RECEIVER STATE SHAPES FUNCTIONAL COMPATIBILITY IN MODEL STITCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Model stitching connects a source model to a target receiver through a learned interface and provides a test of functional compatibility. We show that compatibility depends strongly on receiver state: states with nearly identical unperturbed performance can respond very differently to the same perturbation applied to each sample. Controlled token swaps and attention interventions point to the attention messages delivered from patch tokens to the classification token (CLS) as central to this response. The receiver state effect also appears across stitching settings spanning DeiT, ViT, and CLIP. These observations motivate Receiver Internal Message Adaptation (RIMA), which learns residual corrections conditioned on receiver state while keeping the pretrained models, learned stitch, and task head frozen. On the full ImageNet-1K validation set, RIMA improves DeiT-B to DeiT-S stitching from 68.538% to 73.752% with only 0.02M additional trainable parameters and heterogeneous ViT-B to DeiT-S stitching from 57.000% to 69.658%. Across 38 evaluations spanning multiple model families and classification benchmarks, RIMA improves traditional stitching by 40.01 percentage points on average and exceeds interface adaptation with a matched parameter budget by 8.32 points. RIMA also surpasses the native target model in 18 of 21 reported comparisons. The results link functional compatibility to receiver state and support correcting attention messages inside the receiver as an alternative to further interface adaptation.

1 INTRODUCTION

Modern vision models increasingly build on pretrained Transformers learned under diverse paradigms, including supervised transfer at scale, self supervision, and vision and language pretraining (Dosovitskiy et al., 2021; Radford et al., 2021; Oquab et al., 2024). Despite sharing similar Transformer architectures, these models can develop substantially different internal representations. This raises a basic question about compatibility between pretrained models: to what extent can representations produced by one model support computation learned by another?

Recent work in model stitching has improved how representations are mapped across the stitching interface, but these approaches still place the learned interface at the center of adaptation. Successful stitching also does not necessarily imply representational equivalence (Balogh & Jelasity, 2025). In particular, Mai et al. (2026) showed that local feature matching at the stitch position can leave substantial discrepancies after the stitched representation is processed by the frozen target receiver, and improved heterogeneous stitching by training the interface with final feature matching followed by task optimization. This points to a question beyond interface learning: once a stitched representation enters the target network, what role does the current *receiver state* play in determining whether that representation remains functionally compatible with the receiver?

We study this question through controlled interventions inside the frozen receiver. Our central finding is that receiver states with nearly identical unperturbed performance can respond very differently to the same perturbation applied to each sample. In our controlled DeiT-B to DeiT-S setting, two receiver states with accuracies of 88.96% and 89.10% before the added perturbation lose 19.22 and 68.86 percentage points, respectively, under the same perturbation. Controlled token swaps and attention interventions further point to the attention messages delivered from patch tokens to the classification token (CLS) as central to this response. Replacing or repairing patch information de-

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.

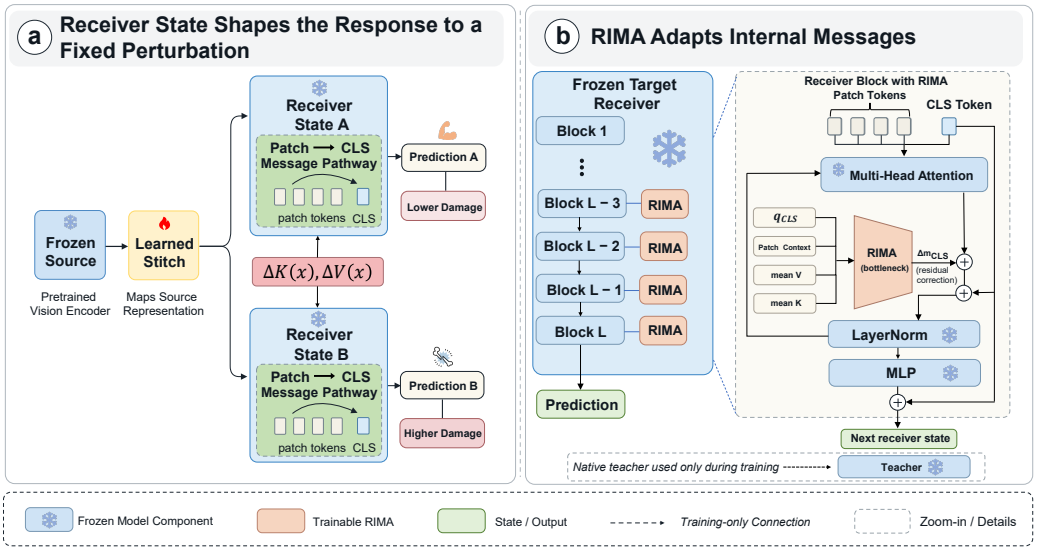

Figure 1: Overview of receiver state effects and Receiver Internal Message Adaptation (RIMA). **(a)** The same perturbation applied to each sample can produce different downstream effects across receiver states. **(b)** RIMA adds lightweight residual corrections conditioned on receiver state to CLS attention outputs inside the frozen receiver. The native target model is used only during training.

livered to CLS recovers substantial performance in both severely degraded and normally functioning stitches. We further reproduce the receiver state effect across multiple stitching settings spanning DeiT, ViT, and CLIP.

Guided by these findings, we introduce **Receiver Internal Message Adaptation (RIMA)**, a lightweight adaptation mechanism that acts inside the receiver rather than further modifying the learned interface. RIMA learns residual corrections conditioned on receiver state while keeping the source model, target receiver, learned stitch, and task head frozen. Figure 1 summarizes the receiver state perspective and the resulting RIMA design. On the full ImageNet-1K validation set, RIMA improves DeiT-B to DeiT-S stitching from 68.538% to 73.752% with only 0.02M additional trainable parameters, and improves heterogeneous ViT-B to DeiT-S stitching from 57.000% to 69.658%. Across 38 evaluations spanning multiple model families and classification benchmarks, RIMA improves traditional stitching by 40.01 percentage points on average and exceeds interface adaptation with a matched parameter budget by 8.32 points on average. Moreover, RIMA also achieves higher accuracy than the native target model in 18 of 21 reported comparisons.

Our key contributions are:

- **Receiver state emerges as an important factor in functional compatibility under model stitching.** Controlled perturbations show that receiver states with nearly identical unperturbed performance can respond very differently to the same perturbation applied to each sample. Token and attention interventions further identify attention messages delivered from patch tokens to CLS as an important internal pathway for this response. The receiver state effect is reproduced across multiple stitching settings spanning DeiT, ViT, and CLIP.
- **Based on these findings, we introduce Receiver Internal Message Adaptation (RIMA).** RIMA learns residual corrections conditioned on receiver state and applies them to CLS attention outputs inside the receiver while keeping the pretrained models, learned stitch, and task head frozen. The main configuration uses only 0.02M additional trainable parameters and requires no native target trajectory at inference time.
- **RIMA improves stitching across model families, datasets, and stitching settings.** On the full ImageNet-1K validation set, RIMA improves the DeiT-B to DeiT-S stitch by 5.214 percentage points and a heterogeneous ViT-B to DeiT-S stitch by 12.658 percentage points.

Across 38 evaluations, RIMA improves traditional stitching by 40.01 percentage points on average and exceeds interface adaptation with a matched parameter budget by 8.32 points on average. Moreover, RIMA achieves higher accuracy than the native target model in 18 of 21 reported comparisons. Experiments with three seeds further show positive RIMA gains in all 18 runs across six confirmation settings.

2 Related Work

Representation Similarity and Functional Compatibility. Neural representations have been compared through feature transfer, SVCCA, projection weighted CCA, and CKA (Yosinski et al., 2014; Raghu et al., 2017; Morcos et al., 2018; Kornblith et al., 2019). Transformation and stitching layers provide a functional test of whether one representation can support another network's computation (Lenc & Vedaldi, 2015; Bansal et al., 2021; Csiszárik et al., 2021; Hernandez et al., 2023). Recent work further shows that functional alignment need not imply representational or informational equivalence (Balogh & Jelasity, 2025; Smith et al., 2025). We build on this distinction by studying how receiver state shapes the response to the same perturbation applied to each sample.

Stitching, Adaptation, and Model Composition. SN-Net uses lightweight stitching layers for flexible deployment (Pan et al., 2023), FuLA uses downstream latent behavior to guide interface learning (Athanasiadis et al., 2026), and final feature matching improves heterogeneous stitching by using representations from the target model together with task optimization (Mai et al., 2026). We instead keep the learned interface fixed and study the receiver through controlled token and attention interventions (Meng et al., 2022; Zhang & Nanda, 2024). RIMA is also related to adapters, LoRA, visual prompt tuning, and AdaptFormer (Houlsby et al., 2019; Hu et al., 2022; Jia et al., 2022; Chen et al., 2022), while broader work on model composition includes weight averaging, permutation alignment, task arithmetic, interference aware merging, and feature merging (Wortsman et al., 2022; Ainsworth et al., 2023; Ilharco et al., 2023; Yadav et al., 2023; Stoica et al., 2024). RIMA differs in that its adaptation site and conditioning are motivated by interventions on receiver state and the attention messages delivered to CLS.

3 Receiver State Shapes Functional Compatibility

Our controlled analysis begins with DeiT-B to DeiT-S stitching on ImageNet-100, with the stitch placed after block 8 and four receiver blocks remaining. This setting allows us to trace how token mismatch evolves through the receiver and to intervene on its internal message flow. Additional directions and model pairs are evaluated with the same receiver state analysis, with full results reported in Appendix B.4.

3.1 Receiver Processing of Token Mismatch

To examine how token mismatch is processed inside the receiver, we compare stitches trained with different token alignment objectives while keeping the model pair, stitching position, receiver, classifier, and interface architecture fixed. The all token objective aligns both CLS and patch tokens at the stitching interface, whereas the CLS only objective aligns only the CLS token. The stitch trained with the all token objective yields 88.60% top-1 accuracy. In contrast, the stitch trained with the CLS only objective yields 2.14% accuracy despite achieving a CLS cosine similarity of 0.9768 with the native target at the stitching interface. The patch tokens remain poorly aligned, with mean cosine similarity 0.0279, and the CLS similarity falls to 0.0493 after receiver processing.

This contrast provides a controlled diagnostic condition in which the CLS component is locally aligned while the patch stream remains strongly mismatched. To trace how this mismatch evolves through the receiver, we evaluate the CLS only condition at increasing receiver depths using the same target final normalization and classifier. At depth 0, with the remaining receiver blocks bypassed, the native and stitched states differ by 13.86 percentage points. The gap rises to 73.88 points after one receiver block and reaches 91.36 points after four blocks, as shown in Figure 2(a). The increasing gap shows that receiver computation substantially changes the downstream consequence of the mismatch. We next ask how information from the patch stream reaches the classification token.

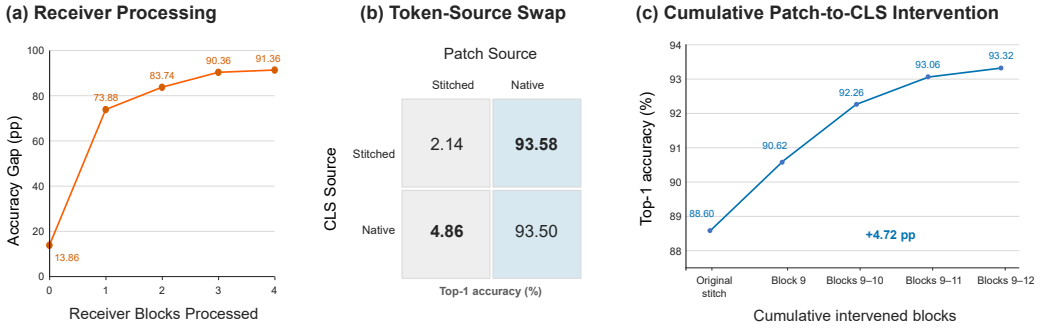

Figure 2: Receiver processing and localization of attention messages delivered from patch tokens to CLS. (a) Under the CLS only diagnostic condition, the gap between native and stitched accuracy increases from 13.86 percentage points at receiver depth 0 to 91.36 points after four receiver blocks. (b) Token-source exchange at the L2 stitching interface, before receiver block 9, shows that combining the stitched CLS with same-image native patches and then processing the result through blocks 9 through 12 raises final accuracy to 93.58%, whereas combining the native CLS with stitched patches yields 4.86%. (c) Cumulative intervention on the patch keys and values used by the CLS query improves the all token stitch from 88.60% to 93.32%.

3.2 Attention Messages from Patch Tokens to CLS Form an Important Pathway

To identify how patch-side mismatch reaches the classification token, we first separate the roles of the CLS and patch streams by exchanging their sources at the L2 stitching interface, immediately after block 8 and before receiver block 9. We construct four combinations from same-image stitched and native tokens and then propagate each combination through the full receiver suffix, blocks 9 through 12. Replacing the stitched patches with same-image native patches while retaining the stitched CLS raises final accuracy from 2.14% to 93.58%, whereas replacing only the stitched CLS with its native counterpart while retaining the stitched patches yields 4.86%. The fully native combination reaches 93.50%, as summarized in Figure 2(b). These results associate the failure exposed by the CLS-only probe primarily with information carried by the patch stream.

We next localize how this patch information reaches CLS. Following standard multihead self attention (Vaswani et al., 2017), let $q_h(s)$ denote the CLS query for attention head h at receiver state s. Let $k_{h,j}(s)$ and $v_{h,j}(s)$ denote the key and value associated with the jth token, where $j \in \mathcal{T}$ and $\mathcal{T}$ contains the CLS token and all patch tokens. The attention message delivered to CLS from head h is

$$m_h(s) = \sum_{j \in \mathcal{T}} \alpha_{h,j}(s) v_{h,j}(s),$$
$$\alpha_{h,j}(s) = \mathrm{softmax}_j\left(\frac{q_h(s)^\top k_{h,j}(s)}{\sqrt{d_h}}\right). \tag{1}$$

Here d_h is the dimension of each attention head, $\alpha_{h,j}(s)$ is the attention weight assigned by the CLS query to the jth token, and $m_h(s)$ is the resulting message delivered to CLS. For a patch token, its key determines how strongly CLS attends to that patch, while its value determines the information contributed by that patch to the CLS message.

To test this pathway directly, we intervene on the patch keys and values used by the CLS query. At each intervened block, we replace only these patch keys and values with native counterparts from the same image. The CLS query, its self key and value, and the patch output rows remain on the current trajectory rather than being replaced by native activations. The resulting trajectory is then processed normally by subsequent receiver blocks. Applying this intervention cumulatively across blocks 9 through 12 raises the CLS only condition from 2.14% to 93.52%. The same intervention improves the normally functioning all token stitch from 88.60% to 93.32%, with the cumulative progression shown in Figure 2(c).

These intervention results identify the patch keys and values used by the CLS query as an important pathway through which patch information affects the CLS. However, the attention computation in Equation 1 also depends on the current receiver state. This raises a different question from pathway localization: can two receiver states with similar unperturbed task performance nevertheless differ substantially in their sensitivity to the same perturbation? We test this possibility next.

3.3 Receiver State Modulates a Fixed Perturbation

To test this question, we first construct receiver states with different internal trajectories and then compare how they respond to the same perturbation. We start from the native target trajectory and choose a subset $P \subseteq \{9, 10, 11\}$ of receiver blocks preceding block 12. At each block in P, we replace only the patch keys and values read by the CLS query with same-image stitched-reference patch keys and values from the frozen stitched trajectory. The current CLS query, residual stream, and subsequent receiver computation remain on the evolving hybrid trajectory rather than being reset to native. When $P = \varnothing$, no replacements are made and the trajectory remains native. These preceding-block interventions construct the receiver state entering block 12. The fixed perturbation tested at block 12 is a separate operation, defined below and held constant across the constructed states for each image.

For each image x, we obtain the patch keys and values at block 12 from the frozen stitched model and the native target model. Their difference defines the perturbation for that image:

$$\begin{aligned} \Delta K(x) &= K_{12}^{C}(x) - K_{12}^{N}(x), \\ \Delta V(x) &= V_{12}^{C}(x) - V_{12}^{N}(x), \end{aligned} \tag{2}$$

where C and N denote the stitched and native reference trajectories, respectively.

We then replay this same perturbation at block 12 across the constructed receiver states:

$$\begin{aligned} K_{12}^{(s,\alpha)}(x) &= K_{12}^{(s,0)}(x) + \alpha \Delta K(x), \\ V_{12}^{(s,\alpha)}(x) &= V_{12}^{(s,0)}(x) + \alpha \Delta V(x), \end{aligned} \tag{3}$$

where α controls perturbation strength. We use $\alpha = 1$ for the main analysis, corresponding to the full perturbation defined above. Results across the full strength sweep, $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$, are reported in Appendix B.3. For each image, the same $\Delta K(x)$ and $\Delta V(x)$ are retained across all evaluated receiver states.

The response differs sharply across states. Consider two states constructed with the same number of preceding interventions. Before the additional perturbation at block 12, the states generated from blocks $\{9, 10\}$ and $\{10, 11\}$ have accuracies of 88.96% and 89.10%, respectively, a difference of only 0.14 percentage points. Applying the same perturbation at $\alpha = 1$ reduces their accuracies to 69.74% and 20.24%, corresponding to losses of 19.22 and 68.86 percentage points. Their perturbation induced damage therefore differs by 49.64 points despite nearly identical starting accuracy, as shown in Figure 3(a). Across all eight receiver states, the loss ranges from 0.38 to 79.36 percentage points, as shown in Figure 3(b). This result goes beyond the pathway localization in Section 3.2: similar unperturbed performance does not imply similar sensitivity to perturbation along that pathway.

A first order expansion of the attention computation clarifies how this dependence on receiver state can arise. Let $\mathcal{P}$ denote the patch token index set, and let

$$\delta = \{\Delta k_{h,j}, \Delta v_{h,j}\}_{j \in \mathcal{P}}$$

denote a perturbation to the patch keys and values. For a given receiver state, the current state is held fixed while considering the local response to this perturbation. The corresponding change in the CLS attention message is approximately

$$\Delta m_h(s; \delta) \approx \sum_{j \in \mathcal{P}} \alpha_{h,j}(s) \left[\Delta v_{h,j} + \frac{q_h(s)^\top \Delta k_{h,j}}{\sqrt{d_h}} \big(v_{h,j}(s) - m_h(s) \big) \right]. \tag{4}$$

Even when the patch key and value perturbation is held fixed, its effect on the CLS attention message can vary because the quantities in Equation 4 are evaluated at the current receiver state. The

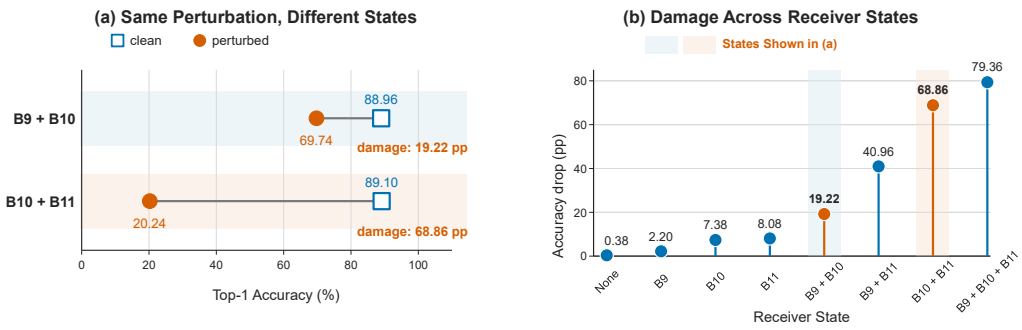

Figure 3: Receiver state modulates the response to the same perturbation. In both panels, None denotes the unmodified native target trajectory; B9, B10, and B11 identify blocks at which same-image stitched-reference patch keys and values are injected into the CLS query, and plus signs combine these blocks along the native-initialized trajectory. (a) Two receiver states with nearly identical accuracies before the perturbation at block 12, 88.96% and 89.10%, exhibit perturbation induced losses of 19.22 and 68.86 percentage points, respectively. (b) Across all eight evaluated receiver states, the same perturbation protocol produces accuracy drops ranging from 0.38 to 79.36 percentage points.

controlled experiment above shows that this dependence is also reflected at the task level, where the same perturbation produces substantially different accuracy drops across receiver states.

We further repeat the fixed perturbation analysis across additional stitching settings spanning DeiT, ViT, and CLIP. Across multiple model pairs, receiver states with comparable unperturbed performance exhibit substantially different responses to the same perturbation. These results reproduce the receiver state effect beyond the original DeiT-B to DeiT-S setting. Full replication results are reported in Appendix B.4.

Together with the attention pathway identified in Section 3.2, these results motivate a correction inside the receiver with two properties: it should depend on the current receiver state and act on the messages delivered to CLS. We introduce this mechanism next as *Receiver Internal Message Adaptation* (RIMA).

4 RIMA: Receiver Internal Message Adaptation

The analysis in Sec. 3 motivates two design choices for adaptation inside the receiver. First, the correction should depend on the current receiver state, since states with similar unperturbed performance can respond very differently to the same perturbation. Second, it should act on the attention messages delivered from patch tokens to CLS, the internal pathway identified in Sec. 3.2. We therefore introduce **Receiver Internal Message Adaptation (RIMA)**, illustrated in Fig. 1(b). RIMA learns residual corrections conditioned on receiver state and applies them to CLS attention outputs while keeping the source model, target receiver, learned stitch, and task head frozen.

4.1 Message Correction Conditioned on Receiver State

Consider receiver block b and attention head h. Following Eq. (1), let $q_{b,h}(s)$ denote the current CLS query, and let $k_{b,h,j}(s)$ and $v_{b,h,j}(s)$ denote the key and value of the jth patch token. RIMA summarizes the patch information available to CLS using an attention context over patch tokens:

$$c_{b,h}(s) = \sum_{j \in \mathcal{P}} \widetilde{\alpha}_{b,h,j}(s) v_{b,h,j}(s),$$
$$\widetilde{\alpha}_{b,h,j}(s) = \mathrm{softmax}_{j \in \mathcal{P}} \left(\frac{q_{b,h}(s)^\top k_{b,h,j}(s)}{\sqrt{d_h}} \right). \tag{5}$$

Unlike the attention weights in Eq. (1), $\widetilde{\alpha}$ is normalized over patch tokens only. We additionally summarize the patch keys and values by

$$\bar{k}_{b,h}(s) = \frac{1}{|\mathcal{P}|} \sum_{j \in \mathcal{P}} k_{b,h,j}(s),$$
$$\bar{v}_{b,h}(s) = \frac{1}{|\mathcal{P}|} \sum_{j \in \mathcal{P}} v_{b,h,j}(s). \tag{6}$$

For each head, we form the state summary

$$u_{b,h}(s) = \left[q_{b,h}(s); c_{b,h}(s); \bar{k}_{b,h}(s); \bar{v}_{b,h}(s)\right] \in \mathbb{R}^{4d_h}. \tag{7}$$

The CLS query and patch summaries provide a compact description of the current receiver computation relevant to the attention messages delivered from patch tokens to CLS.

RIMA maps this state summary to a d_h dimensional correction through a bottleneck of width r:

$$\delta_{b,h}(s) = W_b^{\mathrm{up}} \sigma\left(W_b^{\mathrm{down}} u_{b,h}(s) + \mathbf{b}_b^{\mathrm{down}}\right) + \mathbf{b}_b^{\mathrm{up}}, \tag{8}$$

where σ denotes GELU. The same bottleneck network is applied to each attention head within a block, with parameters shared across heads, while different receiver blocks use independent parameters. We use $r = 16$ throughout. The conditioning features are detached before entering the RIMA module, while the corrected trajectory continues through the frozen receiver computation.

Let $a_b^{\mathrm{cls}}(s)$ denote the projected CLS attention output before the residual addition. The corrections from all attention heads are concatenated and added directly to this output:

$$\widetilde{a}_b^{\mathrm{cls}}(s) = a_b^{\mathrm{cls}}(s) + \mathrm{Concat}_{h=1}^{H}\left[\delta_{b,h}(s)\right]. \tag{9}$$

The corrected CLS output then follows the original residual path and subsequent receiver layers. Unlike the diagnostic interventions in Sec. 3.2, RIMA does not replace patch keys or values with native activations. Each correction is predicted entirely from quantities available on the current stitched trajectory.

In our experiments, RIMA is applied to the last four available receiver blocks. For B adapted blocks, each using bottleneck width r and head dimension d_h, the number of trainable parameters is

$$P_{\mathrm{RIMA}} = B\left(5d_h r + r + d_h\right). \tag{10}$$

With $B = 4$, $d_h = 64$, and $r = 16$, this configuration contains 20,800 trainable parameters. The output layer of each RIMA module is initialized to zero, so the initial model exactly reproduces the underlying stitched model before adaptation.

4.2 Training and Inference

Only the RIMA parameters are optimized during training. The source model, learned stitch, target receiver, and task head remain frozen.

Let $z^R(x)$ denote the logits produced by the stitched model with RIMA, and let $z^N(x)$ denote the logits produced by the frozen native target model. Following the knowledge distillation paradigm (Hinton et al., 2015), we train RIMA with cross-entropy and centered global logit distillation:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \mathcal{L}_{\mathrm{GlobalKD}}. \tag{11}$$

The GlobalKD term is

$$\mathcal{L}_{\mathrm{GlobalKD}} = \frac{1}{NC} \sum_{i=1}^{N} \left\| \mathrm{ctr}\left(z^R(x_i)\right) - \mathrm{ctr}\left(z^N(x_i)\right) \right\|_2^2, \tag{12}$$

where N is the minibatch size, C is the number of classes, and

$$\mathrm{ctr}(z) = z - \frac{1}{C}\left(\mathbf{1}^\top z\right)\mathbf{1} \tag{13}$$

denotes logit centering for each sample.

The native target model provides teacher logits only during training. At inference, RIMA predicts all corrections from the current stitched trajectory and requires neither activations nor logits from the native target model.

Table 1: Top-1 accuracy on the official ImageNet-1K validation set. Confidence intervals compare RIMA with the corresponding stitched baseline.

Source → Receiver	Condition	Stitch	RIMA	Gain (pp)	95% CI
DeiT-B → DeiT-S	main setting	68.538	**73.752**	**+5.214**	$[4.932, 5.496]$
ViT-B → DeiT-S	cut 8, all token	57.000	**69.658**	**+12.658**	$[12.296, 13.016]$
DeiT-S → ViT-B	cut 8, balanced	67.904	**72.840**	**+4.936**	$[4.664, 5.212]$
DeiT-S → ViT-B	cut 6, all token	75.414	**76.962**	**+1.548**	$[1.356, 1.742]$

Table 2: Evaluation with three seeds across model families and classification datasets. Values are mean ± sample standard deviation over three seeds.

Source → Receiver	Dataset	Stitch	RIMA	Gain (pp)
DINOv2-S → DINOv2-B	ImageNet-100	60.56 ± 1.90	$\mathbf{90.47 \pm 0.30}$	$\mathbf{+29.91 \pm 1.78}$
DINOv2-S → DINOv2-B	CIFAR-100	65.39 ± 3.04	$\mathbf{83.28 \pm 0.12}$	$\mathbf{+17.89 \pm 2.95}$
DINOv2-S → DINOv2-B	Pets	79.72 ± 4.54	$\mathbf{95.00 \pm 0.18}$	$\mathbf{+15.28 \pm 4.65}$
ViT-S → ViT-B	CIFAR-100	51.23 ± 0.27	$\mathbf{68.41 \pm 0.20}$	$\mathbf{+17.17 \pm 0.21}$
ViT-B → ViT-S	CIFAR-100	63.46 ± 0.30	$\mathbf{70.67 \pm 0.15}$	$\mathbf{+7.21 \pm 0.35}$
ViT-S → ViT-B	Pets	87.54 ± 0.33	$\mathbf{90.03 \pm 0.16}$	$\mathbf{+2.49 \pm 0.43}$

5 Experiments

We evaluate RIMA on ImageNet-1K, across multiple model families and classification datasets, and against interface adaptation with a matched parameter budget.

5.1 Experimental Setup

Our evaluation spans DeiT, ViT, DINOv2, CLIP, and MAE (Touvron et al., 2021; Dosovitskiy et al., 2021; Oquab et al., 2024; Radford et al., 2021; He et al., 2022) across ImageNet-1K and ImageNet-100 (Deng et al., 2009; Russakovsky et al., 2015), CIFAR-100 (Krizhevsky, 2009), Oxford-IIIT Pets (Parkhi et al., 2012), Flowers102 (Nilsback & Zisserman, 2008), and DTD (Cimpoi et al., 2014). Each experiment starts from a trained affine stitch. During RIMA adaptation, the source model, target receiver, learned stitch, and task head remain frozen. The main ImageNet-1K setting uses 20,000 stitch training steps followed by 5,000 RIMA steps. Heterogeneous ImageNet-1K experiments use 5,000 stitch training steps and 3,000 RIMA steps. These three configurations were fixed after discovery and independent replication, before evaluation on the official validation set of 50,000 images. We report top-1 accuracy, gains in percentage points, means and sample standard deviations over three seeds where applicable, and paired bootstrap confidence intervals. Full protocols are provided in Appendix A.

5.2 ImageNet-1K Evaluation

In the main DeiT-B to DeiT-S setting, RIMA improves ImageNet-1K accuracy from 68.538% to 73.752% (+5.214 pp, 95% CI $[4.932, 5.496]$) with 20,800 trainable parameters. RIMA also improves all three preselected heterogeneous stitching conditions on the official ImageNet-1K validation set, with gains ranging from 1.548 to 12.658 pp. A positive gain also persists for the stronger cut-6 parent stitch, showing that the benefit is not confined to severely degraded interfaces (Table 1).

5.3 Generalization Across Models and Datasets

Across six settings evaluated with three seeds spanning DINOv2 and ViT models, RIMA improves the stitched baseline in all 18 runs, with a positive paired bootstrap 95% confidence interval in every run. Mean gains range from 2.49 ± 0.43 pp for ViT-S to ViT-B on Pets to 29.91 ± 1.78 pp for DINOv2-S to DINOv2-B on ImageNet-100 (Table 2). Additional evaluations with CLIP and MAE model families are provided in Appendix C.3.

Table 3: Comparison between RIMA and interface adaptation with a matched parameter budget on six confirmation conditions. Confidence intervals are for RIMA minus Interface Baseline.

Source → Receiver	Dataset	Stitch	Interface Baseline	RIMA	RIMA−Baseline	95% CI
MAE-B → CLIP-B/16	Flowers102	37.84	78.92	**87.45**	**+8.53**	$[6.18, 10.88]$
ViT-S → DeiT-S	Flowers102	20.49	76.47	**80.39**	**+3.92**	$[1.47, 6.37]$
CLIP-B/16 → DeiT-B	Flowers102	54.71	86.67	**90.49**	**+3.82**	$[2.06, 5.59]$
ViT-B → CLIP-B/16	Flowers102	55.29	81.67	**85.49**	**+3.82**	$[1.47, 6.18]$
CLIP-B/16 → ViT-B	Flowers102	25.69	82.45	**85.98**	**+3.53**	$[1.47, 5.59]$
DeiT-B → CLIP-B/16	Flowers102	57.94	86.86	**90.29**	**+3.43**	$[1.47, 5.39]$

Table 4: Representative cases in which RIMA achieves higher accuracy than the native target model.

Source → Receiver	Dataset	Native	RIMA	RIMA−Native (pp)
DeiT-B → MAE-B	Flowers102	34.71	**75.49**	**+40.78**
CLIP-B/16 → ViT-B	DTD	63.03	**71.28**	**+8.25**

5.4 Interface Adaptation with a Matched Parameter Budget

We compare RIMA with an interface baseline that adapts part of the original affine stitch under the same data, supervision, and optimization schedule, with an approximately matched parameter budget. RIMA contains 20,800 trainable parameters, while the interface baseline uses 20,763 to 20,790 trainable parameters across the evaluated interface dimensions. Across six confirmation conditions, RIMA exceeds the interface baseline by 3.43 to 8.53 pp, with a positive paired 95% confidence interval in every case (Table 3). Across the complete 38-evaluation suite, RIMA improves traditional stitching by 40.01 pp on average and exceeds the interface baseline by 8.32 pp on average. Full results are provided in Appendix C.3.

6 Practical Implications and Discussion

6.1 Beyond Native Target Performance

We further compare RIMA with the frozen native target model in the 21 conditions tabulated in Appendix C.4. RIMA achieves higher accuracy than the native target model in 18 of these 21 comparisons, with such cases appearing across all four datasets. For MAE-B receivers, RIMA outperforms the native model in 11 of 12 comparisons, with a mean gain of 16.18 percentage points. This may reflect unused task-relevant information in the native MAE trajectory. Representative cases are shown in Table 4. A smaller gain over the native target also appears on the official ImageNet-1K validation set, where DeiT-S to ViT-B reaches 76.962% with RIMA compared with 76.786% for the frozen native target model.

6.2 Discussion

Our experiments show that receiver state affects stitching and that attention interventions localize the relevant mismatch to messages from patch tokens to CLS. RIMA uses this receiver-side pathway and outperforms matched-budget interface adaptation. It currently requires native target logits during training and targets Transformer receivers with a CLS token.

7 Conclusion

We found that receiver state affects how a stitched representation is processed: two states with similar unperturbed accuracy can react very differently to the same patch-side perturbation. Interventions place the relevant mismatch in the messages from patch tokens to CLS, which motivates RIMA's state-conditioned residual correction inside the receiver. With this small adaptation module, stitching improves across the evaluated model families and datasets, and the matched-budget comparison favors receiver-side adaptation over interface updates.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and polishing of the manuscript, including improving clarity, grammar, and readability. They were also used for retrieval and discovery, such as identifying potentially relevant literature and related work. All AI-assisted writing was reviewed and edited by the authors, and all references identified with AI assistance were manually verified against the original sources. The authors take full responsibility for the final content of the paper, including its text, claims, citations, and conclusions.

REFERENCES

Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git Re-Basin: Merging Models Modulo Permutation Symmetries. In *International Conference on Learning Representations*, 2023. URL https://mlanthology.org/iclr/2023/ainsworth2023iclr-git/.

Ioannis Athanasiadis, Anmar Karmush, and Michael Felsberg. Grounding functional similarity by invariance-aware model stitching, 2026. URL https://arxiv.org/abs/2505.20142.

András Balogh and Márk Jelasity. How not to stitch representations to measure similarity: Task loss matching versus direct matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15472–15480, 2025.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *Advances in neural information processing systems*, 34:225–236, 2021.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in neural information processing systems*, 35:16664–16678, 2022.

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.

Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 5656–5668. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/2cb274e6ce940f47beb8011d8ecb1462-Paper.pdf.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. URL https://mlanthology.org/iclr/2021/dosovitskiy2021iclr-image/.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15979–15988. IEEE, 2022.

Adriano Hernandez, Rumen Dangovski, Peter Y Lu, and Marin Soljacic. Model stitching: Looking for functional similarity between representations. *arXiv preprint arXiv:2303.11277*, 2023.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

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

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing Models with Task Arithmetic. In *International Conference on Learning Representations*, 2023. URL https://mlanthology.org/iclr/2023/ilharco2023iclr-editing/.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 991–999, 2015.

Zheda Mai, Ke Zhang, Fu-En Wang, Zixiao Ken Wang, Albert Y. C. Chen, Lu Xia, Min Sun, Wei-Lun Chao, and Cheng-Hao Kuo. Revisiting model stitching in the foundation model era. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41342–41351, June 2026.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in neural information processing systems*, 31, 2018.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features Without Supervision. *Transactions on Machine Learning Research*, 2024. URL https://mlanthology.org/tmlr/2024/oquab2024tmlr-dinov2/.

Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Stitchable neural networks. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16102–16112. IEEE, 2023.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

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.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Damian Smith, Harvey Mannering, and Antonia Marcu. Functional alignment can mislead: Examining model stitching. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55972–55998. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/smith25a.html.

George Stoica, Daniel Bolya, Jakob Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. Zipit! merging models from different tasks without training. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 29215–29237, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/7dbb5bfab324e3b86af9bd0df15498dd-Paper-Conference.pdf.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

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

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 1651–1678, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/06a52a54c8ee03cd86771136bc91eb1f-Paper-Conference.pdf.

A Reproducibility and Experimental Details

Organization. This appendix is organized around the main claims of the paper. Appendix A provides implementation and evaluation details needed for reproduction. Appendix B expands the receiver state analysis in Sec. 3, including sensitivity to the stitching objective, the complete perturbation strength sweep, and replication across model pairs. Appendix C provides additional confirmatory evidence for RIMA, including independent ImageNet-1K confirmation, seed level results, the complete matched budget comparison, and native target comparisons.

A.1 RIMA Implementation

For each adapted receiver block, RIMA forms one feature vector per attention head from the current stitched trajectory. The feature concatenates the current CLS query, an attention context normalized over patch tokens only, the mean patch key, and the mean patch value. The conditioning features are detached before entering RIMA. Within a receiver block, the same bottleneck network is shared across attention heads, while different receiver blocks use independent bottlenecks. Each bottleneck is $\mathrm{Linear}(4d_h, r)$, GELU, followed by $\mathrm{Linear}(r, d_h)$ with $r = 16$. The resulting head corrections are concatenated and added to the projected CLS attention output after the attention output projection and dropout and before the original layer scale, drop path, and residual addition. The output projection of each RIMA module is initialized to zero, so the initial model exactly reproduces the underlying stitched model.

RIMA is applied to the last four available receiver blocks in the main configuration. With four adapted blocks, head dimension $d_h = 64$, and bottleneck width $r = 16$, the number of trainable parameters is 20,800. All pretrained model weights, the learned stitch, and the task head remain frozen during RIMA training.

A.2 Training and Statistical Protocol

RIMA is trained with cross entropy and centered logit matching to the frozen native target model, with unit weight on the distillation term. The native target is used only as a training teacher. At inference, RIMA predicts its corrections entirely from the stitched trajectory and uses neither native activations nor native logits.

The main DeiT-B to DeiT-S ImageNet-1K experiment uses a stitch trained for 20,000 steps followed by 5,000 RIMA steps. The three cross family ImageNet-1K confirmation conditions reported here use 5,000 stitch training steps and 3,000 RIMA steps. The repeated same family experiments use 20,000 stitch training steps and 5,000 RIMA steps. Their final evaluation sizes are 3,000 images for ImageNet-100, 10,000 images for CIFAR-100, and 3,669 images for Oxford-IIIT Pets.

In the repeated same family and matched budget experiments, RIMA is optimized with AdamW using learning rate 10^{-3} and zero weight decay. The repeated same family experiments use batch size 64. The matched budget experiments use effective batch size 128, implemented with microbatches of 64 and gradient accumulation of two, without a learning rate scheduler. Reported results use the fixed final checkpoint rather than selecting the best validation checkpoint.

Where paired confidence intervals are reported, the bootstrap resamples predictions on the same evaluation images for the two compared methods. Seed variation is reported separately using the sample standard deviation over the three independent runs.

A.3 Interface Baseline with a Matched Parameter Budget

The interface baseline adapts a fixed subset of output rows of the original affine stitch while leaving the remaining interface parameters frozen. The number of trainable rows is chosen according to the stitch input dimension so that the resulting parameter count closely matches the 20,800 trainable parameters of RIMA. Each trainable output row contains all weights associated with that row together with its bias entry. For a stitch with input dimension 768, 27 trainable rows give $27 \times (768 + 1) = 20{,}763$ parameters. For the ViT-S to DeiT-S condition with input dimension 384, 54 trainable rows give $54 \times (384 + 1) = 20{,}790$ parameters. The selected rows are fixed independently of performance. Within each matched budget condition, RIMA and the interface baseline use

Table 5: Sensitivity of stitched accuracy to the alignment objective. Values are top-1 accuracy in percent under the original target classifier. Balanced denotes the CLS and patch balanced objective.

Direction	Cut	All token	CLS only	Balanced
DeiT-B to DeiT-S	L2	88.60	2.14	88.66
DeiT-B to DeiT-S	L3	2.16	94.18	93.60
DeiT-S to DeiT-B	L2	91.56	3.80	91.60
DeiT-S to DeiT-B	L3	2.90	93.42	91.96

the same training data, labels, native target supervision, optimization schedule, and evaluation set. The comparison therefore changes the adaptation site while keeping the trainable parameter budget and training protocol closely matched.

B Additional Evidence for Receiver State

B.1 Sensitivity to the Stitching Objective

The main text focuses on the L2 contrast between the all token and CLS only objectives because it provides a controlled diagnostic setting in which the stitched CLS is strongly aligned at the interface while the patch stream remains mismatched. Table 5 compares three alignment objectives across both stitching directions and cuts. For each direction and cut, the source model, receiver, classifier, interface parameterization, training budget, and evaluation set are fixed; only the objective used to train the stitch changes. L2 denotes a stitch after block 8, leaving blocks 9 through 12 for subsequent receiver processing. L3 denotes a stitch after block 12, so no Transformer blocks remain after the stitching interface; only the final normalization and classifier are applied. All accuracies are measured with the original target classifier.

The all token objective aligns CLS and patch tokens jointly at the stitching interface. The CLS only objective aligns only the CLS token. The balanced objective assigns separate weight to the CLS and patch terms.

The dependence on the training objective is substantial and changes with the stitching cut. At L2, where four receiver blocks remain after the interface, CLS only yields 2.14% and 3.80% accuracy in the two directions, while all token and balanced matching remain above 88%. At L3, where no Transformer blocks remain after the interface, the pattern reverses: all token yields 2.16% and 2.90%, whereas CLS only and balanced matching yield between 91.96% and 94.18%. Because each objective produces a separately trained stitch, these results do not compare one fixed stitched representation under different evaluations. Instead, they show that the functional outcome of stitching depends strongly on both the alignment objective and where the interface is placed. We therefore use the L2 CLS only stitch as a diagnostic condition, where the mismatched representation must still pass through four receiver blocks, and then isolate the downstream message pathway and receiver state effects with the interventions that follow.

B.2 Additional Evidence for the CLS Message Pathway

Section 3.2 uses interventions on the patch keys and values consumed by the CLS query to localize an important receiver pathway. Here we report the full blockwise intervention results for the normally functioning all token stitch. At an intervened block, only the patch keys and values used by the CLS query are replaced by same image native references. The current CLS query, its self key and value, and the patch output rows remain on the stitched trajectory. These interventions are diagnostic and are not used by RIMA at inference. The complete blockwise results are reported in Table 6.

The cumulative intervention raises the all token stitch from 88.60% to 93.32%. Together with the severe CLS only case in the main text, this shows that the pathway identified by the intervention is not limited to a nearly failed stitched model. The result identifies patch information delivered to CLS as an important recoverable pathway, but it does not imply that patch keys and values are the unique cause of stitching failures.

Table 6: Blockwise intervention on the patch keys and values used by the CLS query in the DeiT-B to DeiT-S all token stitch. The parent stitch has 88.60% accuracy. Cumulative interventions retain the replacement at all preceding receiver blocks.

Intervention	Accuracy (%)	Gain (pp)
Parent stitch	88.60	0.00
Single block 9	90.62	+2.02
Single block 10	91.52	+2.92
Single block 11	90.82	+2.22
Single block 12	90.22	+1.62
Cumulative through block 9	90.62	+2.02
Cumulative through block 10	92.26	+3.66
Cumulative through block 11	93.06	+4.46
Cumulative through block 12	93.32	+4.72

Table 7: Perturbation sensitivity for DeiT-B to DeiT-S. State label P identifies the subset of blocks 9–11 at which same-image stitched-reference patch keys and values read by the CLS query are injected into a native-initialized trajectory; $P = \varnothing$ is the unmodified native trajectory. $A(0)$ is accuracy before the additional block 12 perturbation. The remaining columns report damage in percentage points.

State	$A(0)$	$D(.25)$	$D(.50)$	$D(.75)$	$D(1)$
$\varnothing$	93.50	-0.06	0.00	0.08	0.38
$\{9\}$	91.02	0.50	1.30	1.72	2.20
$\{10\}$	91.84	0.60	1.62	3.92	7.38
$\{11\}$	92.18	0.82	2.44	4.94	8.08
$\{9, 10\}$	88.96	1.78	5.52	10.58	19.22
$\{9, 11\}$	87.96	3.78	12.40	26.80	40.96
$\{10, 11\}$	89.10	3.58	16.66	44.80	68.86
$\{9, 10, 11\}$	79.82	18.46	59.54	77.88	79.36

B.3 Sensitivity to Perturbation Strength

The main text reports the full perturbation at $\alpha = 1$. We additionally evaluate the complete strength grid $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ for both DeiT-B to DeiT-S and DeiT-S to DeiT-B stitching. Receiver states are first constructed from the native target trajectory by injecting same-image stitched-reference patch keys and values read by the CLS query at selected subsets of blocks 9, 10, and 11; the empty subset is therefore the unmodified native trajectory. The perturbation tested at block 12 is then defined separately and held fixed across states. Thus, state construction and state probing are distinct parts of the experiment.

For a receiver state s, let $A_s(\alpha)$ denote top-1 accuracy after applying perturbation strength α. We report the perturbation induced damage

$$D_s(\alpha) = A_s(0) - A_s(\alpha), \tag{14}$$

where larger values indicate greater sensitivity. The complete experiment contains two directions, eight receiver states, and five perturbation strengths, for 80 completed conditions. Each condition is evaluated on the same 5,000 ImageNet-100 validation images with seed 20260902. Tables 7 and 8 report the complete strength sweeps for the two stitching directions.

The complete strength sweep makes the focal comparison in Sec. 3.3 more explicit. In DeiT-B to DeiT-S, states $\{9, 10\}$ and $\{10, 11\}$ begin only 0.14 percentage points apart at $\alpha = 0$, but their damage gap grows from 1.80 points at $\alpha = 0.25$ to 11.14 at $\alpha = 0.5$, 34.22 at $\alpha = 0.75$, and 49.64 at $\alpha = 1$. The reverse direction shows the same qualitative separation: the two states begin 0.98 points apart and their damage gap reaches 52.86 points at $\alpha = 1$. Across the complete grid, the empty and single block states remain substantially less sensitive than the most fragile multi block states. The small negative damage of -0.06 points for the empty DeiT-B to DeiT-S state at $\alpha = 0.25$ is the only visible departure from monotonic damage and corresponds to an accuracy change from 93.50% to 93.56%.

Table 8: Perturbation sensitivity for DeiT-S to DeiT-B. State label P identifies the subset of blocks 9–11 at which same-image stitched-reference patch keys and values read by the CLS query are injected into a native-initialized trajectory; $P = \varnothing$ is the unmodified native trajectory. $A(0)$ is accuracy before the additional block 12 perturbation. The remaining columns report damage in percentage points.

State	$A(0)$	$D(.25)$	$D(.50)$	$D(.75)$	$D(1)$
$\varnothing$	94.50	0.10	0.90	1.60	2.64
$\{9\}$	94.36	0.22	0.92	1.98	4.68
$\{10\}$	94.00	0.28	0.92	2.42	5.72
$\{11\}$	93.68	0.96	2.82	7.32	20.10
$\{9, 10\}$	93.94	0.32	1.54	3.88	11.34
$\{9, 11\}$	93.60	1.28	4.60	15.00	52.58
$\{10, 11\}$	92.96	1.80	6.68	27.60	64.20
$\{9, 10, 11\}$	92.56	2.06	9.98	48.38	87.12

Table 9: Replication of receiver state sensitivity across model pairs. State A is $\{9, 10\}$ and State B is $\{10, 11\}$. Damage and confidence intervals are in percentage points. The first row repeats the original DeiT-B to DeiT-S pair under the replication protocol.

Source → Receiver	$A(0)$	$B(0)$	$D_A(1)$	$D_B(1)$	Gap	95% CI	n
DeiT-B → DeiT-S	89.15	88.15	19.15	67.50	+48.35	$[45.70, 50.95]$	2000
ViT-S → DeiT-B	94.10	93.35	8.05	61.85	+53.80	$[51.45, 56.10]$	2000
ViT-B → DeiT-B	94.05	93.05	7.85	56.50	+48.65	$[46.30, 51.05]$	2000
CLIP-B/16 → ViT-B	91.95	90.75	19.30	56.15	+36.85	$[34.45, 39.30]$	2000
DeiT-B → ViT-B	92.40	91.25	21.60	48.20	+26.60	$[24.45, 28.85]$	2000
DeiT-S → ViT-B	91.65	90.45	38.00	61.95	+23.95	$[21.45, 26.40]$	2000
CLIP-B/16 → DeiT-B	94.50	93.05	8.60	74.35	+65.75	$[63.50, 67.95]$	2000
ViT-B → DeiT-S	90.90	89.10	7.80	50.60	+42.80	$[40.40, 45.25]$	2000
CLIP-B/16 → DeiT-S	89.55	84.65	15.25	64.45	+49.20	$[46.50, 51.85]$	2000

For the two focal states, we also perform paired bootstrap analysis using the existing predictions, with 10,000 resamples over the 5,000 shared images. For DeiT-B to DeiT-S at $\alpha = 1$, the accuracy change relative to $\alpha = 0$ is -19.22 points with 95% CI $[-20.38, -18.10]$ for state $\{9, 10\}$ and -68.86 points with CI $[-70.14, -67.58]$ for state $\{10, 11\}$. In the reverse direction, the corresponding changes are -11.34 points with CI $[-12.28, -10.42]$ and -64.20 points with CI $[-65.56, -62.82]$. These intervals quantify evaluation sample uncertainty for the fixed Stage 3G run.

B.4 Replication Across Model Pairs

We next ask whether the receiver state effect is specific to the original DeiT-B to DeiT-S pair. We repeat the same fixed perturbation analysis across additional model pairs spanning DeiT, ViT, and CLIP. To match the focal analysis, State A denotes $\{9, 10\}$ and State B denotes $\{10, 11\}$. We first repeat the original DeiT-B to DeiT-S pair under the same 2,000 image screening protocol as a reference condition, then evaluate eight additional model pairs. The damage gap is $D_B(1) - D_A(1)$. Table 9 reports the replication results under this screening protocol.

All eight additional model pairs show a positive damage gap with a paired 95% confidence interval excluding zero. State A and State B also start from closely matched accuracy in most conditions: the clean accuracy gap is at most 1.8 points in seven of the eight additional pairs. The CLIP-B/16 to DeiT-S condition has a larger clean gap of 4.9 points and is therefore a weaker matched starting point, although its damage gap remains large. These results reproduce the receiver state effect beyond the original DeiT pair and across both within family and cross family stitching settings.

Two additional conditions were then evaluated on 3,000 images to test whether the screening result persists with a larger evaluation set. Although the two receiver states remain closely matched before perturbation, the difference in accuracy loss under the same fixed perturbation remains 53.97 percentage points for ViT-S to DeiT-B, with 95% CI $[52.10, 55.77]$, and 47.43 points for ViT-B to DeiT-B, with CI $[45.57, 49.33]$. The larger sample results are summarized in Table 10.

Table 10: Larger sample confirmation of two receiver state replication conditions.

Source → Receiver	Clean gap	Damage gap	95% CI	n
ViT-S → DeiT-B	0.53	53.97	$[52.10, 55.77]$	3000
ViT-B → DeiT-B	0.30	47.43	$[45.57, 49.33]$	3000

Table 11: Independent confirmation of three cross family ImageNet-1K conditions before evaluation on the official 50,000 image validation set. Confidence intervals compare RIMA with the corresponding stitched baseline.

Source → Receiver	Condition	Stitch	RIMA	Gain	95% CI
ViT-B → DeiT-S	cut 8, all token	62.98	77.12	+14.14	$[13.00, 15.32]$
DeiT-S → ViT-B	cut 8, balanced	77.10	81.56	+4.46	$[3.62, 5.28]$
DeiT-S → ViT-B	cut 6, all token	86.56	87.92	+1.36	$[0.78, 1.96]$

C Additional Evidence for RIMA

C.1 Independent Confirmation Before Official ImageNet-1K Evaluation

We report independent confirmation for three cross family ImageNet-1K conditions from Table 1. These configurations were fixed before the official 50,000 image validation evaluation. Before using the official validation set, we evaluated them on an independent 5,000 image confirmation split. The stitch was trained for 5,000 steps and RIMA was newly initialized and trained for 3,000 steps. All three conditions remain positive on this independent confirmation split. Table 11 reports the corresponding confirmation results.

The independent confirmation and the subsequent official validation therefore agree on the sign of the RIMA improvement in all three reported conditions, including the stronger cut 6 stitched baseline.

C.2 Seed Level Results for Repeated Same Family Experiments

Table 12 reports all 18 runs underlying the means in Table 2. The three base seeds are 20260819, 20260820, and 20260821. Every run has a positive RIMA gain, and every paired 95% confidence interval has a positive lower bound. The ImageNet-100 final evaluation contains 3,000 images, CIFAR-100 uses the 10,000 image test set, and Oxford-IIIT Pets uses 3,669 test images.

The repeated experiments therefore support the aggregate pattern in Table 2 at the run level: the gain remains positive across all six settings, both ViT directions, all three datasets, and all three seeds.

C.3 Complete Matched Budget Evaluation

Tables 13 and 14 report all 38 conditions used for the aggregate matched budget comparison in Sec. 5.4. Across the 38 conditions, the mean RIMA gain over traditional stitching is 40.01 percentage points, the mean interface baseline gain is 31.69 points, and the mean RIMA advantage over the interface baseline is 8.32 points. The paired 95% confidence interval for RIMA minus the interface baseline has a positive lower bound in every condition.

Table 12: Seed level results for the six repeated same family settings. Confidence intervals compare RIMA with the stitched baseline within each run.

Source → Receiver	Dataset	Seed	Stitch	RIMA	Gain	95% CI
DINOv2-S → DINOv2-B	ImageNet-100	20260819	59.07	90.13	+31.07	[29.33, 32.83]
DINOv2-S → DINOv2-B	ImageNet-100	20260820	59.90	90.70	+30.80	[29.07, 32.50]
DINOv2-S → DINOv2-B	ImageNet-100	20260821	62.70	90.57	+27.87	[26.17, 29.57]
DINOv2-S → DINOv2-B	CIFAR-100	20260819	62.44	83.14	+20.70	[19.79, 21.61]
DINOv2-S → DINOv2-B	CIFAR-100	20260820	65.22	83.37	+18.15	[17.28, 19.02]
DINOv2-S → DINOv2-B	CIFAR-100	20260821	68.51	83.33	+14.82	[14.02, 15.64]
DINOv2-S → DINOv2-B	Pets	20260819	84.79	94.93	+10.14	[9.08, 11.23]
DINOv2-S → DINOv2-B	Pets	20260820	78.36	94.88	+16.52	[15.26, 17.83]
DINOv2-S → DINOv2-B	Pets	20260821	76.02	95.20	+19.19	[17.85, 20.52]
ViT-S → ViT-B	CIFAR-100	20260819	51.28	68.60	+17.32	[16.42, 18.23]
ViT-S → ViT-B	CIFAR-100	20260820	51.48	68.41	+16.93	[16.02, 17.82]
ViT-S → ViT-B	CIFAR-100	20260821	50.94	68.21	+17.27	[16.37, 18.18]
ViT-B → ViT-S	CIFAR-100	20260819	63.12	70.72	+7.60	[6.84, 8.35]
ViT-B → ViT-S	CIFAR-100	20260820	63.69	70.79	+7.10	[6.36, 7.83]
ViT-B → ViT-S	CIFAR-100	20260821	63.58	70.50	+6.92	[6.18, 7.65]
ViT-S → ViT-B	Pets	20260819	87.35	89.94	+2.59	[1.80, 3.38]
ViT-S → ViT-B	Pets	20260820	87.93	89.94	+2.02	[1.28, 2.78]
ViT-S → ViT-B	Pets	20260821	87.35	90.22	+2.86	[2.07, 3.65]

Table 13: Complete matched budget results on CIFAR-100 and DTD. The confidence interval is for RIMA minus the interface baseline.

Dataset	Source → Receiver	Stitch	Interface	RIMA	RIMA minus Interface	95% CI
CIFAR-100	CLIP-B/16 → DeiT-B	60.70	74.80	77.15	+2.35	[0.85, 3.80]
CIFAR-100	CLIP-B/16 → MAE-B	6.95	41.40	53.35	+11.95	[9.85, 14.10]
CIFAR-100	CLIP-B/16 → ViT-B	24.00	67.55	72.95	+5.40	[3.65, 7.10]
CIFAR-100	DeiT-B → CLIP-B/16	21.50	61.40	70.00	+8.60	[6.55, 10.65]
CIFAR-100	DeiT-B → MAE-B	14.15	49.45	55.25	+5.80	[3.80, 7.75]
CIFAR-100	MAE-B → CLIP-B/16	7.30	44.60	54.90	+10.30	[8.05, 12.60]
CIFAR-100	MAE-B → DeiT-B	52.90	68.05	70.75	+2.70	[1.00, 4.40]
CIFAR-100	MAE-B → ViT-B	26.75	54.30	61.15	+6.85	[4.85, 8.80]
CIFAR-100	ViT-B → CLIP-B/16	22.25	58.60	68.25	+9.65	[7.55, 11.70]
CIFAR-100	ViT-B → MAE-B	9.85	49.00	54.00	+5.00	[3.00, 7.00]
DTD	CLIP-B/16 → DeiT-B	45.64	69.73	71.97	+2.23	[0.64, 3.88]
DTD	CLIP-B/16 → MAE-B	2.23	54.15	61.76	+7.61	[5.43, 9.79]
DTD	CLIP-B/16 → ViT-B	39.15	69.52	71.28	+1.76	[0.11, 3.46]
DTD	DeiT-B → CLIP-B/16	24.63	60.64	65.74	+5.11	[2.98, 7.18]
DTD	DeiT-B → MAE-B	28.14	55.27	61.97	+6.70	[4.68, 8.72]
DTD	MAE-B → CLIP-B/16	16.91	59.04	64.31	+5.27	[3.19, 7.34]
DTD	MAE-B → DeiT-B	52.45	64.89	66.70	+1.81	[0.11, 3.51]
DTD	MAE-B → ViT-B	48.51	62.87	65.05	+2.18	[0.48, 3.88]
DTD	ViT-B → CLIP-B/16	21.01	55.80	64.04	+8.24	[6.06, 10.48]
DTD	ViT-B → MAE-B	27.13	53.83	58.72	+4.89	[2.82, 6.97]
DTD	ViT-B → ViT-S	55.05	60.32	62.87	+2.55	[0.74, 4.36]

Table 14: Complete matched budget results on Flowers102 and ImageNet-100. The confidence interval is for RIMA minus the interface baseline.

Dataset	Source → Receiver	Stitch	Interface	RIMA	RIMA minus Interface	95% CI
Flowers102	CLIP-B/16 → DeiT-B	54.71	86.67	90.49	+3.82	[2.06, 5.59]
Flowers102	CLIP-B/16 → MAE-B	3.82	30.20	68.43	+38.24	[35.00, 41.57]
Flowers102	CLIP-B/16 → ViT-B	25.69	82.45	85.98	+3.53	[1.47, 5.59]
Flowers102	DeiT-B → CLIP-B/16	57.94	86.86	90.29	+3.43	[1.47, 5.39]
Flowers102	DeiT-B → MAE-B	15.39	35.20	75.49	+40.29	[37.06, 43.63]
Flowers102	MAE-B → CLIP-B/16	37.84	78.92	87.45	+8.53	[6.18, 10.88]
Flowers102	MAE-B → DeiT-B	67.65	86.37	88.53	+2.16	[0.20, 4.12]
Flowers102	MAE-B → ViT-B	35.49	81.08	86.57	+5.49	[3.24, 7.75]
Flowers102	ViT-B → CLIP-B/16	55.29	81.67	85.49	+3.82	[1.47, 6.18]
Flowers102	ViT-B → MAE-B	17.84	40.29	71.57	+31.27	[28.04, 34.61]
Flowers102	ViT-S → DeiT-S	20.49	76.47	80.39	+3.92	[1.47, 6.37]
ImageNet-100	CLIP-B/16 → MAE-B	1.34	41.90	68.32	+26.42	[24.94, 27.90]
ImageNet-100	DeiT-B → CLIP-B/16	47.20	76.26	79.76	+3.50	[2.42, 4.58]
ImageNet-100	DeiT-B → MAE-B	41.06	70.44	75.92	+5.48	[4.42, 6.54]
ImageNet-100	MAE-B → CLIP-B/16	34.34	69.78	76.86	+7.08	[5.90, 8.30]
ImageNet-100	ViT-B → CLIP-B/16	42.48	71.42	77.00	+5.58	[4.38, 6.80]
ImageNet-100	ViT-B → MAE-B	26.24	65.12	71.60	+6.48	[5.32, 7.60]

At the dataset level, the mean RIMA advantage over the interface baseline is 6.86 points on CIFAR-100, 4.40 on DTD, 13.14 on Flowers102, and 9.09 on ImageNet-100. The smallest observed advantage is 1.76 points and the largest is 40.29 points. Thus, the aggregate 8.32 point advantage is not driven by a single dataset. The comparison remains specific to the matched parameterization used here: the interface baseline and RIMA differ in both adaptation location and parameterization.

C.4 Native Target Comparisons

The matched budget suite also permits a direct comparison between RIMA and the corresponding frozen native target model. Tables 15, 16, and 17 report native target comparisons for 21 matched budget conditions. RIMA exceeds the native target in 18 of these 21 comparisons, with such cases appearing on all four datasets: 3 of 4 conditions on CIFAR-100, 6 of 7 on DTD, 7 of 7 on Flowers102, and 2 of 3 on ImageNet-100.

Table 15: Native target comparisons on CIFAR-100. Difference is RIMA minus the corresponding frozen native target accuracy in percentage points.

Source → Receiver	Native	RIMA	Difference
CLIP-B/16 → MAE-B	45.50	53.35	+7.85
CLIP-B/16 → ViT-B	73.15	72.95	-0.20
DeiT-B → MAE-B	45.50	55.25	+9.75
ViT-B → MAE-B	45.50	54.00	+8.50

Table 16: Native target comparisons on DTD and ImageNet-100. Difference is RIMA minus the corresponding frozen native target accuracy in percentage points.

Source → Receiver	Native	RIMA	Difference
DTD			
CLIP-B/16 → DeiT-B	68.24	71.97	+3.73
CLIP-B/16 → MAE-B	43.67	61.76	+18.09
CLIP-B/16 → ViT-B	63.03	71.28	+8.25
DeiT-B → MAE-B	43.67	61.97	+18.30
MAE-B → ViT-B	63.03	65.05	+2.02
ViT-B → MAE-B	43.67	58.72	+15.05
ViT-B → ViT-S	63.09	62.87	-0.22
ImageNet-100			
CLIP-B/16 → MAE-B	70.18	68.32	-1.86
DeiT-B → MAE-B	70.18	75.92	+5.74
ViT-B → MAE-B	70.18	71.60	+1.42

Table 17: Native target comparisons on Flowers102. Difference is RIMA minus the corresponding frozen native target accuracy in percentage points.

Source → Receiver	Native	RIMA	Difference
CLIP-B/16 → DeiT-B	87.94	90.49	+2.55
CLIP-B/16 → MAE-B	34.71	68.43	+33.72
CLIP-B/16 → ViT-B	82.55	85.98	+3.43
DeiT-B → MAE-B	34.71	75.49	+40.78
MAE-B → DeiT-B	87.94	88.53	+0.59
MAE-B → ViT-B	82.55	86.57	+4.02
ViT-B → MAE-B	34.71	71.57	+36.86

The strongest concentration of results above the native target occurs when MAE-B is the receiver. RIMA exceeds the native MAE-B model in 11 of the 12 MAE-B receiver comparisons tabulated here, with a mean RIMA minus native margin of +16.18 percentage points across all 12 comparisons. The largest margins occur on Flowers102 and DTD. This receiver-specific concentration motivates the discussion in Sec. 6.1: the amount of headroom available to receiver-internal adaptation can vary substantially with receiver pretraining. A smaller result above the native target also appears outside this suite on the official ImageNet-1K validation set, where DeiT-S to ViT-B at cut 6 reaches 76.962% with RIMA compared with 76.786% for the frozen native target model (Table 1).