ERROR-SUBSPACE CALIBRATION FOR MULTI-TASK MODEL MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Model merging combines task-specific expert models into a single model without jointly retraining it on their original datasets, but often incurs a performance gap relative to the individual experts. Post-merging calibration can reduce this gap through task-specific representation alignment, but the alignment objective alone does not specify which feature directions to correct. We propose *Error-Subspace Calibration* (**ESCAL**), a post-merging calibration framework that identifies low-dimensional error subspaces from weight differences between expert and merged models. At each intermediate block boundary, a BRIDGE module projects the merged representation onto the next block's input error subspace and maps these coordinates to an additive correction in the current block's output error subspace. The subspace bases remain fixed, so calibration learns only these linear maps, yielding a parameter-efficient mechanism for feature correction. We find that ESCAL brings the merged model closer to the experts in both representations and predictions, indicating more faithful recovery of task-specific behavior. Experiments on vision and natural language understanding benchmarks show that ESCAL improves merged-model performance across diverse architectures and merging methods using a small calibration set for each task.

1 INTRODUCTION

Model merging aims to integrate the capabilities of multiple task-specific *expert* models into a single *merged* model without jointly training it on their original datasets (Ilharco et al., 2023; Yadav et al., 2023; Cheng et al., 2025; Xu et al., 2025). By reusing fine-tuned expert models, it offers a practical alternative to multi-task training or deploying a separate model for each task. However, the merged model often performs worse than the expert models, leaving a performance gap that limits the effectiveness of multi-task model merging. Reducing this gap while retaining the efficiency of model merging remains an important research question (Yang et al., 2026).

To narrow this gap, post-merging calibration methods use a small calibration set to align the representations of merged models with those of the corresponding experts (Yang et al., 2024a;b; Gu et al., 2026). The expert representations serve as alignment targets, enabling calibration without labels. Calibration can be applied at the final representation or at intermediate layers to address discrepancies that arise throughout the network.

Although representation alignment is effective, the alignment objective alone does not explicitly specify the feature directions that calibration modules can modify. For example, SURGERY and SURGERYV2 learn the projection matrices of their low-rank calibration modules from calibration samples, with the correction directions determined by the alignment objective. This motivates us to ask whether these directions can be identified from merging errors before calibration, leaving fewer parameters to learn from the calibration data. *Can we improve both the effectiveness and efficiency of representation alignment by restricting calibration to the feature subspaces associated with merging errors?*

To investigate this question, we examine *merging error*, defined as the difference between expert weights and the merged weights. Its dominant singular subspaces capture the largest components of these weight differences. Based on this insight, we propose *Error-Subspace Calibration* (**ESCAL**), which constructs task-specific calibration subspaces from merging errors. At each intermediate block boundary, a BRIDGE module reads coordinates from the next block's input error subspace and maps

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.

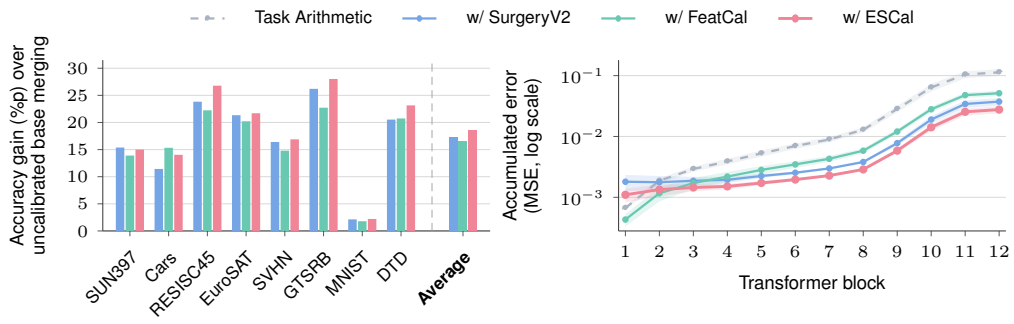

Figure 1: **Post-merging calibration of an eight-task TASK ARITHMETIC merge of ViT-B/32 experts. (Left)** Test-accuracy gains over the uncalibrated merge (percentage points); "Average" denotes the eight-task mean. **(Right)** Mean squared error between merged and expert hidden features at depth ℓ, averaged over eight tasks; shading indicates standard errors. ESCAL achieves the largest accuracy gains on six tasks and on average, and the lowest feature error at most block boundaries.

them to an additive correction in the current block's output error subspace. With both subspaces fixed, calibration learns only a mapping between them (Section 3).

We evaluate ESCAL on vision and natural language understanding benchmarks across various architectures and base merging methods, with additional experiments on natural language generation. ESCAL improves merged-model performance using a small calibration set per task (Section 4). Our ablation studies examine how subspace construction and constraints affect calibration performance (Section 5.1). Our alignment analyses further show that ESCAL brings the merged model closer to the expert models in both representations and predictions (Section 5.2).

Keeping the subspace bases fixed also makes ESCAL parameter-efficient, as each BRIDGE module learns only an $r \times r$ mapping for each task and block boundary. Across all evaluated configurations, ESCAL uses fewer trainable parameters than SURGERYV2, while achieving higher average performance in most settings (Section 5.3).

2 RELATED WORK

2.1 MULTI-TASK MODEL MERGING

A foundational approach is TASK ARITHMETIC (Ilharco et al., 2023), which represents the knowledge acquired during fine-tuning as a *task vector*, defined as the difference between the fine-tuned and pretrained model parameters. It constructs a multi-task model by adding the sum of task vectors, scaled by a single merging coefficient, to the pretrained model parameters. Building on this approach, ADAMERGING (Yang et al., 2024c) adaptively learns task-wise or layer-wise merging coefficients using unlabeled test data, rather than relying on a single fixed coefficient.

A major line of work focuses on mitigating interference among task updates. TIES-MERGING (Yadav et al., 2023) reduces interference by removing redundant parameter updates and resolving sign conflicts before merging. WUDI-MERGING (Cheng et al., 2025) leverages task-vector-induced subspaces to reduce parameter interference without additional data. PRODISTILL (Xu et al., 2025) instead uses few-shot data to progressively calibrate intermediate features through layer-wise distillation with element-wise merging coefficients, allowing more fine-grained adaptation of the merged model toward task-specific experts. These approaches modify how the merged model is constructed, in contrast to post-merging calibration methods that operate on an already merged checkpoint. Additional related work on model merging is discussed in Appendix A.

2.2 POST-MERGING CALIBRATION

Post-merging calibration aims to reduce the performance gap between the merged model and its task-specific experts by correcting representation discrepancies after merging.

SURGERY (Yang et al., 2024a) characterizes these discrepancies as *representation bias* and learns a lightweight task-specific low-rank module to align the merged model's final-layer representation with that of the corresponding expert. Its projection matrices are learned from unlabeled test

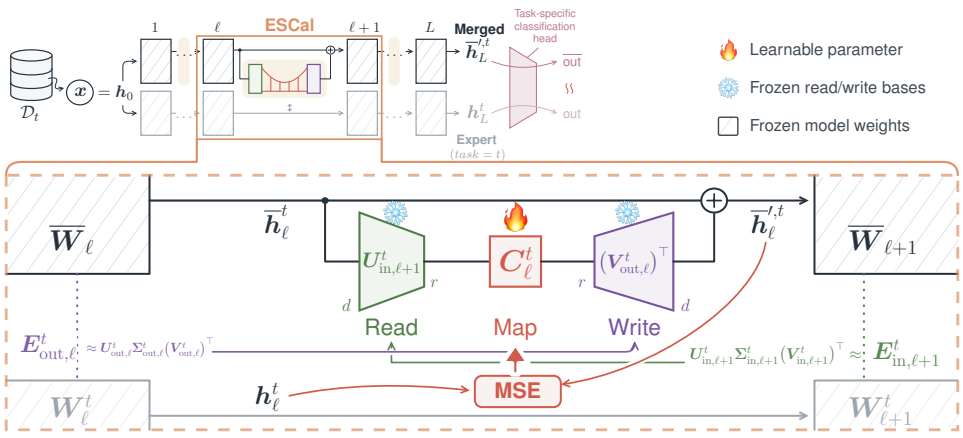

Figure 2: **Overview of Error-Subspace Calibration (ESCAL).** Task-specific BRIDGE modules align the frozen merged model with its expert at complete block outputs, after both residual additions. At intermediate boundaries, read and write bases come from merging errors in the next block's joint QKV input projection and the current block's MLP output projection, respectively.

samples, allowing the directions of the representation correction to be determined by the alignment objective. PROBSURGERY (Wei et al., 2025) extends this framework by probabilistically modeling representation bias and its uncertainty.

While these approaches calibrate only the final representation, SURGERYV2 (Yang et al., 2024b) extends calibration to intermediate layers, motivated by the observation that representation bias can arise throughout the network and accumulate with depth. It places task-specific calibration modules at multiple locations and aligns the merged representations with those of the corresponding expert. INTERVMERGE (Osial et al., 2024) uses a similar layer-wise correction strategy, formulating task-specific feature adjustment as low-rank interventions within Transformer blocks. Its parameter-efficient mini-interventions restrict the correction to selected dimensions of the intermediate representation.

More recently, FEATCAL (Gu et al., 2026) takes a different approach by interpreting the discrepancy between merged and expert representations as *feature drift*. It explicitly accounts for both locally introduced errors and their propagation through subsequent layers and, rather than introducing additional calibration modules, directly recalibrates the merged weights layer by layer using closed-form updates. In contrast, ESCAL keeps the merged weights frozen and learns only small mappings between fixed error subspaces derived from weight differences between the expert and merged models.

3 METHOD: ERROR-SUBSPACE CALIBRATION

We characterize the discrepancy introduced by model merging through low-dimensional error subspaces derived from weight differences between expert and merged models, and use them to constrain post-merging calibration. The subspaces determine where a calibration module reads from and where its correction can act, leaving only the mapping between them to be learned.

Notation. For a positive integer N, we write $[N] = \{1, \ldots, N\}$. We consider T tasks and a Transformer backbone with L blocks, using $t \in [T]$ to identify a task and $\ell \in [L]$ to identify a block. For a given linear transformation in block ℓ, $\boldsymbol{W}_\ell^t$ denotes the weight matrix of the expert for task t, while $\overline{\boldsymbol{W}}_\ell$ denotes the corresponding weight matrix of the merged model. The merged weights are shared across tasks so we omit the task index. Throughout, we represent hidden features as row vectors: an input $\boldsymbol{x}$ is multiplied by the weight matrix on the right, giving $\boldsymbol{x}\boldsymbol{W}$ for a linear transformation.

3.1 MERGING ERROR SUBSPACES

For a linear transformation, the discrepancy between expert and merged outputs under a common input is determined by their weight difference. We define the merging error as

$$\boldsymbol{E}_\ell^t = \boldsymbol{W}_\ell^t - \overline{\boldsymbol{W}}_\ell. \tag{1}$$

We use the leading r left and right singular vectors of E_ℓ^t as bases for the low-dimensional input and output error subspaces, respectively. For a fixed rank budget, truncated SVD minimizes the approximation error of the weight difference in Frobenius norm, supporting the choice of leading singular directions. This argument concerns the local output discrepancy of a linear transformation. Appendix B provides the linear-map analysis underlying this argument.

3.2 Cross-Layer Error-Subspace Calibration

Feature drift in a merged model reflects both locally introduced mismatch and drift propagated from preceding layers (Gu et al., 2026). Thus, a discrepancy at one block also affects the representations processed by subsequent blocks. This propagation view suggests calibrating at block boundaries, where a representation serves as both the output of the current block and the input to the next.

For a selected linear transformation under a common input, its weight-induced output discrepancy depends only on the input's coordinates in its full input error subspace. This motivates using the next block's input error subspace to select the coordinates the calibration module reads, while using the current block's output error subspace to specify its correction directions. This local linear argument does not assume that these subspaces capture all drift propagated through nonlinear blocks.

Read and write bases. Within block ℓ, we use the joint query, key, and value (QKV) input projection of self-attention and the MLP output projection to define the input and output error subspaces, respectively. We denote their expert weights by $\boldsymbol{W}_{\text{in},\ell}^t$ and $\boldsymbol{W}_{\text{out},\ell}^t$. We write both projection weights in input-by-output form. The input projection has d rows, and the output projection has d columns, so their left and right singular vectors, respectively, lie in the same d-dimensional hidden representation space.

Applying the merging error definition to these projections gives

$$\boldsymbol{E}_{\text{in},\ell}^t = \boldsymbol{W}_{\text{in},\ell}^t - \overline{\boldsymbol{W}}_{\text{in},\ell}, \qquad \boldsymbol{E}_{\text{out},\ell}^t = \boldsymbol{W}_{\text{out},\ell}^t - \overline{\boldsymbol{W}}_{\text{out},\ell}. \tag{2}$$

We use $\boldsymbol{U}_{\text{in},\ell}^t$ and $\boldsymbol{V}_{\text{out},\ell}^t$ to denote their top-r left and right singular-vector bases, respectively.

At the boundary between blocks ℓ and $\ell+1$, we pair the read basis $\boldsymbol{U}_{\text{in},\ell+1}^t$ with the write basis $\boldsymbol{V}_{\text{out},\ell}^t$. This pairing reads coordinates along directions most relevant to block $\ell+1$'s merging error and writes corrections along the dominant output-error directions of block ℓ. The two bases need not span the same subspace.

Bridge module. Using these fixed rank-r bases, we define the task-specific Bridge module at each intermediate block boundary $\ell = 1, \ldots, L-1$ as

$$\boldsymbol{B}_\ell^t = \boldsymbol{U}_{\text{in},\ell+1}^t \boldsymbol{C}_\ell^t \left(\boldsymbol{V}_{\text{out},\ell}^t\right)^\top, \qquad \boldsymbol{C}_\ell^t \in \mathbb{R}^{r\times r}, \tag{3}$$

where the core $\boldsymbol{C}_\ell^t$ maps coordinates in the read basis to correction coefficients in the write basis. Only $\boldsymbol{C}_\ell^t$ is trainable, requiring r^2 parameters per task and location.

At the terminal location, where there is no following Transformer block, we use the task-specific output head's top-r input singular vectors as the read basis $\boldsymbol{U}_{\text{head}}^t$, giving

$$\boldsymbol{B}_L^t = \boldsymbol{U}_{\text{head}}^t \boldsymbol{C}_L^t \left(\boldsymbol{V}_{\text{out},L}^t\right)^\top. \tag{4}$$

Unlike the intermediate read bases, $\boldsymbol{U}_{\text{head}}^t$ is derived from the head weights, capturing directions to which the head is most sensitive.

3.3 Module Training and Feature Calibration

Calibration is performed independently for each task t using a task-specific calibration set $\mathcal{D}_t$. The read and write bases are computed once and kept fixed, while both the merged model and the corresponding expert remain frozen. We optimize only the Bridge cores $\boldsymbol{C}_\ell^t$ to align the merged model's representations with those of the expert.

For an input $\boldsymbol{x}$, let $\overline{\boldsymbol{h}}_\ell^t(\boldsymbol{x})$ denote the output of merged Transformer block ℓ after both residual additions in its self-attention and MLP sublayers, including Bridge corrections propagated from

preceding block boundaries. At each intermediate boundary $\ell = 1, \ldots, L-1$, BRIDGE applies the residual correction

$$\begin{aligned} \overline{\boldsymbol{h}}_{\ell}^{\prime,t}(\boldsymbol{x}) &= \overline{\boldsymbol{h}}_{\ell}^{t}(\boldsymbol{x}) + \overline{\boldsymbol{h}}_{\ell}^{t}(\boldsymbol{x})\boldsymbol{B}_{\ell}^{t} \\ &= \overline{\boldsymbol{h}}_{\ell}^{t}(\boldsymbol{x}) + \left(\overline{\boldsymbol{h}}_{\ell}^{t}(\boldsymbol{x})\boldsymbol{U}_{\mathrm{in},\ell+1}^{t}\right)\boldsymbol{C}_{\ell}^{t}\left(\boldsymbol{V}_{\mathrm{out},\ell}^{t}\right)^{\top}. \end{aligned} \tag{5}$$

The corrected representation $\overline{\boldsymbol{h}}_{\ell}^{\prime,t}(\boldsymbol{x})$ is then passed to merged block $\ell+1$, so corrections are propagated sequentially through the backbone. At the terminal location, the analogous update using $\boldsymbol{B}_{L}^{t}$ produces the calibrated final representation $\overline{\boldsymbol{h}}_{L}^{\prime,t}(\boldsymbol{x})$.

Let $\boldsymbol{h}_{\ell}^{t}(\boldsymbol{x})$ denote the corresponding expert representation. We jointly optimize all BRIDGE cores for task t by minimizing the feature discrepancy across calibration locations:

$$\min_{\{\boldsymbol{C}_{\ell}^{t}\}_{\ell=1}^{L}} \sum_{\ell=1}^{L} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_{t}}\left[\left\|\overline{\boldsymbol{h}}_{\ell}^{\prime,t}(\boldsymbol{x}) - \boldsymbol{h}_{\ell}^{t}(\boldsymbol{x})\right\|_{F}^{2}\right]. \tag{6}$$

Since the frozen expert provides the target representations, calibration requires no task labels.

4 EXPERIMENTAL RESULTS

We compare ESCAL with existing post-merging calibration methods under different base merging methods. Our evaluation examines both the improvement over uncalibrated merged models and the remaining performance gap to the expert models. We conduct our main evaluation on vision and natural language understanding (NLU), and additionally evaluate natural language generation (NLG) to examine the applicability of ESCAL beyond these settings. The NLG results and discussion are provided in Appendix C.5.

Tasks and Models. For vision, we use CLIP ViT-B/32, ViT-B/16, and ViT-L/14 models (Radford et al., 2021) fine-tuned on eight image classification tasks. For NLU, we use RoBERTa-base (Liu et al., 2019) models fine-tuned on eight GLUE tasks (Wang et al., 2018). In both settings, we merge the eight task-specific models and evaluate the resulting model across all tasks. For NLG, we use three LLaMA-2-13B (Touvron et al., 2023) experts specialized for instruction following, mathematical reasoning, and code generation, and consider all pairwise merges.

Merging and Calibration Methods. We consider TASK ARITHMETIC (Ilharco et al., 2023), TIES-MERGING (Yadav et al., 2023), ADAMERGING (Yang et al., 2024c), WUDI-MERGING (Cheng et al., 2025), and PRODISTILL (Xu et al., 2025) as base merging methods. For post-merging calibration, we compare SURGERY (Yang et al., 2024a), SURGERYV2 (Yang et al., 2024b), FEATCAL (Gu et al., 2026), and ESCAL (ours). We report results for each base merged model before and after calibration. We also include the pretrained model and task-specific experts as references.

Calibration Protocol. For each base merging method, all calibration methods start from the same merged checkpoint and use the same task-specific calibration sets, each containing 64 input samples. SURGERY, SURGERYV2, and ESCAL optimize only their calibration modules with the merged weights frozen, whereas FEATCAL applies closed-form updates to the merged weights. Calibration and evaluation sets are disjoint, and results are averaged over five random seeds unless otherwise specified. Detailed hyperparameter settings are provided in Appendix C.2.

4.1 RESULTS ON VISION TASKS

Table 1 presents representative ViT-B/32 results with TASK ARITHMETIC, TIES-MERGING, and WUDI-MERGING. Complete results across architectures and base merging methods are provided in Appendix C.3.

ESCAL improves the average accuracy of all three base merged models shown in the table and achieves the highest average among the compared calibration methods. The largest improvement over the uncalibrated model occurs with TASK ARITHMETIC, whose average accuracy rises from 69.06% to 87.54%, a gain of 18.48 percentage points. ESCAL also improves WUDI-MERGING, which has the highest uncalibrated average accuracy among the three base merging methods, from 84.93% to 89.47%, a gain of 4.54 percentage points. These results show that calibration remains beneficial even when the base merged model already achieves high accuracy.

Table 1: **Vision model merging results on ViT-B/32.** Task-wise test accuracy (%) and average performance across eight image classification tasks.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Pretrained	63.18	59.73	60.33	45.37	31.61	32.56	48.25	43.99	48.13
Expert	75.34	77.73	95.98	99.74	97.46	98.73	99.69	79.36	90.50
TASK ARITHMETIC	55.32	54.98	66.82	77.78	80.21	69.68	97.34	50.37	69.06
w/ SURGERY	60.39	52.62	71.38	88.65	81.67	74.27	97.82	52.77	72.45
w/ SURGERYV2	**70.70**	66.41	90.65	99.12	96.61	95.88	99.47	70.90	86.22
w/ FEATCAL	69.22	**70.31**	89.06	98.00	95.01	92.40	99.13	71.11	85.53
w/ ESCAL	70.33	69.02	**93.60**	**99.48**	**97.10**	**97.70**	**99.55**	**73.52**	**87.54**
TIES-MERGING	65.05	64.43	74.92	76.48	81.28	69.37	96.53	54.52	72.82
w/ SURGERY	67.64	59.57	78.50	89.24	81.30	76.07	97.14	60.03	76.19
w/ SURGERYV2	70.87	66.26	90.14	98.94	96.54	95.52	99.45	70.89	86.08
w/ FEATCAL	70.12	**70.74**	89.28	97.93	94.71	92.55	99.10	71.13	85.69
w/ ESCAL	**70.93**	69.23	**93.31**	**99.36**	**97.09**	**97.70**	**99.53**	**73.35**	**87.56**
WUDI-MERGING	69.11	71.38	85.62	95.63	94.27	94.51	99.47	69.47	84.93
w/ SURGERY	69.24	69.42	87.70	97.53	94.81	95.43	99.48	71.37	85.62
w/ SURGERYV2	73.24	73.57	94.15	**99.59**	97.17	98.38	**99.64**	**77.66**	89.18
w/ FEATCAL	71.98	**75.80**	93.71	99.31	96.53	98.42	99.50	77.52	89.10
w/ ESCAL	**73.55**	74.68	**94.80**	99.58	**97.31**	**98.68**	**99.64**	77.52	**89.47**

These gains extend to both task-wise and layer-wise ADAMERGING (Appendix C.3). On ViT-B/32, ESCAL increases their average accuracies from 70.82% to 86.87% and from 79.31% to 87.91%, respectively, showing that calibration can provide further gains after merging coefficients have been adapted. The improvements also hold for PRODISTILL across all three architectures. Thus, ESCAL can complement merging methods that already use intermediate-feature distillation.

4.2 RESULTS ON NLU TASKS

Table 2 reports representative results with TASK ARITHMETIC, TIES-MERGING, and WUDI-MERGING. Complete results across base merging methods are provided in Appendix C.4.

Table 2: **NLU model merging results on RoBERTa-base.** Task-wise performance and average score across eight GLUE tasks.

Method	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	Avg.
Pretrained	0.0000	0.5975	0.1581	-0.0758	0.4633	0.3183	0.5078	0.5271	0.3120
Expert	0.5895	0.9438	0.9091	0.8997	0.8994	0.8718	0.9198	0.7726	0.8507
TASK ARITHMETIC	0.1232	0.6135	0.5500	0.5356	0.5881	0.6270	0.5365	0.5343	0.5135
w/ SURGERY	0.2553	0.8532	0.7657	0.7877	0.7793	0.7215	0.7839	0.7184	0.7081
w/ SURGERYV2	0.5358	**0.9326**	0.8908	0.8978	0.8002	0.8404	0.9084	0.7408	0.8184
w/ FEATCAL	0.3879	0.9140	0.8162	0.4900	0.7680	0.7569	0.8499	0.7011	0.7105
w/ ESCAL	**0.5588**	0.9319	**0.8937**	**0.8998**	**0.8061**	**0.8469**	**0.9117**	**0.7415**	**0.8238**
TIES-MERGING	0.1972	0.7626	0.5075	0.4390	0.6039	0.6670	0.5380	0.4946	0.5262
w/ SURGERY	0.2680	0.8635	0.7570	0.7770	0.7770	0.7347	0.7988	0.6989	0.7094
w/ SURGERYV2	0.5420	0.9319	**0.8848**	**0.8965**	**0.8205**	**0.8474**	**0.9079**	0.7394	**0.8213**
w/ FEATCAL	0.4055	0.9106	0.8111	0.4765	0.7702	0.7569	0.8433	0.7061	0.7100
w/ ESCAL	**0.5552**	**0.9339**	0.8769	0.8935	0.8144	0.8443	0.9070	**0.7430**	0.8210
WUDI-MERGING	0.0625	0.8073	0.2565	0.0927	0.7680	0.4431	0.4622	0.4874	0.4225
w/ SURGERY	0.2226	0.9025	0.7860	0.7758	0.8270	0.7755	0.8118	0.7206	0.7277
w/ SURGERYV2	0.5529	0.9392	0.8908	0.8868	**0.8798**	**0.8620**	0.9136	**0.7603**	0.8357
w/ FEATCAL	0.4387	0.9190	0.8215	0.4953	0.8234	0.7748	0.8358	0.6888	0.7247
w/ ESCAL	**0.5663**	**0.9397**	**0.8950**	**0.8953**	0.8744	0.8616	**0.9144**	0.7545	**0.8376**

ESCAL improves the average GLUE score for all three base merging methods. WUDI-MERGING benefits most, with its average score rising from 0.4225 to 0.8376. Despite having the lowest uncalibrated score among the three methods, WUDI-MERGING achieves the highest average after ESCAL calibration. Across these settings, ESCAL consistently outperforms SURGERY and FEATCAL. Relative to SURGERYV2, it achieves higher averages with TASK ARITHMETIC and WUDI-MERGING, and a comparable average with TIES-MERGING.

The improvements also extend to the additional base merging methods in Appendix C.4. For task-wise and layer-wise AdaMerging, ESCal raises the average scores from 0.4989 to 0.8202 and from 0.4823 to 0.8276, respectively; for ProDistill, it improves the average from 0.7298 to 0.7999. These results support the effectiveness of error-subspace calibration across merged checkpoints produced by different merging strategies.

5 Further Analyses

We further analyze ESCal to examine how its error-subspace design contributes to calibration performance and how closely the calibrated model matches expert predictions. We also assess its calibration data and parameter requirements, along with the additional storage and inference overhead of its calibration modules.

5.1 Effectiveness of Error-Subspace Construction

Table 3: We compare different basis sources on ViT-B/32 merged with Task Arithmetic using rank $r = 128$. We report mean accuracy here, with standard deviations provided in Table 16 in Appendix D.1.

Basis	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Random orthonormal	65.42	63.88	88.54	98.61	96.06	95.81	99.32	64.98	84.08
Fine-tuned weight, top-r	66.94	65.02	89.85	99.03	96.17	96.32	99.36	65.70	84.80
Task vector, top-r	69.29	67.72	93.01	99.44	97.08	97.47	99.53	72.53	87.01
Merging error, top-r	**70.33**	**69.02**	**93.60**	**99.48**	**97.10**	**97.70**	**99.55**	**73.52**	**87.54**

Source of calibration subspaces. We first compare top-r singular subspaces derived from fine-tuned weights, task vectors, and merging errors, together with a random orthonormal baseline. We construct the random baseline by sampling Gaussian random matrices and applying reduced QR decomposition.

As shown in Table 3, error-derived bases achieve the highest average accuracy of 87.54%, followed by task-vector bases with 87.01% and fine-tuned-weight bases with 84.80%. Fine-tuned weights represent the full expert transformation and do not explicitly isolate the parameter changes introduced by fine-tuning. Task vectors capture these changes relative to the pretrained model, whereas merging errors isolate the discrepancy between the expert and the merged model being calibrated. The observed ordering, therefore, supports constructing calibration subspaces from the discrepancy that the calibration is intended to correct.

Table 4: We independently vary the read and write directions for ESCal with Task Arithmetic on ViT-B/32 using rank $r = 128$. Random-r denotes r directions selected from the singular-vector basis of the merging error.

Read basis	Write basis		
	Top-r	Random-r	Bottom-r
Top-r	$\mathbf{87.54 \pm 0.07}$	84.24 ± 0.18	81.69 ± 0.12
Random-r	85.96 ± 0.15	82.93 ± 0.22	80.89 ± 0.19
Bottom-r	85.35 ± 0.07	82.63 ± 0.20	80.86 ± 0.13

Selection of read and write directions. We next examine which directions within the merging-error subspaces are most effective for reading and writing corrections. Table 4 independently varies the read and write bases among the top-r, random-r, and bottom-r singular directions of the merging error. Unlike the random orthonormal baseline considered above, Random-r selects r directions from the singular vectors of the merging error. Using the top-r singular directions for both the read and write bases achieves the highest average accuracy of 87.54%.

We find that the choice of write basis has a larger effect on performance. For a fixed read basis, replacing the top-r write directions with random-r or bottom-r directions consistently causes substantial degradation. This suggests that corrections should be written along the dominant output-error directions rather than random or bottom singular directions.

We also find that, for a fixed write basis, the top-r read directions consistently perform best, while the bottom-r directions perform worst. This indicates that, beyond correcting directions in which the current representation deviates from the expert, reading from input directions that most strongly affect the following transformation's local output discrepancy is beneficial. Together, these results support the ESCAL design of writing along the dominant current output-error directions while reading from the dominant input-error directions of the next transformation.

Table 5: Constrained uses a fixed top-r error basis. Unconstrained learns the corresponding projection from a random initialization without restricting its directions to the error subspace.

Read	**Write**	$r = 64$	$r = 128$	$r = 256$
Constrained	Constrained	86.91 ± 0.06	87.54 ± 0.07	87.62 ± 0.12
Unconstrained	Constrained	87.23 ± 0.05	87.50 ± 0.04	87.62 ± 0.14
Constrained	Unconstrained	87.53 ± 0.10	87.76 ± 0.04	87.75 ± 0.07

Role of the read- and write-subspace constraints. The preceding ablations examine which basis sources and singular directions are most effective when the read and write bases are fixed. We next examine whether relaxing either subspace constraint provides additional gains by replacing the corresponding fixed basis with a learnable projection. Specifically, we make either the read or write projection learnable from a random initialization while keeping the other constrained to its fixed top-r error basis.

As shown in Table 5, relaxing the write constraint improves accuracy across all three ranks, whereas relaxing the read constraint provides a gain only at $r = 64$. The more consistent gains from adapting the write projection further highlight the importance of the directions along which corrections are applied. However, relaxing either constraint requires learning a $d \times r$ projection in place of the corresponding fixed basis, increasing the number of trainable parameters for that projection by a factor of d/r compared with the $r \times r$ core. Given the modest performance gains relative to this additional parameter cost, we keep both bases fixed in ESCAL.

5.2 Prediction Alignment with Expert Models

To complement task-performance evaluation, we measure how closely the calibrated merged models match expert predictions using KL divergence and prediction agreement. Following the standard evaluation protocol for multi-task model merging, we pair each calibrated merged backbone with the corresponding fixed expert head and compare its predictions with those of the expert. For each task, we compute KL divergence from the expert predictive distribution to that of the calibrated merged model and prediction agreement, defined as the fraction of matching predicted labels, and report the arithmetic mean across tasks. For RoBERTa, we compute these metrics over the seven classification tasks, excluding STS-B, which is a regression task.

Table 6: **Prediction alignment with expert models.** KL divergence and prediction agreement are reported across vision and NLU architectures.

Method	ViT-B/32		ViT-B/16		ViT-L/14		RoBERTa	
	KL ↓	Agreement ↑	KL ↓	Agreement ↑	KL ↓	Agreement ↑	KL ↓	Agreement ↑
TASK ARITHMETIC	0.9383	74.51%	0.7918	79.56%	0.4447	86.83%	0.6926	61.38%
w/ SURGERYV2	0.2671	90.70%	0.2054	92.94%	0.1259	95.27%	0.2872	87.76%
w/ FEATCAL	0.3204	89.48%	0.2987	90.96%	0.1610	94.40%	0.4697	83.35%
w/ **ESCAL**	**0.1821**	**92.09%**	**0.1341**	**94.16%**	**0.0910**	**96.09%**	**0.2439**	**88.74%**

With TASK ARITHMETIC as the base merging method, ESCAL achieves the lowest KL divergence and the highest prediction agreement among the compared methods across all four evaluated architectures. On ViT-B/32, for example, relative to SURGERYV2, ESCAL reduces KL divergence from 0.2671 to 0.1821 and increases prediction agreement from 90.70% to 92.09%. These results show that ESCAL more closely matches expert predictions, with stronger alignment in both predictive distributions and predicted labels.

5.3 CALIBRATION EFFICIENCY AND OVERHEAD

We examine the parameter efficiency and calibration data requirements of ESCAL on ViT-B/32 with TASK ARITHMETIC. We also assess the additional storage and inference overhead associated with its calibration modules.

Table 7: Calibration cost and accuracy.

Method	Trainable params.	Runtime (min)	Avg. acc. (%)
SURGERY	131,072	8.48	72.45
SURGERYV2	2,490,368	13.76	86.22
FEATCAL	0	0.63	85.53
ESCAL	1,572,864	13.22	**87.54**

Table 8: Calibration budget.

Examples per task	Avg. acc. (%)
32	86.48 ± 0.21
64	87.54 ± 0.07
128	88.07 ± 0.06
256	88.29 ± 0.07

Parameter efficiency and inference overhead. ESCAL achieves higher average accuracy with 36.8% fewer trainable parameters than SURGERYV2 on ViT-B/32 (Table 7), while requiring comparable calibration runtime under the selected configurations. For FEATCAL, we report zero trainable parameters because its calibration updates are computed in closed form without gradient-based optimization. While only the core mappings are learned, the fixed read/write bases must also be retained at inference time, requiring additional storage beyond the trainable parameters. For TASK ARITHMETIC on ViT-B/32 with $r = 128$, the bases and cores together add 2.91% per task to the shared image encoder's parameter storage at the same precision. We additionally measure the inference overhead relative to the uncalibrated model in terms of latency and throughput, and report these results together with detailed storage accounting in Appendix D.2.

Calibration data budget. We vary the number of calibration examples per task from 4 to 256 for ESCAL with TASK ARITHMETIC on ViT-B/32 and evaluate on the full test sets. Table 8 reports the results from 32 to 256 examples, with the full sweep provided in Appendix D.3. Our default budget of 64 examples per task provides a favorable trade-off between accuracy and calibration data requirements.

6 CONCLUSION AND LIMITATIONS

We introduced ESCAL, a post-merging calibration framework that restricts feature correction to low-dimensional error subspaces derived from weight differences between expert and merged models. Its BRIDGE modules learn only small linear mappings between fixed subspaces, enabling parameter-efficient calibration. Across vision and natural language understanding, our results show that these error subspaces provide an effective structure for recovering task-specific capabilities after merging. This highlights the importance of identifying *which directions* to correct, in addition to optimizing representation alignment.

Limitations & Future Directions. First, each BRIDGE learns a low-rank linear correction to the blockwise hidden-representation gap between the merged model and an expert, with both subspace bases fixed. This design draws on prior work on lightweight post-merging calibration (Yang et al., 2024a;b). Our experiments demonstrate its effectiveness in the evaluated settings, but its general applicability remains unverified. Thus, ESCAL may be less effective when the required corrections fall outside the fixed error subspaces or require nonlinear transformations.

Second, the error-subspace dimension r must be chosen before BRIDGE training and may need to vary across architectures and domains. We treat r as a hyperparameter, but do not systematically investigate its relationship with the correction capacity of BRIDGE modules. We leave this analysis and adaptive rank allocation to future work.

Third, ESCAL retains its fixed read/write bases at inference time, requiring additional storage beyond the learned BRIDGE core modules. However, this overhead is relatively small. For example, with TASK ARITHMETIC on ViT-B/32 and $r = 128$, the bases and cores together add 2.91% per task to the shared image encoder's parameter storage at the same precision (Table 17).

Finally, we study BRIDGE modules only at block boundaries. Exploring alternative placements, including within blocks, could help clarify how correction placement affects representation alignment. More expressive correction modules also remain an architectural direction for future work.

AI USE STATEMENT

We used generative AI tools to assist with drafting and revising the manuscript, translating technical descriptions into English, organizing discussions of related work, and editing figures and their LaTeX source. We also used generative AI tools to assist with implementing and debugging the experimental code. We did not use generative AI tools for hypothesis formulation, the design of the proposed method, or experimental design, including ablation studies. We have reviewed and revised the AI-assisted material, checking technical statements, citations, and numerical values against the original sources and experimental results. We take full responsibility for the content of the final manuscript.

REFERENCES

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=xR9msNaREW.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

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.

Yanggan Gu, Shuo Cai, Zihao Wang, Wenjun Wang, Yuanyi Wang, Pengkai Wang, Sirui Huang, Su Lu, Jianmin Wu, and Hongxia Yang. Featcal: Feature calibration for post-merging models. *arXiv preprint arXiv:2605.13030*, 2026.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

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.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, et al. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *International Conference on Learning Representations*, pp. 49573–49609, 2025.

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.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, number 2, pp. 4. Granada, 2011.

Marcin Osial, Daniel Marczak, et al. Parameter-efficient interventions for enhanced model merging. *arXiv preprint arXiv:2412.17023*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Ronald Skorobogat, Karsten Roth, and Mariana-Iuliana Georgescu. Subspace-boosted model merging. *arXiv preprint arXiv:2506.16506*, 2025.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pp. 1453–1460. IEEE, 2011.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Qi Wei, Shuo He, Enneng Yang, Tingcong Liu, Haobo Wang, Lei Feng, and Bo An. Representation surgery in model merging with probabilistic modeling. In *Forty-second International Conference on Machine Learning*, 2025.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Learning Representations*, pp. 30745–30766, 2024.

Jing Xu, Jiazheng Li, and Jingzhao Zhang. Scalable model merging with progressive layer-wise distillation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=xX8NJShgny.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. 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. 56332–56356. PMLR, 21–27 Jul 2024a. URL https://proceedings.mlr.press/v235/yang24t.html.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xingwei Wang, Xiaocun Cao, Jie Zhang, and Dacheng Tao. Surgeryv2: Bridging the gap between model merging and multi-task learning with deep representation surgery. *arXiv preprint arXiv:2410.14389*, 2024b.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, pp. 22743–22763, 2024c.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications, and opportunities. *ACM Computing Surveys*, 58(8), February 2026. ISSN 0360-0300. doi: 10.1145/3787849. URL https://doi.org/10.1145/3787849.

CONTENTS

1 Introduction **1**

2 Related Work **2**
2.1 Multi-task Model Merging 2
2.2 Post-merging Calibration 2

3 Method: Error-Subspace Calibration **3**
3.1 Merging Error Subspaces 3
3.2 Cross-Layer Error-Subspace Calibration 4
3.3 Module Training and Feature Calibration 4

4 Experimental Results **5**
4.1 Results on Vision Tasks 5
4.2 Results on NLU Tasks 6

5 Further Analyses **7**
5.1 Effectiveness of Error-Subspace Construction 7
5.2 Prediction Alignment with Expert Models 8
5.3 Calibration Efficiency and Overhead 9

6 Conclusion and Limitations **9**

A Additional Related Work on Model Merging **14**

B Motivation for Error-Subspace Calibration **14**

C Experimental Details and Additional Results **16**
C.1 Experimental Settings 16
C.2 Hyperparameter Search 16
C.3 Vision Results 16
C.4 NLU Results 18
C.5 NLG Results 19

D Additional Analyses **20**
D.1 Error-Subspace Construction 20
D.2 Computational Cost 21
D.3 Calibration Data Budget 22

A Additional Related Work on Model Merging

Several model-merging approaches study the spectral structure of task updates to reduce interference and preserve task-specific information. TSV-Merge (Gargiulo et al., 2025) decomposes layer-wise task matrices into singular components, retains their dominant directions, and orthogonalizes these singular vectors across tasks before reconstructing the merged update. This combines low-rank compression with interference reduction. Iso-Merging (Marczak et al., 2025) studies the alignment between task-specific and merged updates and proposes flattening the singular value spectrum to balance the contributions of different directions. Its framework also incorporates common and task-specific subspaces to preserve information that may be underrepresented in the aggregate update. Subspace Boosting (Skorobogat et al., 2025) examines rank collapse as more experts are merged and boosts smaller singular values of the merged task update to retain a broader set of task-relevant directions. Alignment between task updates also matters when merging models fine-tuned with low-rank adaptation. KnOTS (Stoica et al., 2025) observes that LoRA task updates exhibit less alignment than their fully fine-tuned counterparts, which can limit the effectiveness of existing merging methods. It uses SVD to jointly transform the task updates into a common aligned space, applies an existing merging method in that space, and reconstructs the merged update.

B Motivation for Error-Subspace Calibration

We examine the discrepancy between a task-specific model and the merged model across consecutive transformations. We consider two properties of the merging error: the subspace containing the locally induced discrepancy at the current transformation, and the input directions relevant to the merging error of the following transformation. Here, the transformations are the selected linear maps used for the error decompositions, not entire nonlinear Transformer blocks. We use a row-vector convention in which a linear transformation maps $\boldsymbol{x}$ to $\boldsymbol{x}\boldsymbol{W}$, with all weight matrices written in input-by-output form.

We assume that the expert and merged model share the same architecture, with potentially different weights. We fix a task index $t \in [T]$ and omit it when clear from context.

Local merging error lies in the output error subspace. For a block $\ell \in [L]$, choose the same linear layer and take its weight matrix: $\boldsymbol{W}_\ell^t$ for the expert and $\overline{\boldsymbol{W}}_\ell$ for the merged model. Recall that the corresponding merging error is $\boldsymbol{E}_\ell^t = \boldsymbol{W}_\ell^t - \overline{\boldsymbol{W}}_\ell$. Write its reduced singular value decomposition (SVD) as $\boldsymbol{E}_\ell^t = \boldsymbol{U}_\ell^t \boldsymbol{\Sigma}_\ell^t \boldsymbol{V}_\ell^{t\top}$. Under our row-vector convention, the left singular matrix $\boldsymbol{U}_\ell^t$ spans the input-side singular directions of the error operator, while the right singular matrix $\boldsymbol{V}_\ell^t$ spans the corresponding output-side directions.

For a shared input representation $\boldsymbol{x}$, the discrepancy between the task-specific and merged transformations is

$$\begin{aligned} \boldsymbol{x}\boldsymbol{W}_\ell^t - \boldsymbol{x}\overline{\boldsymbol{W}}_\ell &= \boldsymbol{x}\boldsymbol{E}_\ell^t \\ &= \left(\boldsymbol{x}\boldsymbol{U}_\ell^t\right) \boldsymbol{\Sigma}_\ell^t \boldsymbol{V}_\ell^{t\top}. \end{aligned} \tag{7}$$

Therefore,

$$\left(\boldsymbol{x}\boldsymbol{E}_\ell^t\right)^\top \in \operatorname{span}(\boldsymbol{V}_\ell^t) \tag{8}$$

for any shared input $\boldsymbol{x}$. Thus, the discrepancy directly introduced by the weight difference at transformation ℓ is confined to the output singular subspace of the merging error.

This observation also implies that a correction outside this subspace cannot reduce the locally induced discrepancy. Let a correction be decomposed as

$$\Delta\boldsymbol{h}_\ell = \Delta\boldsymbol{h}_{\ell,\parallel} + \Delta\boldsymbol{h}_{\ell,\perp}, \tag{9}$$

where

$$\Delta\boldsymbol{h}_{\ell,\parallel}^\top \in \operatorname{span}(\boldsymbol{V}_\ell^t), \qquad \Delta\boldsymbol{h}_{\ell,\perp}^\top \perp \operatorname{span}(\boldsymbol{V}_\ell^t). \tag{10}$$

Since $(\boldsymbol{x}\boldsymbol{E}_\ell^t)^\top$ lies in $\operatorname{span}(\boldsymbol{V}_\ell^t)$,

$$\left\|\boldsymbol{x}\boldsymbol{E}_\ell^t - \Delta\boldsymbol{h}_\ell\right\|_2^2 = \left\|\boldsymbol{x}\boldsymbol{E}_\ell^t - \Delta\boldsymbol{h}_{\ell,\parallel}\right\|_2^2 + \left\|\Delta\boldsymbol{h}_{\ell,\perp}\right\|_2^2. \tag{11}$$

Hence, the orthogonal component $\Delta\boldsymbol{h}_{\ell,\perp}$ cannot improve the correction of the discrepancy directly induced by $\boldsymbol{E}_\ell^t$. This motivates restricting the correction to $\operatorname{span}(\boldsymbol{V}_\ell^t)$.

This argument concerns the local discrepancy induced by the weight difference under a shared input. The total representation discrepancy may also contain errors propagated from preceding layers.

The following transformation identifies error-relevant input directions. We next consider the merging error of the transformation following the calibration location. For an input representation $\boldsymbol{h}$,

$$\boldsymbol{h}\boldsymbol{E}^t_{\ell+1} = \left(\boldsymbol{h}\boldsymbol{U}^t_{\ell+1}\right)\boldsymbol{\Sigma}^t_{\ell+1}\boldsymbol{V}^{t\top}_{\ell+1}. \tag{12}$$

The discrepancy induced by the following transformation therefore depends on $\boldsymbol{h}$ through its coordinates along $\boldsymbol{U}^t_{\ell+1}$.

More specifically, let $\boldsymbol{u}^t_{\ell+1,i}$ and $\boldsymbol{v}^t_{\ell+1,i}$ denote the i-th left and right singular vectors of $\boldsymbol{E}^t_{\ell+1}$. Then

$$\boldsymbol{u}^{t\top}_{\ell+1,i}\boldsymbol{E}^t_{\ell+1} = \sigma^t_{\ell+1,i}\boldsymbol{v}^{t\top}_{\ell+1,i}, \tag{13}$$

and, since $\boldsymbol{v}^t_{\ell+1,i}$ has unit norm,

$$\left\|\boldsymbol{u}^{t\top}_{\ell+1,i}\boldsymbol{E}^t_{\ell+1}\right\|_2 = \sigma^t_{\ell+1,i}. \tag{14}$$

Thus, $\sigma^t_{\ell+1,i}$ measures the magnitude of the discrepancy induced by the error operator for a unit input along $\boldsymbol{u}^t_{\ell+1,i}$. Larger singular values therefore identify input directions along which the following transformation's weight difference can induce larger local discrepancies. This motivates using $\boldsymbol{U}^t_{\ell+1}$ as the read subspace of BRIDGE.

Scope of the argument. The quantity $\boldsymbol{h}\boldsymbol{E}^t_{\ell+1}$ is a local discrepancy caused by the following transformation's weight difference. It is distinct from the propagation of an existing representation perturbation $\Delta\boldsymbol{h}$ through the merged linear transformation, which is $\Delta\boldsymbol{h}\overline{\boldsymbol{W}}_{\ell+1}$. For a nonlinear block, such propagation depends on the block's local Jacobian. Thus, the argument motivates the read and write subspaces of BRIDGE, but does not imply optimality with respect to propagated error or the total discrepancy of the nonlinear network.

Why retain the leading rank-r subspaces? The arguments above concern the full singular subspaces. In practice, we retain only the leading r directions. For any transformation ℓ, the rank-r truncated SVD satisfies

$$\left\|\boldsymbol{E}^t_\ell - \boldsymbol{U}^t_{\ell,r}\boldsymbol{\Sigma}^t_{\ell,r}\boldsymbol{V}^{t\top}_{\ell,r}\right\|^2_F = \sum_{i>r}\left(\sigma^t_{\ell,i}\right)^2. \tag{15}$$

Truncated SVD minimizes this residual among rank-r approximations of the merging-error operator. For a fixed rank budget, it retains the dominant merging-error directions and minimizes the discarded error energy in the Frobenius norm. This optimality concerns approximation of the error matrix, not minimization of a data-weighted feature loss or end-to-end prediction error.

These observations motivate using $\boldsymbol{V}^t_{\ell,r}$ as the output space for the correction at layer ℓ and $\boldsymbol{U}^t_{\ell+1,r}$ as the input space from which the correction is determined. Accordingly, our BRIDGE module is parameterized as

$$\boldsymbol{B}^t_\ell = \boldsymbol{U}^t_{\ell+1,r}\boldsymbol{C}^t_\ell\boldsymbol{V}^{t\top}_{\ell,r}, \qquad \boldsymbol{C}^t_\ell \in \mathbb{R}^{r\times r}. \tag{16}$$

The error subspaces are fixed by the task-specific and merged weights, while calibration learns only the mapping $\boldsymbol{C}^t_\ell$ between them.

The resulting feature correction can then be written as

$$\Delta\boldsymbol{h}^t_\ell = \boldsymbol{h}_\ell\boldsymbol{B}^t_\ell = \left(\boldsymbol{h}_\ell\boldsymbol{U}^t_{\ell+1,r}\right)\boldsymbol{C}^t_\ell\boldsymbol{V}^{t\top}_{\ell,r}. \tag{17}$$

Hence, BRIDGE reads coordinates from the following transformation's input error subspace, maps them through $\boldsymbol{C}^t_\ell$, and writes the resulting correction into the current transformation's output error subspace.

C Experimental Details and Additional Results

We report results for additional model architectures, base merging methods, and NLG task combinations, along with full per-task results and variation across random seeds. Unless otherwise noted, post-hoc calibration results are reported as the mean $\pm$ sample standard deviation over five random seeds. For stochastic base merging methods, we also report five-seed statistics where available. Results from a single evaluation omit standard deviations.

C.1 Experimental Settings

Vision. We consider three CLIP vision transformer architectures, ViT-B/32, ViT-B/16, and ViT-L/14 (Radford et al., 2021). For each architecture, we use task-specific models on eight image classification datasets: SUN397 (Xiao et al., 2010), Cars (Krause et al., 2013), RESISC45 (Cheng et al., 2017), EuroSAT (Helber et al., 2019), SVHN (Netzer et al., 2011), GTSRB (Stallkamp et al., 2011), MNIST (LeCun et al., 1998), and DTD (Cimpoi et al., 2014). The eight task-specific models are merged within each architecture, and performance is evaluated on the corresponding test sets. We report task-wise classification accuracy and its average across the eight tasks.

NLU. We use RoBERTa-base (Liu et al., 2019) as the shared pretrained model and consider eight tasks from the GLUE benchmark (Wang et al., 2018): CoLA, SST-2, MRPC, STS-B, QQP, MNLI, QNLI, and RTE. We merge the eight task-specific models and evaluate the resulting model using the corresponding task-specific metrics. The reported average is the arithmetic mean of the eight task scores.

NLG. We use LLaMA-2-13B (Touvron et al., 2023) as the common base model and consider task experts specialized for instruction following, mathematical reasoning, and code generation. The instruction-following and mathematical-reasoning experts are based on WizardLM (Xu et al., 2024) and WizardMath (Luo et al., 2025), respectively. We evaluate all three pairwise combinations of the task experts. Detailed evaluation settings for the NLG experiments are provided together with the corresponding results.

C.2 Hyperparameter Search

We select hyperparameters independently for each base merging method and benchmark setting. Here, r denotes the calibration rank, η the learning rate, and E the number of training epochs. We choose method-specific rank ranges with reference to trainable parameter budgets. In the selected configurations, ESCal uses more trainable parameters than Surgery and fewer than SurgeryV2.

Vision. For Surgery and SurgeryV2, we fix the calibration rank at $r = 16$. Then we search the learning rate $\eta \in \{1e\text{-}4, 3e\text{-}4, 1e\text{-}3\}$ and the number of training epochs $E \in \{50, 100, 200, 250, 500\}$. The search extends to 500 epochs (1,000 outer iterations under our vision calibration setup), while the selected best setting for accuracy is 250 epochs (500 outer iterations). The separate timing comparison in Table 18 uses 200 epochs. For FeatCal, across all vision architectures and base merging methods, we fix $\lambda = 0.05$ and search $\rho \in \{1, 1.5, 2, 2.5\}$ and $\alpha \in \{0.1, 0.3, 0.5\}$. For ESCal, we fix $r = 128$ and search $\eta \in \{1e\text{-}4, 3e\text{-}4, 1e\text{-}3\}$ and $E \in \{50, 100, 200\}$.

NLU. For Surgery, SurgeryV2, and ESCal, we search $E \in \{200, 500, 1000\}$. For Surgery and SurgeryV2, we search $r \in \{4, 16, 64\}$ and $\eta \in \{1e\text{-}5, 3e\text{-}5, 1e\text{-}4, 3e\text{-}4, 1e\text{-}3, 3e\text{-}3, 1e\text{-}2\}$. For FeatCal, we search $\lambda \in \{1e\text{-}3, 3e\text{-}3, 1e\text{-}2, 0.05, 0.2\}$, $\rho \in \{1, 1.5, 2, 2.5\}$, and $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. For ESCal, we search $r \in \{32, 64, 128\}$ and $\eta \in \{1e\text{-}4, 3e\text{-}4, 1e\text{-}3, 3e\text{-}3, 6e\text{-}3, 1e\text{-}2\}$.

NLG. For Surgery, we search $r \in \{16, 32\}$, $\eta \in \{1e\text{-}6, 1e\text{-}5, 1e\text{-}4, 1e\text{-}3\}$, and $E \in \{5, 25, 50\}$. For SurgeryV2, we search $r \in \{8, 16\}$, $\eta \in \{1e\text{-}5, 1e\text{-}4, 1e\text{-}3\}$, and $E \in \{5, 25, 50\}$. Because NLG evaluation is computationally expensive, we use a smaller hyperparameter grid for FeatCal. We fix $\rho = 1$ and search $\lambda \in \{1e\text{-}3, 1e\text{-}1\}$ and $\alpha \in \{0.1, 0.3, 0.5\}$. For ESCal, we search $r \in \{64, 128, 256\}$, $\eta \in \{1e\text{-}6, 1e\text{-}5, 1e\text{-}4, 1e\text{-}3\}$, and $E \in \{5, 25, 50\}$.

C.3 Vision Results

Tables 9, 10, and 11 report the complete task-wise test accuracies for ViT-B/32, ViT-B/16, and ViT-L/14, respectively, across all six base merging methods and four post-merging calibration methods. The *Avg.* column reports the mean accuracy across the eight tasks.

ESCAL consistently improves over the corresponding uncalibrated models and achieves the highest average accuracy among the compared calibration methods in most settings. FEATCAL performs best for PRODISTILL across all three architectures.

Table 9: Full task-wise test accuracy (%) on ViT-B/32.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
TASK ARITHMETIC	55.32	54.98	66.82	77.78	80.21	69.68	97.34	50.37	69.06
w/ SURGERY	60.39 ± 0.46	52.62 ± 0.59	71.38 ± 1.76	88.65 ± 0.79	81.67 ± 1.65	74.27 ± 0.91	97.82 ± 0.16	52.77 ± 2.73	72.45 ± 0.52
w/ SURGERYV2	**70.70** ± 0.19	66.41 ± 0.49	90.65 ± 1.23	99.12 ± 0.18	96.61 ± 0.07	95.88 ± 0.62	99.47 ± 0.06	70.90 ± 0.75	86.22 ± 0.17
w/ FEATCAL	69.22	**70.31**	89.06	98.00	95.01	92.40	99.13	71.11	85.53
w/ ESCAL	70.33 ± 0.22	69.02 ± 0.19	**93.60** ± 0.36	**99.48** ± 0.07	**97.10** ± 0.05	**97.70** ± 0.22	**99.55** ± 0.03	**73.52** ± 0.44	**87.54** ± 0.07
TIES-MERGING	65.05	64.43	74.92	76.48	81.28	69.37	96.53	54.52	72.82
w/ SURGERY	67.64 ± 0.39	59.57 ± 0.54	78.50 ± 1.77	89.24 ± 1.30	81.30 ± 1.43	76.07 ± 0.29	97.14 ± 0.31	60.03 ± 1.81	76.19 ± 0.39
w/ SURGERYV2	70.87 ± 0.20	66.26 ± 0.75	90.14 ± 1.71	98.94 ± 0.21	96.54 ± 0.09	95.52 ± 0.77	99.45 ± 0.07	70.89 ± 0.72	86.08 ± 0.21
w/ FEATCAL	70.12 ± 0.02	**70.74** ± 0.17	89.28 ± 0.31	97.93 ± 0.13	94.71 ± 0.02	92.55 ± 0.28	99.10 ± 0.03	71.13 ± 0.70	85.69 ± 0.13
w/ ESCAL	**70.93** ± 0.23	69.23 ± 0.18	**93.31** ± 0.48	**99.36** ± 0.10	**97.09** ± 0.03	**97.70** ± 0.25	**99.53** ± 0.04	**73.35** ± 0.43	**87.56** ± 0.07
ADAMERGING (tw)	58.80 ± 3.16	53.42 ± 1.76	64.91 ± 6.63	86.51 ± 1.86	79.67 ± 4.41	83.34 ± 4.13	90.32 ± 2.60	49.55 ± 8.56	70.82 ± 1.04
w/ SURGERY	61.90 ± 2.48	49.77 ± 2.25	69.97 ± 7.03	92.50 ± 1.03	80.29 ± 4.93	85.75 ± 4.32	94.73 ± 1.04	52.66 ± 9.39	73.45 ± 1.22
w/ SURGERYV2	69.86 ± 1.86	61.67 ± 2.43	88.89 ± 2.83	99.25 ± 0.13	96.37 ± 0.22	96.97 ± 0.82	99.32 ± 0.15	68.57 ± 5.36	85.11 ± 0.55
w/ FEATCAL	68.86 ± 2.21	**66.32** ± 2.14	86.92 ± 2.34	98.42 ± 0.39	94.72 ± 0.58	94.68 ± 1.13	98.51 ± 0.37	69.31 ± 4.99	84.72 ± 0.52
w/ ESCAL	**70.10** ± 1.65	65.55 ± 1.43	**92.79** ± 1.32	**99.59** ± 0.05	**97.03** ± 0.12	**98.06** ± 0.27	**99.51** ± 0.05	**72.35** ± 3.07	**86.87** ± 0.29
ADAMERGING (lw)	63.28 ± 2.89	66.71 ± 2.16	79.15 ± 3.50	89.81 ± 1.31	83.82 ± 3.25	90.56 ± 3.91	97.87 ± 0.48	63.28 ± 3.76	79.31 ± 0.76
w/ SURGERY	66.33 ± 2.19	64.87 ± 2.88	84.24 ± 2.56	94.24 ± 0.46	87.64 ± 2.57	92.83 ± 3.24	98.30 ± 0.41	67.12 ± 3.61	81.95 ± 0.85
w/ SURGERYV2	**71.45** ± 1.26	70.24 ± 2.27	92.79 ± 0.71	99.14 ± 0.07	96.54 ± 0.12	98.08 ± 0.31	99.54 ± 0.03	**75.45** ± 1.56	87.90 ± 0.31
w/ FEATCAL	70.68 ± 1.70	**74.08** ± 1.41	91.16 ± 0.59	98.05 ± 0.13	94.69 ± 0.27	97.03 ± 0.52	99.21 ± 0.09	75.06 ± 1.31	87.50 ± 0.31
w/ ESCAL	69.51 ± 1.89	70.99 ± 1.43	**93.40** ± 0.36	**99.48** ± 0.11	**97.04** ± 0.06	**98.38** ± 0.23	**99.58** ± 0.03	74.87 ± 0.94	**87.91** ± 0.24
WUDI-MERGING	69.11	71.38	85.62	95.63	94.27	94.51	99.47	69.47	84.93
w/ SURGERY	69.24 ± 0.22	69.42 ± 0.22	87.70 ± 0.20	97.53 ± 0.33	94.81 ± 0.07	95.43 ± 0.34	99.48 ± 0.04	71.37 ± 0.49	85.62 ± 0.17
w/ SURGERYV2	73.24 ± 0.08	73.57 ± 0.45	94.15 ± 0.13	**99.59** ± 0.06	97.17 ± 0.07	98.38 ± 0.17	**99.64** ± 0.01	**77.66** ± 0.61	89.18 ± 0.09
w/ FEATCAL	71.98 ± 0.14	**75.80** ± 0.17	93.71 ± 0.08	99.31 ± 0.07	96.53 ± 0.04	98.42 ± 0.03	99.50 ± 0.01	77.52 ± 0.24	89.10 ± 0.05
w/ ESCAL	**73.55** ± 0.13	74.68 ± 0.15	**94.80** ± 0.14	99.58 ± 0.04	**97.31** ± 0.01	**98.68** ± 0.07	**99.64** ± 0.02	77.52 ± 0.40	**89.47** ± 0.06
PRODISTILL	68.84 ± 0.20	70.50 ± 0.29	90.74 ± 0.81	99.06 ± 0.14	96.35 ± 0.35	96.17 ± 0.63	99.50 ± 0.03	67.53 ± 0.74	86.09 ± 0.12
w/ SURGERY	69.55 ± 0.14	**70.56** ± 0.42	90.90 ± 0.80	99.08 ± 0.12	96.33 ± 0.35	96.07 ± 0.68	99.51 ± 0.03	68.98 ± 0.79	86.37 ± 0.13
w/ SURGERYV2	**70.09** ± 0.25	70.38 ± 0.35	91.21 ± 0.99	98.93 ± 0.16	96.85 ± 0.14	96.25 ± 0.67	99.48 ± 0.03	69.29 ± 0.88	86.56 ± 0.19
w/ FEATCAL	69.63 ± 0.14	69.80 ± 0.45	**92.60** ± 0.31	99.08 ± 0.16	**96.88** ± 0.11	96.35 ± 0.55	99.51 ± 0.04	**72.44** ± 0.68	**87.03** ± 0.13
w/ ESCAL	69.21 ± 0.26	70.44 ± 0.40	91.30 ± 0.87	**99.19** ± 0.13	96.87 ± 0.11	**96.89** ± 0.51	**99.53** ± 0.03	68.59 ± 1.02	86.50 ± 0.14

Table 10: Full task-wise test accuracy (%) on ViT-B/16.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
TASK ARITHMETIC	61.74	65.96	74.15	77.07	88.06	73.92	98.48	53.03	74.05
w/ SURGERY	65.25 ± 0.43	63.49 ± 1.63	76.73 ± 1.36	89.67 ± 0.99	88.03 ± 0.40	76.13 ± 2.11	98.39 ± 0.12	58.62 ± 2.21	77.04 ± 0.40
w/ SURGERYV2	74.17 ± 0.16	76.68 ± 0.75	93.02 ± 0.59	98.95 ± 0.24	97.21 ± 0.06	96.51 ± 0.62	99.52 ± 0.04	74.85 ± 0.95	88.86 ± 0.17
w/ FEATCAL	71.96 ± 0.06	**79.84** ± 0.19	90.12 ± 0.32	97.95 ± 0.26	96.44 ± 0.03	93.37 ± 0.03	99.16 ± 0.03	73.41 ± 0.65	87.78 ± 0.11
w/ ESCAL	**74.22** ± 0.27	79.82 ± 0.54	**95.07** ± 0.26	**99.46** ± 0.09	**97.61** ± 0.08	**98.47** ± 0.26	**99.58** ± 0.02	**77.31** ± 0.63	**90.19** ± 0.20
TIES-MERGING	69.02	72.63	80.70	84.56	85.08	71.56	98.19	54.95	77.09
w/ SURGERY	71.14 ± 0.22	68.61 ± 1.21	82.55 ± 2.36	92.36 ± 1.13	84.04 ± 1.31	75.18 ± 2.78	98.10 ± 0.26	63.81 ± 2.36	79.47 ± 0.40
w/ SURGERYV2	74.30 ± 0.09	75.93 ± 0.92	92.40 ± 1.02	98.70 ± 0.36	97.18 ± 0.09	96.22 ± 0.55	99.51 ± 0.02	74.89 ± 1.25	88.64 ± 0.20
w/ FEATCAL	73.67 ± 0.03	**81.47** ± 0.10	90.71 ± 0.19	98.01 ± 0.14	96.38 ± 0.01	93.99 ± 0.09	99.13 ± 0.03	74.04 ± 0.46	88.42 ± 0.07
w/ ESCAL	**74.50** ± 0.30	79.98 ± 0.32	**95.13** ± 0.26	**99.40** ± 0.09	**97.53** ± 0.06	**98.38** ± 0.28	**99.57** ± 0.02	**77.36** ± 0.86	**90.23** ± 0.15
ADAMERGING (tw)	66.63 ± 2.45	62.80 ± 4.01	71.15 ± 5.51	84.53 ± 6.46	86.88 ± 4.56	91.13 ± 1.93	96.95 ± 0.57	42.65 ± 7.08	75.34 ± 0.71
w/ SURGERY	68.31 ± 2.16	60.42 ± 4.33	75.25 ± 5.05	91.07 ± 3.68	87.31 ± 5.25	90.51 ± 2.74	97.08 ± 0.46	49.45 ± 8.83	77.43 ± 1.00
w/ SURGERYV2	74.03 ± 1.56	72.72 ± 3.69	91.17 ± 2.14	98.97 ± 0.37	97.07 ± 0.22	97.98 ± 0.46	99.44 ± 0.07	67.66 ± 5.50	87.38 ± 0.47
w/ FEATCAL	72.91 ± 1.72	75.85 ± 3.47	87.93 ± 2.15	98.01 ± 0.63	96.27 ± 0.49	96.52 ± 0.57	98.87 ± 0.10	62.30 ± 7.94	86.08 ± 0.66
w/ ESCAL	**74.84** ± 1.18	**77.85** ± 2.18	**94.78** ± 0.59	**99.47** ± 0.19	**97.50** ± 0.11	**98.80** ± 0.12	**99.51** ± 0.03	**72.20** ± 3.90	**89.37** ± 0.32
ADAMERGING (lw)	68.96 ± 1.85	80.02 ± 2.77	76.80 ± 6.10	86.30 ± 4.65	92.06 ± 0.63	94.35 ± 2.48	98.22 ± 0.59	62.54 ± 8.08	82.41 ± 1.08
w/ SURGERY	70.45 ± 1.38	78.87 ± 2.64	81.87 ± 3.01	91.79 ± 1.85	92.04 ± 0.84	94.88 ± 1.86	98.50 ± 0.47	68.65 ± 5.21	84.63 ± 0.71
w/ SURGERYV2	**75.42** ± 1.08	83.29 ± 1.17	93.90 ± 1.17	99.06 ± 0.18	97.24 ± 0.17	98.70 ± 0.18	99.55 ± 0.05	**77.53** ± 2.69	90.59 ± 0.26
w/ FEATCAL	72.58 ± 0.99	**84.67** ± 1.29	89.75 ± 3.05	98.09 ± 0.15	96.20 ± 0.20	97.11 ± 0.30	99.11 ± 0.13	74.84 ± 3.85	89.04 ± 0.45
w/ ESCAL	74.66 ± 0.69	83.25 ± 0.87	**95.06** ± 0.64	**99.43** ± 0.09	**97.58** ± 0.06	**98.85** ± 0.10	**99.57** ± 0.05	77.21 ± 2.40	**90.70** ± 0.26
WUDI-MERGING	73.29	81.71	90.07	98.19	95.30	96.55	99.44	73.78	88.54
w/ SURGERY	73.68 ± 0.18	81.00 ± 0.36	91.34 ± 0.30	98.60 ± 0.13	95.09 ± 0.21	96.42 ± 0.25	99.45 ± 0.02	76.21 ± 0.85	88.97 ± 0.11
w/ SURGERYV2	76.68 ± 0.11	84.13 ± 0.41	95.70 ± 0.14	**99.64** ± 0.08	97.67 ± 0.07	98.90 ± 0.10	**99.67** ± 0.01	80.45 ± 0.70	91.60 ± 0.10
w/ FEATCAL	75.64 ± 0.10	**84.62** ± 0.15	94.52 ± 0.10	98.96 ± 0.06	97.27 ± 0.03	98.46 ± 0.04	99.46 ± 0.02	79.50 ± 0.29	91.06 ± 0.02
w/ ESCAL	**76.78** ± 0.06	84.28 ± 0.27	**96.02** ± 0.05	99.61 ± 0.07	**97.78** ± 0.04	**98.95** ± 0.07	99.65 ± 0.02	**80.50** ± 0.19	**91.70** ± 0.05
PRODISTILL	72.58 ± 0.08	81.01 ± 0.28	93.07 ± 0.86	99.19 ± 0.10	97.34 ± 0.16	97.65 ± 0.33	99.59 ± 0.02	71.35 ± 0.89	88.97 ± 0.17
w/ SURGERY	73.26 ± 0.22	81.12 ± 0.36	92.98 ± 0.94	99.21 ± 0.11	97.34 ± 0.15	97.41 ± 0.38	99.59 ± 0.03	72.66 ± 1.07	89.20 ± 0.18
w/ SURGERYV2	**73.73** ± 0.18	**81.32** ± 0.51	93.14 ± 0.98	99.24 ± 0.15	97.42 ± 0.10	97.47 ± 0.28	99.60 ± 0.02	73.30 ± 0.79	89.40 ± 0.19
w/ FEATCAL	72.26 ± 0.35	80.29 ± 0.23	**94.65** ± 0.39	**99.27** ± 0.06	**97.53** ± 0.10	97.93 ± 0.22	**99.68** ± 0.01	**76.56** ± 0.50	**89.77** ± 0.05
w/ ESCAL	72.96 ± 0.02	81.02 ± 0.37	93.52 ± 0.70	99.19 ± 0.16	97.50 ± 0.13	**97.95** ± 0.27	99.60 ± 0.01	72.47 ± 1.04	89.27 ± 0.17

Table 11: Full task-wise test accuracy (%) on ViT-L/14.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
TASK ARITHMETIC	74.16	82.09	86.67	93.93	87.91	86.77	98.94	65.69	84.52
w/ SURGERY	74.71 ± 0.27	80.50 ± 0.43	87.89 ± 1.66	95.47 ± 0.57	88.82 ± 0.59	86.87 ± 1.01	98.94 ± 0.13	69.84 ± 1.94	85.38 ± 0.19
w/ SURGERYV2	78.61 ± 0.28	86.93 ± 0.31	94.57 ± 1.00	99.39 ± 0.04	97.73 ± 0.04	97.55 ± 0.54	99.58 ± 0.03	78.62 ± 0.99	91.62 ± 0.11
w/ FEATCAL	79.05 ± 0.04	**90.18** ± 0.09	94.11 ± 0.11	99.30 ± 0.05	96.47 ± 0.04	96.85 ± 0.08	99.36 ± 0.03	78.96 ± 0.06	91.78 ± 0.02
w/ ESCAL	**79.12** ± 0.17	89.28 ± 0.16	**96.11** ± 0.28	**99.66** ± 0.05	**97.98** ± 0.03	**98.75** ± 0.23	**99.64** ± 0.02	**80.52** ± 0.28	**92.63** ± 0.05
TIES-MERGING	75.03	84.54	87.86	94.70	85.66	82.07	98.65	67.50	84.50
w/ SURGERY	75.63 ± 0.17	81.36 ± 0.54	88.41 ± 2.55	95.84 ± 0.54	86.21 ± 0.96	83.44 ± 2.05	98.61 ± 0.17	70.46 ± 2.15	84.99 ± 0.18
w/ SURGERYV2	78.26 ± 0.23	85.99 ± 0.51	93.92 ± 1.11	99.22 ± 0.16	97.63 ± 0.07	96.52 ± 0.88	99.58 ± 0.04	77.60 ± 1.04	91.09 ± 0.14
w/ FEATCAL	**79.25** ± 0.06	**90.28** ± 0.07	94.02 ± 0.11	99.42 ± 0.06	96.68 ± 0.03	96.59 ± 0.06	99.40 ± 0.02	78.67 ± 0.20	91.79 ± 0.03
w/ ESCAL	78.95 ± 0.16	89.25 ± 0.21	**96.01** ± 0.41	**99.59** ± 0.09	**97.90** ± 0.02	**98.55** ± 0.24	**99.61** ± 0.03	**80.39** ± 0.30	**92.53** ± 0.06
ADAMERGING (tw)	73.63 ± 2.88	81.74 ± 2.92	79.62 ± 4.97	90.39 ± 2.74	87.37 ± 1.60	96.10 ± 0.31	97.48 ± 0.34	70.63 ± 4.05	84.62 ± 0.66
w/ SURGERY	74.57 ± 2.53	81.35 ± 3.45	82.71 ± 4.36	94.08 ± 2.23	88.03 ± 1.42	96.21 ± 0.52	97.83 ± 0.32	73.29 ± 4.13	86.01 ± 0.64
w/ SURGERYV2	79.77 ± 1.49	88.15 ± 2.05	93.48 ± 1.80	99.44 ± 0.12	97.69 ± 0.05	99.07 ± 0.10	99.54 ± 0.04	81.93 ± 1.08	92.38 ± 0.28
w/ FEATCAL	79.90 ± 1.60	**90.73** ± 1.15	92.30 ± 1.90	99.35 ± 0.22	96.20 ± 0.26	98.80 ± 0.10	99.15 ± 0.10	81.97 ± 1.22	92.30 ± 0.20
w/ ESCAL	**79.97** ± 0.71	89.69 ± 1.07	**95.42** ± 0.57	**99.69** ± 0.04	**97.93** ± 0.06	**99.15** ± 0.05	**99.61** ± 0.03	**82.16** ± 0.93	**92.95** ± 0.11
ADAMERGING (lw)	76.97 ± 1.29	88.59 ± 1.48	88.31 ± 3.43	95.52 ± 1.18	91.38 ± 2.02	97.60 ± 0.49	98.79 ± 0.40	78.39 ± 1.46	89.44 ± 0.54
w/ SURGERY	77.79 ± 1.25	88.58 ± 0.99	90.96 ± 3.21	96.81 ± 0.45	91.97 ± 2.48	97.74 ± 0.35	98.82 ± 0.32	79.72 ± 1.83	90.30 ± 0.60
w/ SURGERYV2	**81.11** ± 0.42	91.15 ± 0.41	95.90 ± 0.68	99.59 ± 0.15	97.65 ± 0.10	99.13 ± 0.09	**99.61** ± 0.07	**83.40** ± 0.46	93.44 ± 0.12
w/ FEATCAL	80.62 ± 0.15	**91.42** ± 0.13	94.50 ± 0.71	99.39 ± 0.09	95.95 ± 0.32	98.69 ± 0.09	99.32 ± 0.10	83.27 ± 0.51	92.89 ± 0.11
w/ ESCAL	80.91 ± 0.37	91.34 ± 0.29	**96.21** ± 0.30	**99.73** ± 0.08	**97.90** ± 0.03	**99.17** ± 0.03	**99.61** ± 0.04	82.83 ± 0.51	**93.46** ± 0.08
WUDI-MERGING	79.65	90.95	93.98	99.41	96.21	98.04	99.63	80.64	92.31
w/ SURGERY	79.81 ± 0.12	90.82 ± 0.12	94.65 ± 0.44	99.49 ± 0.03	96.30 ± 0.12	98.11 ± 0.18	99.69 ± 0.02	80.90 ± 0.60	92.47 ± 0.11
w/ SURGERYV2	**81.13** ± 0.12	91.36 ± 0.22	96.64 ± 0.14	99.74 ± 0.04	98.06 ± 0.03	99.18 ± 0.04	**99.73** ± 0.01	83.21 ± 0.37	93.63 ± 0.02
w/ FEATCAL	79.92 ± 0.03	90.64 ± 0.05	95.91 ± 0.07	99.74 ± 0.04	97.73 ± 0.02	99.17 ± 0.02	99.70 ± 0.01	**83.91** ± 0.11	93.34 ± 0.02
w/ ESCAL	81.08 ± 0.13	**91.56** ± 0.11	**96.76** ± 0.13	**99.79** ± 0.05	**98.14** ± 0.02	**99.21** ± 0.03	99.69 ± 0.02	82.86 ± 0.21	**93.64** ± 0.04
PRODISTILL	77.70 ± 0.17	**89.96** ± 0.21	95.16 ± 0.67	99.56 ± 0.07	97.79 ± 0.11	98.40 ± 0.34	99.60 ± 0.03	78.11 ± 0.49	92.04 ± 0.07
w/ SURGERY	77.89 ± 0.16	89.72 ± 0.14	95.18 ± 0.76	99.59 ± 0.10	97.79 ± 0.10	98.39 ± 0.35	99.61 ± 0.01	78.40 ± 0.18	92.07 ± 0.04
w/ SURGERYV2	77.98 ± 0.15	89.74 ± 0.21	95.22 ± 0.69	99.56 ± 0.10	97.80 ± 0.09	98.38 ± 0.36	**99.62** ± 0.01	78.53 ± 0.18	92.11 ± 0.06
w/ FEATCAL	**79.14** ± 0.17	89.71 ± 0.22	**96.17** ± 0.24	**99.62** ± 0.04	**97.91** ± 0.08	**98.66** ± 0.22	99.60 ± 0.03	**81.11** ± 0.42	**92.74** ± 0.10
w/ ESCAL	77.88 ± 0.14	89.91 ± 0.26	95.32 ± 0.61	99.56 ± 0.06	97.84 ± 0.09	98.46 ± 0.33	99.61 ± 0.02	78.37 ± 0.18	92.12 ± 0.03

C.4 NLU RESULTS

Table 12 reports the complete task-wise results for RoBERTa-base across all six base merging methods and the corresponding post-merging calibration methods. The *Avg.* column reports the arithmetic mean across the eight NLU tasks, with the pretrained and expert models included as reference points.

ESCAL improves over all uncalibrated models and achieves the highest average score among most compared calibration methods. The exception is TIES-MERGING, where ESCAL and SURGERYV2 obtain comparable average scores.

Table 12: Full task-wise performance on the NLU benchmark using RoBERTa-base.

Method	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	Avg.
Pretrained	0.0000	0.5975	0.1581	-0.0758	0.4633	0.3183	0.5078	0.5271	0.3120
Expert	0.5895	0.9438	0.9091	0.8997	0.8994	0.8718	0.9198	0.7726	0.8507
TASK ARITHMETIC	0.1232	0.6135	0.5500	0.5356	0.5881	0.6270	0.5365	0.5343	0.5135
w/ SURGERY	0.2553 ± 0.0253	0.8532 ± 0.0124	0.7657 ± 0.0155	0.7877 ± 0.0089	0.7793 ± 0.0192	0.7215 ± 0.0118	0.7839 ± 0.0224	0.7184 ± 0.0099	0.7081 ± 0.0088
w/ SURGERYV2	0.5358 ± 0.0106	**0.9326** ± 0.0029	0.8908 ± 0.0037	0.8978 ± 0.0018	0.8002 ± 0.0137	0.8404 ± 0.0016	0.9084 ± 0.0019	0.7408 ± 0.0110	0.8184 ± 0.0016
w/ FEATCAL	0.3879 ± 0.0111	0.9140 ± 0.0095	0.8162 ± 0.0116	0.4900 ± 0.0353	0.7680 ± 0.0222	0.7569 ± 0.0026	0.8499 ± 0.0116	0.7011 ± 0.0065	0.7105 ± 0.0024
w/ ESCAL	**0.5588** ± 0.0090	0.9319 ± 0.0026	**0.8937** ± 0.0039	**0.8998** ± 0.0019	**0.8061** ± 0.0204	**0.8469** ± 0.0023	**0.9117** ± 0.0029	**0.7415** ± 0.0137	**0.8238** ± 0.0037
TIES-MERGING	0.1972	0.7626	0.5075	0.4390	0.6039	0.6670	0.5380	0.4946	0.5262
w/ SURGERY	0.2680 ± 0.0401	0.8635 ± 0.0213	0.7570 ± 0.0082	0.7770 ± 0.0222	0.7770 ± 0.0231	0.7347 ± 0.0130	0.7988 ± 0.0067	0.6989 ± 0.0257	0.7094 ± 0.0126
w/ SURGERYV2	0.5420 ± 0.0127	0.9319 ± 0.0021	**0.8848** ± 0.0069	**0.8965** ± 0.0030	**0.8205** ± 0.0117	**0.8474** ± 0.0016	**0.9079** ± 0.0027	0.7394 ± 0.0118	**0.8213** ± 0.0027
w/ FEATCAL	0.4055 ± 0.0130	0.9106 ± 0.0110	0.8111 ± 0.0130	0.4765 ± 0.0498	0.7702 ± 0.0183	0.7569 ± 0.0039	0.8433 ± 0.0115	0.7061 ± 0.0121	0.7100 ± 0.0046
w/ ESCAL	**0.5552** ± 0.0253	**0.9339** ± 0.0044	0.8769 ± 0.0113	0.8935 ± 0.0020	0.8144 ± 0.0231	0.8443 ± 0.0024	0.9070 ± 0.0043	**0.7430** ± 0.0103	0.8210 ± 0.0052
ADAMERGING (tw)	0.0283	0.4805	0.2626	0.7031	0.2204	0.8103	0.7818	0.7040	0.4989
w/ SURGERY	0.1338 ± 0.0470	0.8234 ± 0.0293	0.7918 ± 0.0095	0.7757 ± 0.0188	0.7438 ± 0.0072	0.8336 ± 0.0034	0.8711 ± 0.0066	0.7401 ± 0.0068	0.7142 ± 0.0076
w/ SURGERYV2	0.5227 ± 0.0259	0.9339 ± 0.0056	0.8850 ± 0.0079	**0.8910** ± 0.0031	0.7570 ± 0.0195	**0.8651** ± 0.0015	**0.9151** ± 0.0015	**0.7783** ± 0.0253	0.8185 ± 0.0071
w/ FEATCAL	0.4004 ± 0.0193	0.9183 ± 0.0075	0.8189 ± 0.0057	0.4628 ± 0.0479	**0.7737** ± 0.0164	0.7525 ± 0.0032	0.8194 ± 0.0106	0.6606 ± 0.0142	0.7008 ± 0.0042
w/ ESCAL	**0.5383** ± 0.0163	**0.9344** ± 0.0036	**0.8892** ± 0.0074	0.8905 ± 0.0023	0.7594 ± 0.0243	0.8648 ± 0.0017	0.9110 ± 0.0028	0.7740 ± 0.0083	**0.8202** ± 0.0048
ADAMERGING (lw)	0.0137	0.5115	0.3069	0.7205	0.1868	0.8038	0.6077	0.7076	0.4823
w/ SURGERY	0.1864 ± 0.0594	0.8314 ± 0.0149	0.7948 ± 0.0099	0.7699 ± 0.0338	0.7763 ± 0.0111	0.8182 ± 0.0077	0.8242 ± 0.0106	0.7372 ± 0.0181	0.7173 ± 0.0122
w/ SURGERYV2	0.5145 ± 0.0201	0.9333 ± 0.0074	0.8867 ± 0.0055	0.8930 ± 0.0031	0.8073 ± 0.0204	0.8585 ± 0.0008	0.9080 ± 0.0028	**0.7711** ± 0.0197	0.8215 ± 0.0059
w/ FEATCAL	0.4052 ± 0.0288	0.9092 ± 0.0136	0.8297 ± 0.0133	0.4788 ± 0.0306	0.7767 ± 0.0181	0.7368 ± 0.0027	0.8354 ± 0.0092	0.6801 ± 0.0261	0.7065 ± 0.0030
w/ ESCAL	**0.5391** ± 0.0140	**0.9339** ± 0.0029	**0.8926** ± 0.0057	**0.8968** ± 0.0010	**0.8204** ± 0.0175	**0.8592** ± 0.0015	**0.9082** ± 0.0035	0.7704 ± 0.0119	**0.8276** ± 0.0040
WUDI-MERGING	0.0625	0.8073	0.2565	0.0927	0.7680	0.4431	0.4622	0.4874	0.4225
w/ SURGERY	0.2226 ± 0.0296	0.9025 ± 0.0095	0.7860 ± 0.0111	0.7758 ± 0.0264	0.8270 ± 0.0054	0.7755 ± 0.0179	0.8118 ± 0.0165	0.7206 ± 0.0139	0.7277 ± 0.0059
w/ SURGERYV2	0.5529 ± 0.0130	0.9392 ± 0.0049	0.8908 ± 0.0087	0.8868 ± 0.0027	**0.8798** ± 0.0032	**0.8620** ± 0.0017	0.9136 ± 0.0016	**0.7603** ± 0.0124	0.8357 ± 0.0018
w/ FEATCAL	0.4387 ± 0.0245	0.9190 ± 0.0084	0.8215 ± 0.0082	0.4953 ± 0.0427	0.8234 ± 0.0112	0.7748 ± 0.0033	0.8358 ± 0.0132	0.6888 ± 0.0213	0.7247 ± 0.0069
w/ ESCAL	**0.5663** ± 0.0217	**0.9397** ± 0.0055	**0.8950** ± 0.0051	**0.8953** ± 0.0027	0.8744 ± 0.0061	0.8616 ± 0.0013	**0.9144** ± 0.0025	0.7545 ± 0.0120	**0.8376** ± 0.0024
PRODISTILL (ew)	0.4076	0.9106	0.8463	0.5288	0.7938	0.7759	0.8609	0.7148	0.7298
w/ SURGERY	0.4470 ± 0.0225	0.9158 ± 0.0071	0.8371 ± 0.0075	0.7119 ± 0.0494	0.7756 ± 0.0253	0.7851 ± 0.0061	0.8647 ± 0.0032	0.7249 ± 0.0164	0.7578 ± 0.0049
w/ SURGERYV2	**0.5006** ± 0.0219	**0.9257** ± 0.0042	0.8717 ± 0.0074	0.8001 ± 0.0274	**0.8089** ± 0.0233	0.8283 ± 0.0032	0.9012 ± 0.0023	**0.7256** ± 0.0137	0.7953 ± 0.0051
w/ FEATCAL	0.4384 ± 0.0244	0.9122 ± 0.0045	0.8246 ± 0.0148	0.4523 ± 0.0716	0.8062 ± 0.0160	0.7914 ± 0.0074	0.8493 ± 0.0138	0.7141 ± 0.0160	0.7235 ± 0.0116
w/ ESCAL	0.4880 ± 0.0297	0.9239 ± 0.0045	**0.8759** ± 0.0069	**0.8734** ± 0.0106	0.7913 ± 0.0388	**0.8292** ± 0.0023	**0.9017** ± 0.0027	0.7155 ± 0.0179	**0.7999** ± 0.0062

C.5 NLG RESULTS

We further evaluate post-merging calibration on natural language generation (NLG) using LLaMA-2-13B experts specialized for instruction following, mathematical reasoning, and code generation, considering all three pairwise merges.

Aggregation and expert references. For each expert pair, Avg. is the arithmetic mean over all benchmarks shown in the corresponding table, including AlpacaEval when present. Norm Avg. divides each benchmark score by the score of the corresponding calibration target expert and then averages these ratios. For benchmark set $\mathcal{B}$, we report

$$\text{Avg.}(m) = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} s_{m,b}, \qquad \text{Norm Avg.}(m) = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \frac{s_{m,b}}{s_{e(b),b}}, \tag{18}$$

where $s_{m,b}$ is the score of model m on benchmark b, and $e(b)$ is the designated calibration target expert for that benchmark. The reference is fixed across all rows within each table. A per-benchmark normalized score above one indicates performance above the corresponding target expert; a normalized average above one need not indicate improvements on every benchmark.

For math–code merging, WizardMath-13B is the reference for GSM8K and MATH, while Llama-2-13b-code-alpaca is the reference for HumanEval and MBPP. For instruction–code merging, Llama-2-13b-code-alpaca is the reference for HumanEval and MBPP, and WizardLM-13B-V1.2 is the reference for AlpacaEval. For instruction–math merging, WizardMath-13B is the reference for GSM8K and MATH, while WizardLM-13B-V1.2 is the reference for HumanEval, MBPP, and AlpacaEval. The reference therefore follows the calibration target rather than whichever constituent expert obtains the higher benchmark score. The reported aggregates are computed from the displayed benchmark scores and rounded to four decimal places.

Table 13: Post-merging calibration for LLaMA-2-13B math–code merging.

Method	GSM8K	MATH	HumanEval	MBPP	Avg.	Norm Avg.
WizardMath-13B	0.6343	0.1468	0.0732	0.2156	0.2675	0.7655
Llama-2-13b-code-alpaca	0.0000	0.0000	0.2195	0.2959	0.1289	0.5000
TASK ARITHMETIC	0.6494	0.1446	0.0122	0.0917	0.2245	0.5936
w/ SURGERY	0.6542	0.1468	0.0488	0.1514	0.2503	0.6913
w/ SURGERYV2	0.6255	0.1430	0.0793	0.2133	0.2653	0.7606
w/ FEATCAL	0.6406	0.1452	0.2073	0.2683	**0.3154**	**0.9625**
w/ ESCAL	0.6351	0.1544	0.1829	0.2523	0.3062	0.9348
TIES-MERGING	0.5912	0.1194	0.0976	0.1330	0.2353	0.6599
w/ SURGERY	0.6159	0.1202	0.0976	0.1628	0.2491	0.6962
w/ SURGERYV2	0.5976	0.1180	0.1890	0.2867	0.2978	0.8940
w/ FEATCAL	0.6263	0.1352	0.1829	0.2569	0.3003	0.9025
w/ ESCAL	0.6112	0.1298	0.1951	0.2798	**0.3040**	**0.9206**

Table 14: Post-merging calibration for LLaMA-2-13B instruction–code merging.

Method	HumanEval	MBPP	AlpacaEval	Avg.	Norm Avg.
WizardLM-13B-V1.2	0.3293	0.3509	0.1646	0.2816	1.2287
Llama-2-13b-code-alpaca	0.2195	0.2959	0.0196	0.1783	0.7064
TASK ARITHMETIC	0.3232	0.3440	0.1576	0.2749	1.1975
w/ SURGERY	0.3171	0.3486	0.1604	**0.2754**	1.1991
w/ SURGERYV2	0.3293	0.3349	0.1615	0.2752	**1.2044**
w/ FEATCAL	0.3110	0.3372	0.1520	0.2667	1.1600
w/ ESCAL	0.3110	0.3417	0.1570	0.2699	1.1752
TIES-MERGING	0.2317	0.3326	0.0501	0.2048	0.8280
w/ SURGERY	0.2561	0.3394	0.0527	0.2161	0.8780
w/ SURGERYV2	0.2622	0.3372	0.0601	0.2198	0.8997
w/ FEATCAL	0.2744	0.3211	0.0622	0.2192	0.9044
w/ ESCAL	0.2805	0.3326	0.0819	**0.2317**	**0.9665**

Calibration when the merged model already performs well. Task Arithmetic already outperforms its calibration targets on some benchmarks. For example, Task Arithmetic achieves a GSM8K score

Table 15: Post-merging calibration for LLaMA-2-13B instruction–math merging.

Method	GSM8K	MATH	HumanEval	MBPP	AlpacaEval	Avg.	Norm Avg.
WizardLM-13B-V1.2	0.0223	0.0004	0.3293	0.3280	0.1646	0.1689	0.6076
WizardMath-13B	0.6343	0.1468	0.0732	0.2100	0.0487	0.2226	0.6317
TASK ARITHMETIC	0.6614	0.1440	0.2622	0.3000	0.1053	0.2946	0.8749
w/ SURGERY	0.6693	0.1358	0.2683	0.3300	0.1101	**0.3027**	**0.8940**
w/ SURGERYV2	0.6566	0.1434	0.2500	0.3160	0.1149	0.2962	0.8865
w/ FEATCAL	0.6454	0.1404	0.2439	0.3040	0.1041	0.2876	0.8548
w/ ESCAL	0.6653	0.1450	0.2561	0.2980	0.1073	0.2943	0.8749
TIES-MERGING	0.5761	0.1042	0.0915	0.3080	0.0609	0.2281	0.6410
w/ SURGERY	0.6008	0.1142	0.1951	0.3240	0.0899	0.2648	0.7703
w/ SURGERYV2	0.6064	0.1156	0.2317	0.3220	0.0920	0.2735	0.7975
w/ FEATCAL	0.6375	0.1366	0.1890	0.3020	0.1095	0.2749	**0.8191**
w/ ESCAL	0.6104	0.1252	0.2378	0.3340	0.0859	**0.2787**	0.8155

of 0.6614 in the instruction–math merge, exceeding WizardMath-13B's 0.6343. In instruction–code merging, its HumanEval and MBPP scores also exceed those of the code expert, although they remain below WizardLM-13B-V1.2. Its normalized average is 1.1975 relative to the designated targets, although its AlpacaEval score remains below the instruction expert. A merged model can also exceed both experts' overall averages by combining their strengths, even if it does not outperform the stronger expert on every benchmark.

SURGERY, SURGERYV2, FEATCAL, and ESCAL aim to reduce merging-induced discrepancies by using the constituent experts as calibration targets or references. When merging already improves a benchmark score over the corresponding expert, closer alignment with that expert need not yield further performance gains. Calibration may still improve other benchmarks. For Task Arithmetic, its effect on the overall average varies across calibration methods. In instruction–code merging, the full-benchmark average changes from 0.2749 to 0.2754 with SURGERY and 0.2752 with SURGERYV2, but to 0.2667 with FEATCAL and 0.2699 with ESCAL.

Calibration when merging leaves an expert-performance gap. The TIES merges remain below their designated calibration targets on average across all three expert pairs. ESCAL improves both Avg. and Norm Avg. for all three TIES merges, and achieves the highest values for both metrics among the evaluated calibration variants in the math–code and instruction–code settings. For math–code merging, it increases Avg. from 0.2353 to 0.3040 and Norm Avg. from 0.6599 to 0.9206. For instruction–code merging, the corresponding improvements are 0.2048 to 0.2317 and 0.8280 to 0.9665. These results show that corrections between fixed error subspaces can improve performance when the merged model remains below its calibration targets.

D ADDITIONAL ANALYSES

D.1 ERROR-SUBSPACE CONSTRUCTION

Table 16 provides the full per-task results for the basis-source comparison in Table 3, together with the bottom-r merging-error baseline. Bases derived from the top-r singular directions of the merging error achieve the highest mean accuracy on all eight tasks. Their average accuracy is 87.54%, compared with 87.01% for task-vector bases and 84.80% for fine-tuned-weight bases. Random orthonormal bases and bottom-r merging-error bases yield 84.08% and 80.86%, respectively. These results support using the dominant singular subspaces of the expert–merged weight discrepancy to constrain calibration.

Table 16: **Full results for the error-subspace construction ablation.** We report per-task and average accuracy (%; mean ± standard deviation) for the basis-source comparison in Table 3, together with the bottom-r merging-error baseline.

Basis construction	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Random orthonormal	65.42 ± 0.27	63.88 ± 0.55	88.54 ± 0.57	98.61 ± 0.23	96.06 ± 0.10	95.81 ± 0.22	99.32 ± 0.05	64.98 ± 1.02	84.08 ± 0.15
Fine-tuned weights, top-r	66.94 ± 0.22	65.02 ± 0.48	89.85 ± 0.61	99.03 ± 0.15	96.17 ± 0.04	96.32 ± 0.45	99.36 ± 0.06	65.70 ± 0.84	84.80 ± 0.24
Task vector, top-r	69.29 ± 0.15	67.72 ± 0.67	93.01 ± 0.45	99.44 ± 0.11	97.08 ± 0.04	97.47 ± 0.28	99.53 ± 0.04	72.53 ± 0.39	87.01 ± 0.09
Merging error, bottom-r	62.11 ± 0.21	62.17 ± 0.30	82.26 ± 0.36	97.20 ± 0.35	94.02 ± 0.19	91.05 ± 0.17	98.83 ± 0.07	59.21 ± 0.58	80.86 ± 0.13
Merging error, top-r	**70.33** ± 0.22	**69.02** ± 0.19	**93.60** ± 0.36	**99.48** ± 0.07	**97.10** ± 0.05	**97.70** ± 0.22	**99.55** ± 0.03	**73.52** ± 0.44	**87.54** ± 0.07

D.2 COMPUTATIONAL COST

We evaluate trainable parameter counts, additional inference-time storage, and computation time on ViT-B/32 with $T = 8$ tasks and 64 calibration examples per task. Storage and inference measurements assume a single active task.

Parameters and storage. Table 17 summarizes the trainable parameters across all eight tasks and the additional tensor storage required for single-task inference, relative to the shared image encoder. We use $L = 12$, $d = 768$, $d_{\text{out}} = 512$, $r_s = 16$, and $r_b = 128$.

Table 17: Trainable parameters across eight tasks and additional storage for single-task inference on ViT-B/32.

	Training ($T = 8$ tasks)		Inference (one task)			
Method	**Trainable Parameters**	**# Trainable**	**Stored Entries**	**# Stored**	**BF16 Storage**	**Overhead**
Shared ViT-B/32 image encoder	–	–	P_{img}	87,849,216	167.56 MiB	Reference
SURGERY	$2Td_{\text{out}}r_s$	131,072	$2d_{\text{out}}r_s$	16,384	0.031 MiB	0.02%
SURGERYV2	$T(2Ldr_s + 2d_{\text{out}}r_s)$	2,490,368	$2Ldr_s + 2d_{\text{out}}r_s$	311,296	0.594 MiB	0.35%
FEATCAL	0 (closed form)	0	0	0	0.000 MiB	0.00%
ESCAL	TLr_b^2	1,572,864	$Lr_b^2 + 2Ldr_b$	2,555,904	4.875 MiB	2.91%

ESCAL learns $TLr_b^2 = 1{,}572{,}864$ core parameters across eight tasks, which is 36.8% fewer than SURGERYV2. For single-task inference, it stores $Lr_b^2 = 196{,}608$ core entries and $2Ldr_b = 2{,}359{,}296$ fixed read/write basis entries. The resulting 2,555,904 additional entries require 4.875 MiB at BF16, equivalent to 2.91% of the shared image encoder's storage. Thus, ESCAL reduces the number of trainable parameters relative to SURGERYV2, while requiring more inference-time storage for its fixed bases.

SURGERY applies low-rank calibration at the output embedding, while SURGERYV2 additionally applies calibration at each residual block. FEATCAL updates the merged encoder in closed form and stores the resulting weights in place, requiring no additional inference-time tensors for the active task. This accounting excludes activations, optimizer states, temporary buffers, and serialization metadata; it does not quantify peak training memory or total training computation.

Construction and calibration time. Table 18 reports base-model construction and calibration times separately. We compare the time needed for base merging followed by calibration with that of more expensive merging procedures. The total time is the sum of base-model construction and calibration. Stage timings include setup, training or closed-form updates, and saving model artifacts, but exclude test evaluation, library imports, and reporting.

Table 18: Base-model construction and calibration times in minutes on ViT-B/32 with eight tasks and 64 calibration examples per task (single run on an NVIDIA RTX A6000). Calibration times exclude base-model construction; gradient-based calibration uses 200 epochs. The task-wise and layer-wise variants of ADAMERGING are denoted by tw and lw, respectively.

	Base construction	Calibration			
Base merging method	**Time**	**SURGERY**	**SURGERYV2**	**FEATCAL**	**ESCAL**
TASK ARITHMETIC	0.09	6.28	13.41	0.63	13.22
TIES-MERGING	0.39	5.48	12.05	0.58	12.92
WUDI-MERGING	5.61	5.53	12.18	0.53	13.05
ADAMERGING (tw)	39.46	6.87	13.69	0.65	14.06
ADAMERGING (lw)	40.04	6.96	13.69	0.89	14.05
PRODISTILL	23.26	6.73	13.38	0.61	14.05

For this timing comparison, we use 200 calibration epochs for all gradient-based methods: 400 outer iterations for SURGERY and SURGERYV2, and 200 epochs for ESCAL. With 64 calibration examples per task and a batch size of 16, each method performs 800 updates per task. This timing protocol is separate from the best-setting accuracy evaluation. FEATCAL uses the same calibration examples for its closed-form update. Base-model construction uses 500 epochs for ADAMERGING and 100 epochs per block for PRODISTILL; for TASK ARITHMETIC, it includes loading, merging, and saving the model.

Inference latency and throughput. We compare inference cost with and without calibration modules on a TASK ARITHMETIC ViT-B/32 model. As shown in Table 19, ESCAL increases per-image latency by 28.53% at batch size 1 and reduces throughput by 5.29% at batch size 64. These overheads are similar to those of SURGERYV2 (31.91% and 5.04%, respectively) in this setting.

We benchmark a model merged with coefficient 0.3, using SUN397 as the active task, on an NVIDIA RTX A6000 with BF16 autocast. Timing covers the image encoder, calibration modules, and classification head, with preprocessed inputs already on the GPU. We perform 10 paired repetitions of each calibrated model and its uncalibrated counterpart, alternating their execution order. Each repetition uses 100 warm-up passes and 200 timed passes with CUDA synchronization.

Table 19: **Inference cost relative to the uncalibrated TASK ARITHMETIC ViT-B/32 model.** None denotes the model without calibration. Absolute values are mean $\pm$ standard deviation. Δt denotes additional per-image latency. Relative changes are computed against paired uncalibrated runs and reported with 95% bootstrap confidence intervals; they may therefore differ from ratios based on the pooled None row.

	Batch size 1			Batch size 64		
Calibration	**Image latency (ms)**	**Δt (ms)**	**Overhead (%)**	**Batch latency (ms)**	**Throughput (img/s)**	**Throughput reduction (%)**
None	5.51 ± 0.02	Ref.	Ref.	14.83 ± 0.05	4316.5 ± 15.6	Ref.
SURGERY	5.60 ± 0.03	+0.095	+1.72 [+1.51, +1.95]	14.81 ± 0.07	4320.1 ± 20.0	+0.15 [+0.06, +0.25]
SURGERYV2	7.27 ± 0.02	+1.758	+31.91 [+31.60, +32.27]	15.63 ± 0.04	4095.0 ± 11.6	+5.04 [+4.99, +5.08]
FEATCAL	5.50 ± 0.02	+0.004	+0.07 [-0.30, +0.48]	14.84 ± 0.03	4311.4 ± 8.3	+0.01 [-0.01, +0.04]
ESCAL	7.09 ± 0.02	+1.575	+28.53 [+28.23, +28.84]	15.66 ± 0.04	4086.9 ± 10.3	+5.29 [+5.22, +5.35]

D.3 CALIBRATION DATA BUDGET

We evaluate how ESCAL performance changes with the number of calibration examples per task on ViT-B/32 with TASK ARITHMETIC as the base merging method, using 64 examples as the reference. For each data budget, we use the same merged checkpoint and vary only the task-specific calibration examples. We report task-wise accuracy and its average across tasks.

Table 20: Effect of the per-task support size on ESCAL, ViT-B/32 with a task-arithmetic base. Epochs are held at 200, so the optimizer's update count scales with the support size. Accuracy is on the full test set; the default setting (64) is set in bold.

Support / task	**SUN397**	**Cars**	**RESISC45**	**EuroSAT**	**SVHN**	**GTSRB**	**MNIST**	**DTD**	**Avg.**
4	66.64 ± 0.21	63.17 ± 1.20	85.49 ± 1.08	91.32 ± 3.07	93.87 ± 0.33	92.01 ± 1.37	97.47 ± 0.49	60.33 ± 1.44	81.29 ± 0.36
8	67.83 ± 0.42	65.05 ± 1.21	88.01 ± 0.91	94.77 ± 3.28	95.47 ± 0.33	93.60 ± 0.75	98.27 ± 0.64	63.48 ± 1.61	83.31 ± 0.49
16	68.71 ± 0.27	66.50 ± 0.84	90.03 ± 0.56	96.72 ± 0.91	96.02 ± 0.51	94.87 ± 0.69	98.79 ± 0.34	65.49 ± 0.86	84.64 ± 0.34
32	69.68 ± 0.14	67.77 ± 0.41	92.35 ± 0.28	99.09 ± 0.23	96.79 ± 0.11	96.90 ± 0.63	99.38 ± 0.07	69.89 ± 1.42	86.48 ± 0.21
64	70.33 ± 0.22	69.02 ± 0.19	93.60 ± 0.36	99.48 ± 0.07	97.10 ± 0.05	97.70 ± 0.22	99.55 ± 0.03	73.52 ± 0.44	87.54 ± 0.07
128	70.66 ± 0.08	69.70 ± 0.24	94.09 ± 0.19	99.57 ± 0.04	97.22 ± 0.04	98.18 ± 0.18	99.64 ± 0.02	75.46 ± 0.21	88.07 ± 0.06
256	**70.76** ± 0.18	**70.11** ± 0.27	**94.33** ± 0.15	**99.64** ± 0.03	**97.29** ± 0.03	**98.37** ± 0.02	**99.67** ± 0.01	**76.19** ± 0.34	**88.29** ± 0.07