supports an empirical assessment of screening capabilities and limits within the investigated adapter space.

USE OF ARTIFICIAL INTELLIGENCE

The authors used generative artificial intelligence tools, including Claude (Anthropic) and ChatGPT (OpenAI), to assist with language editing, grammar checking, and improving the clarity and readability of the manuscript. These tools were not used to generate the scientific content, research methodology, experimental design, data analysis, or conclusions. All scientific contributions, interpretations, and final decisions regarding the manuscript were made exclusively by the authors.

ACKNOWLEDGMENTS

This paper is the academic result of the Research KHSV2504 fully funded by University of Transport Ho Chi Minh City.

Conflict of interests: The authors declare no financial, personal, copyright, or other conflicts that could have influenced the study or its results

Financing: This research was funded by Ho Chi Minh City University of Transport under research project KHSV2504

Data and code availability: The source models and datasets used in this study are publicly available. The experimental code and model-merging benchmark results are available from the corresponding author upon reasonable request

REFERENCES

1. Tran, T. V., Nguyen, T. K. T. & Pham, T. N. L. "Context-aware educational chatbots via Retrieval-augmented generation". *Herald of Advanced Information Technology*. 2026; 9 (3): 289–306, https://www.scopus.com/pages/publications/105044438253. DOI: https://doi.org/10.15276/hait.09.2026.19.

2. Boiko, O. A. & Danylov, V. Ya. "Implicit vs Explicit reasoning in strategic narrative detection". *Herald of Advanced Information Technology*. 2026; 9 (2): 168–180. DOI: https://doi.org/10.15276/hait.09.2026.12.

3. Bocharova, M. Y. & Malakhov, E. V. "ResJobFit: End-to-End artificial neural networks based technology for job-resume matching". *Applied Aspects of Information Technology*. 2024; 7 (4): 378–391. DOI: https://doi.org/10.15276/aait.07.2024.27.

4. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. & Chen, W. "LoRA: Low-rank adaptation of large language models". *arXiv*. 2021. DOI: https://doi.org/10.48550/arXiv.2106.09685.

5. Yang, E., Shen, L., Guo, G., Wang, X., Cao, X., Zhang, J. & Tao, D. "Model merging in LLMs, MLLMs, and Beyond: Methods, theories, applications and opportunities". *arXiv*. 2024. DOI: https://doi.org/10.48550/arXiv.2408.07666.

6. Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H. & Farhadi, A. "Editing models with task arithmetic". *arXiv*. 2022. DOI: https://doi.org/10.48550/arXiv.2212.04089.

7. Yadav, P., Tam, D., Choshen, L., Raffel, C. & Bansal, M. "TIES-Merging: Resolving interference when merging models". *arXiv*. 2023. DOI: https://doi.org/10.48550/arXiv.2306.01708.

8. Yu, L., Yu, B., Yu, H., Huang, F. & Li, Y. "Language models are super mario: absorbing abilities from homologous models as a free lunch". *arXiv*. 2023. DOI: https://doi.org/10.48550/arXiv.2311.03099

9. Jang, D.-H., Yun, S. & Han, D. "Model Stock: All we need is just a few fine-tuned models". *arXiv*. 2024. DOI: https://doi.org/10.48550/arXiv.2403.19522.

10. Yang, E., Wang, Z., Shen, L., Liu, S., Guo, G., Wang, X. & Tao, D. "AdaMerging: Adaptive model merging for multi-task learning". *arXiv*. 2023. DOI: https://doi.org/10.48550/arXiv.2310.02575.

11. Akiba, T., Shing, M., Tang, Y., Sun, Q. & Ha, D. "Evolutionary optimization of model merging recipes". *Nature Machine Intelligence*. 2025; 7; 195–204, https://www.scopus.com/pages/publications/85217225093. DOI: 10.1038/s42256-024-00975-8.

12. Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., et al. "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time". *arXiv*. 2022. DOI: https://doi.org/10.48550/arXiv.2203.05482.

13. Matena, M. & Raffel, C. "Merging models with fisher-weighted averaging". *arXiv*. 2021. DOI: https://doi.org/10.48550/arXiv.2111.09832.

14. Ainsworth, S. K., Hayase, J. & Srinivasa, S. "Git Re-Basin: Merging Models modulo Permutation Symmetries". *arXiv*. 2022. DOI: https://doi.org/10.48550/arXiv.2209.04836.

15. Ortiz-Jimenez, G., Favero, A. & Frossard, P. "Task arithmetic in the tangent space: Improved editing of pre-trained models". *arXiv*. 2023. DOI: https://doi.org/10.48550/arXiv.2305.12827.

16. Stoica, G., Ramesh, P., Ecsedi, B., Choshen, L. & Hoffman, J. "Model merging with SVD to tie the Knots". *arXiv*. 2024. DOI: https://doi.org/10.48550/arXiv.2410.19735.

17. Deep, P. T., Bhardwaj, R. & Poria, S. "DELLA-Merging: reducing interference in model merging through magnitude-based sampling". *arXiv*. 2024. DOI: https://doi.org/10.48550/arXiv.2406.11617.

18. Tam, D., Bansal, M. & Raffel, C. "Merging by Matching Models in Task Parameter Subspaces". *arXiv*. 2023. DOI: https://doi.org/10.48550/arXiv.2312.04339.

19. Davari, M. & Belilovsky, E. "Model breadcrumbs: scaling multi-task model merging with sparse masks". *arXiv*. 2023, https://www.scopus.com/pages/publications/85209936683. DOI: https://doi.org/10.48550/arXiv.2312.06795.

20. Jin, X., Ren, X., Preotiuc-Pietro, D. & Cheng, P. "Dataless knowledge fusion by merging weights of language models". *arXiv*. 2022. DOI: https://doi.org/10.48550/arXiv.2212.09849.

21. Ramé, A., Couairon, G., Shukor, M., Dancette, C., Gaya, J.-B., Soulier, L. & Cord, M. "Rewarded soups: Towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards". *arXiv*. 2023. DOI: https://doi.org/10.48550/arXiv.2306.04488.

22. Zhou, Y., Song, L., Wang, B. & Chen, P. "MetaGPT: Merging Large language models using model exclusive task arithmetic". *Proceedings of the 2024 Conference on Empirical Methods in Natural Language*

Processing. 2024. p. 1711–1724, https://www.scopus.com/pages/publications/85217824420. DOI: https://doi.org/10.18653/v1/2024.emnlp-main.102.

23. He, Y., Hu, Y., Lin, Y., Zhang, T. & Zhao, H. "Localize-and-Stitch: Efficient model merging via sparse task arithmetic". *arXiv*. 2024. DOI: https://doi.org/10.48550/arXiv.2408.13656.

24. Kornblith, S., Norouzi, M., Lee, H. & Hinton, G. "Similarity of neural network representations revisited". arXiv 2019. DOI: https://doi.org/10.48550/arXiv.1905.00414.

25. Gemma Team. "Gemma 2: Improving open language models at a practical size". *arXiv*. 2024. DOI: https://doi.org/10.48550/arXiv.1201.0490.

26. Pedregosa, F., et al. "Scikit-learn: Machine learning in Python". *Journal of Machine Learning Research*. 2011; 12; 2825–2830, https://jmlr.org/papers/v12/pedregosa11a.html. DOI: https://doi.org/10.48550/arXiv.1201.0490.

27. Chen, T. & Guestrin, C. "XGBoost: A scalable tree boosting system". *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016. p. 785–794. DOI: https://doi.org/10.1145/2939672.2939785.

28. Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R. & Radev, D. "SummEval: Re-evaluating summarization evaluation". *Transactions of the Association for Computational Linguistics*. 2021; 9: 391–409. DOI: https://doi.org/10.1162/tacl_a_00373.

Received 28.08.2026
Received after revision 25.10.2026
Accepted 30.10.2026

DOI: https://doi.org/10.15276/aait.09.2026.38
UDC 004.8:004.89:378.147

Попередній відбір пар вихідних моделей за сумісністю для об'єднання мовних моделей: емпіричне оцінювання

Фам Чан Нят Лінь[1)]
ORCID: https://orcid.org/0009-0008-9245-3387; linhptn9746@ut.edu.vn
Нгуєн Тхі Кхань Тієн[1)]*
ORCID: https://orcid.org/0000-0001-5379-7226; tienntk@ut.edu.vn
Чан Тхе Вінь[1)]
ORCID: https://orcid.org/0000-0002-4241-1065; vinhtt@ut.edu.vn
[1)] Університет транспорту міста Хошимін, вул. Во Оань, 2. Хошимін, 700000, В'єтнам
*Автор для кореспонденції. E-mail: tienntk@ut.edu.vn

АНОТАЦІЯ

Актуальність. Зростання репозиторіїв спеціалізованих адаптерів мовних моделей збільшує витрати на повне оцінювання пар. **Мета.** Визначити, чи забезпечують ознаки сумісності додаткову цінність для прогнозування та відбору порівняно з типом пари й прямим ранжуванням за ознакою. **Методи.** Дванадцять низькорангових адаптерів зі спільною основою утворюють шістдесят шість діалогових і математичних пар. Напрям оновлень, знакова інтерференція та вирівнювання центрованих ядер використовуються для прогнозування якості об'єднання. Оцінювання охоплює перехресну перевірку на навчальній частині, шістнадцять відкладених пар, аналіз ознак, однорідні домени та виключення адаптерів, конфігурацій і рецептів. **Результати.** Лінійна регресія дає додатні кореляції на відкладених парах, але тип пари прогнозує якість точніше. Для діалогу середня абсолютна похибка зменшується з приблизно двадцяти п'яти тисячних до восьми тисячних, а для математики — з дев'ятнадцяти до десяти тисячних. Додавання ознак сумісності до типу пари дещо покращує математичну похибку, але погіршує діалогову. Прогнозувач і подібність представлень знаходять найкращу відкладену пару вже за одного вибору. Групові перевірки виявляють нестійке калібрування та відсутність послідовної переваги прогнозувача; кореляції в межах однорідних доменів від'ємні. **Висновки.** Значна частина загальної передбачуваності пояснюється доменним складом. Ранжування має обмежену цінність у дослідженому репозиторії, проте перевагу навчання прогнозувача над простішими правилами не встановлено. Економія часу залишається умовним прогнозом через відсутність успішних вимірювань тривалості повного оцінювання. Кількість пар зростає квадратично; практична вигода залежить від витрат на калібрування, повторного використання, кешування та перенесення на нові джерела.

Ключові слова: об'єднання мовних моделей; попередній відбір пар вихідних моделей; прогнозування сумісності; ранжування кандидатних пар; відбір адаптерів; мовні моделі

ABOUT THE AUTHORS



Tran Nhat Linh Pham - Student of Data Science major, Institute of Information Technology & Electrical and Electronics Engineering. Ho Chi Minh City University of Transport, 2, Vo Oanh Str. Ho Chi Minh City, 700000, Vietnam
ORCID: https://orcid.org/0009-0008-9245-3387; linhptn9746@ut.edu.vn
Research field: Data science, deep learning, computer vision, natural language processing

Фам Чан Нят Лінь - студент спеціальності "Наука про дані". Інститут інформаційних технологій та електротехніки та електронної інженерії. Університет транспорту міста Хошимін, вул. Во Оань, 2, Хошимін, 700000, В'єтнам
Галузь досліджень: Наука про дані, глибоке навчання, комп'ютерний зір, обробка природної мови



Thi Khanh Tien Nguyen - Doctor of Philosophy, Lecturer. Ho Chi Minh City University of Transport, 2, Vo Oanh Str. Ho Chi Minh City, 700000, Vietnam
ORCID: https://orcid.org/0000-0001-5379-7226; tienntk@ut.edu.vn. Scopus Author ID: 58735109800
Research field: Data science, deep learning, computer vision, natural language processing

Нгуєн Тхі Кхань Тієн - доктор філюософії, старший викладач. Університет транспорту міста Хошимін, вул. Во Оань, 2, Хошимін, 700000, В'єтнам
Галузь досліджень: Наука про дані, глибоке навчання, комп'ютерний зір, обробка природної мови



The Vinh Tran - Doctor of Philosophy, Program Director of Data Science. Ho Chi Minh City University of Transport, 2, Vo Oanh Str., Ho Chi Minh City, 700000, Vietnam
ORCID: https://orcid.org/0000-0002-4241-1065; vinhtt@ut.edu.vn. Scopus Author ID: 56835334700
Research field: Data Science, cybersecurity, blockchain technologies, deep learning and systems architecture

Чан Тхе Вінь - доктор філософії, директор програми з науки про дані. Університет транспорту міста Хошимін, вул. Во Оань, 2. Хошимін, 700000, В'єтнам
Галузь досліджень: Наука про дані, кібербезпека, технології блокчейн, глибоке навчання та системна архітектура