Localizing and Repairing Sparse-Prompt Failure in SAM Decoders via Box-to-Point Counterfactual

Xiaobing Yu[1,2], **Peijie Qiu**[2]*, **Jin Yang**[3], **Zhaoqi An**[2], **Weiwei Ma**[2], **Xiao Wu**[4], **Xuanzhao Dong**[5], **Wenhui Zhu**[5], **Xiwen Chen**[6], **Xiaoqi Zhao**[1,7], **Xiaofeng Liu**[1]†

[1]Yale University [2]Washington University in St. Louis
[3]Icahn School of Medicine at Mount Sinai
[4]University of Electronic Science and Technology of China
[5]Arizona State University [6]Clemson University [7]Nanyang Technological University
xiaobing.yu@yale.edu, xiaofeng.liu@yale.edu

Abstract

Promptable segmentation foundation models such as SAM and its variants achieve strong performance with informative prompts, yet exhibit a striking prompt asymmetry where dense prompts like bounding boxes yield accurate segmentation, whereas sparse point prompts substantially degrade performance. This performance gap is widely observed, yet its internal decoder mechanism remains underexplored. In this work, we revisit the failure of point prompts from a causal perspective. By adapting activation patching to SAM-style decoders with a token-aligned intervention scheme, we localize the earliest recoverable divergence to a single early operation, i.e., the first decoder image-to-token cross-attention update, where prompt-conditioned information first enters the image stream. A sparse autoencoder further reveals that this mediation is concentrated in approximately 50 feature directions out of 4,096, and reverse corruption confirms that the identified layer is the entry point of a propagated image-stream route rather than a locally sufficient state. Guided by this localization, we propose the Causal Bottleneck Adapter (CBA), a lightweight residual module placed at the identified entry layer with the base model frozen. Evaluated on 97,996 medical image-target pairs across CT, MRI, ultrasound, and endoscopy, and validated across various SAM-family models, CBA closes 0.891 of the box-to-point gap with fewer than 9K trainable parameters, outperforming prior repair baselines up to 50× larger. Code is available at `https://github.com/VisualReasoner/medsam-mica`.

1 Introduction

Promptable segmentation foundation models have reshaped visual segmentation by allowing users to specify targets with simple prompts such as points, boxes, or masks. The Segment Anything Model (SAM) [16] is a representative milestone, showing strong zero-shot segmentation across diverse natural images. This paradigm is particularly attractive for medical image segmentation, where ambiguous anatomy, costly annotations, and variable pathologies make human interaction valuable. Medical variants such as MedSAM [20] adapt this architecture to medical imaging and often achieve strong results when targets are specified by informative bounding boxes. However, these models exhibit a striking prompt asymmetry in medical scenarios. On the same image and target, a tight box prompt can produce an accurate mask, whereas a sparse point prompt may collapse to near-zero Dice. This empirical box-to-point gap has been reported across medical SAM benchmarks [23, 12], especially in settings where sparse clicks provide insufficient extent information, yet its internal cause remains poorly understood.

*This work was done prior to joining Amazon.
†Corresponding Author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Existing studies usually treat this gap as a prompt-quality problem. One direction improves prompt robustness with stronger prompt encoders, prompt refinement, or stable prompt propagation, while another direction modifies the mask decoder or introduces adapter-style modules for insufficient prompts [8, 10]. Although these methods improve output masks, they do not answer a prior mechanistic question: *where inside the frozen SAM-family decoder does the successful box trajectory diverge from the failed point trajectory?* This question is well posed in the same-image and same-target setting, where the image, target, and model weights are fixed and only the prompt changes. Thus, the gap should arise from prompt-dependent computation inside the frozen decoder rather than from image content or target identity. Without identifying this, many prompt-repair designs remain largely heuristic in their placement, with modules inserted where architecturally convenient rather than where the failure is causally mediated. Note that we use "causal" in the mechanistic-intervention sense [29] of replacing or corrupting internal activations and measuring the counterfactual effect on the output.

Fig. 1 illustrates our motivation. The left panel shows the SAM-style two-way attention decoder, where sparse prompt-token embeddings interact with dense image embeddings. Specifically, cross-attention from prompt token (as queries) to image embedding (as key&value) updates the prompt embedding stream with image information, and from image embedding (as queries) to prompt token [16] updates the image embedding stream with prompt information. Box and point prompts share this decoder but differ in sparse-token structure and spatial information, i.e., a box provides two corner tokens, while a point provides one positive location.

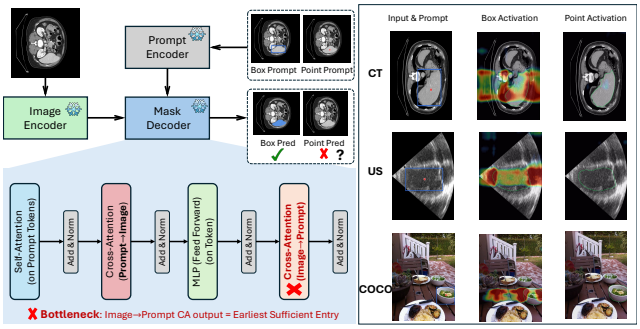

Figure 1: SAM-family mask decoder includes the image-to-prompt token entry layer, where prompt-conditioned information first enters the image stream. CT, ultrasound, and COCO examples show box prompts producing target-aligned activations while point prompts fail to segment the target.

The right panel shows CT, ultrasound, and COCO examples where box-conditioned trajectories produce target-aligned activations, but point-conditioned trajectories fail to inject information to the target region. This suggests that point-prompt failure is an internal decoder routing failure rather than merely an output-level mask error.

In this paper, we revisit point-prompt failure from a causal perspective in the mechanistic-intervention sense [29] and develop a causal diagnosis-to-repair framework for SAM-style decoders.

First, we adapt activation patching from mechanistic interpretability [29] to promptable segmentation. Since box and point prompts induce asymmetric sparse-token structures, direct prompt-token swapping would conflate prompt semantics with token-alignment artifacts. We therefore introduce token-aligned activation patching, restricting interventions to comparable decoder states across prompt types. ***Second***, by sweeping all layers through forward patching (box→point) and reverse corruption (point→box)(Sec.3.1), we find that the box-to-point gap concentrates at the first decoder image-to-token update, where prompt-conditioned information first writes into the image stream. Box→point patching shows that the first decoder image-to-token update contains the missing signal needed to recover point-prompt performance. Point→box corruption further shows that this signal is not confined to that layer: replacing box-conditioned activations with point-conditioned activations at this layer and at downstream image-stream layers consistently destroys box performance. We refer to this chain of dependent image-stream layers as the decoder image-stream route. ***Third***, we characterize this route through sparse-feature and head-level interventions (Sec.3.2). A sparse autoencoder identifies 50 causally selected bottleneck features whose patching generalizes to held-out cases, whereas matched random features have almost no effect; single heads are individually insufficient. ***Finally***, guided by this localization, we propose the *Causal Bottleneck Adapter* (CBA), a lightweight residual module placed at the identified bottleneck with the base model frozen (Sec.3.3). CBA maps the point-conditioned bottleneck activation toward its box-conditioned counterpart during training and uses only the point prompt at inference, enabling efficient repair through causal placement rather than decoder-scale redesign or finetuning.

We conduct extensive experiments on five medical image datasets across four modalities: CT, MRI, ultrasound, and endoscopy. Forward patching at the identified bottleneck recovers 0.985 of the box-to-point Dice gap, and sparse-feature intervention achieves strong held-out rescue with only 50 selected features. We additionally evaluate on COCO [18] as a natural-image control. We further probe homologous entry-layer mediation across SAM-family variants and report volumetric/text-promptable models as external scope checks rather than direct evidence of the same activation route, including SAM [16], MedSAM [20], Medical-SAM2 [22], SAM2 [26], and SAM3 [4]. With a minimal residual design, CBA recovers most missing point-prompt capability with fewer than 9K trainable parameters, keeps the SAM-family backbone frozen, and preserves the native promptable segmentation pipeline. It also outperforms parameter-matched adapters at other decoder locations, confirming that the gain comes from causal placement rather than adapter scale. We also investigate 3D SAM Medical-SAM3 [14] and SAM-Med3D [28] in Appendix N. Our core contributions are summarized as follows:

- We reinterpret the box-to-point gap as a frozen-decoder routing failure and derive a causal-localization-guided repair strategy, instantiated as the lightweight Causal Bottleneck Adapter.
- We introduce token-aligned activation patching for SAM-style decoders, enabling valid causal intervention under asymmetric box and point prompt-token structures.
- We localize point-prompt failure to a propagated image-stream route starting at the first decoder image-to-token update, and characterize it through reverse corruption, sparse-feature mediation, and head-level intervention.
- Extensive experiments across four medical modalities and COCO with multiple SAM-family models show that activation patching at the identified bottleneck recovers 0.985 of the gap, while CBA closes 0.891 from point prompts alone and outperforms stronger repair baselines.

2 Related Work

Promptable Segmentation Foundation Models. Promptable segmentation foundation models segment user-specified targets from flexible prompts such as points, boxes, masks, or text. SAM [16] is the representative model, using a two-way mask decoder to fuse sparse prompt tokens with dense image features. Medical variants such as MedSAM [20] extend this paradigm to medical images while largely preserving the SAM-style decoder architecture. SAM2 [26] further improves temporal modeling and point interaction. Despite strong performance, medical SAM benchmarks [23, 12] consistently show a large box-to-point gap, where boxes substantially outperform sparse clicks across modalities. Unlike prior works that treat this gap mainly as an evaluation phenomenon, we use it as a controlled causal probe where image, target, model weights are fixed, and only the prompt changes.

Prompt Robustness and Decoder Repair. A growing line of work improves SAM-family models under insufficient or noisy prompts by enhancing prompt robustness or modifying the decoder [25, 15, 1]. Stable-SAM [8] stabilizes segmentation under weak prompts through deformable sampling and dynamic routing, while DeSAM [10] adapts the medical SAM decoder for better generalization. Related repair methods introduce additional tokens, prompt refinement, feature fusion, or adapter-style modules to improve output masks. These approaches are effective engineering solutions, but their repair locations are usually chosen by architectural heuristics rather than causal evidence. In contrast, our goal is first diagnostic and then reparative. We first identify the internal decoder computation that mediates the box-to-point gap, and then place a lightweight repair at the localized causal bottleneck[27].

Activation Patching and Mechanistic Intervention. Activation patching is widely used in mechanistic interpretability to localize model behaviors to internal layers of transformer language models [29, 7, 24, 32, 11]. Sparse autoencoders further decompose distributed activations into interpretable or intervention-ready feature bases [3, 6, 30, 9]. Applying these tools to promptable segmentation is non-trivial. Unlike language models, SAM-style decoders operate on dense spatial feature grids and sparse prompt tokens, and their outputs are mask fields rather than next-token distributions. Moreover, box and point prompts induce sparse-token structures with different sizes and semantics, making direct token swapping invalid. We therefore introduce token-aligned activation patching, which restricts interventions to decoder states with comparable shape and semantics across prompt types.

Adapters for Foundation Segmentation Models. Adapters provide a parameter-efficient way to specialize foundation segmentation models to downstream domains or prompting conditions. SAM adapters [31, 5] are typically inserted into the image encoder, prompt pathway, or multiple decoder

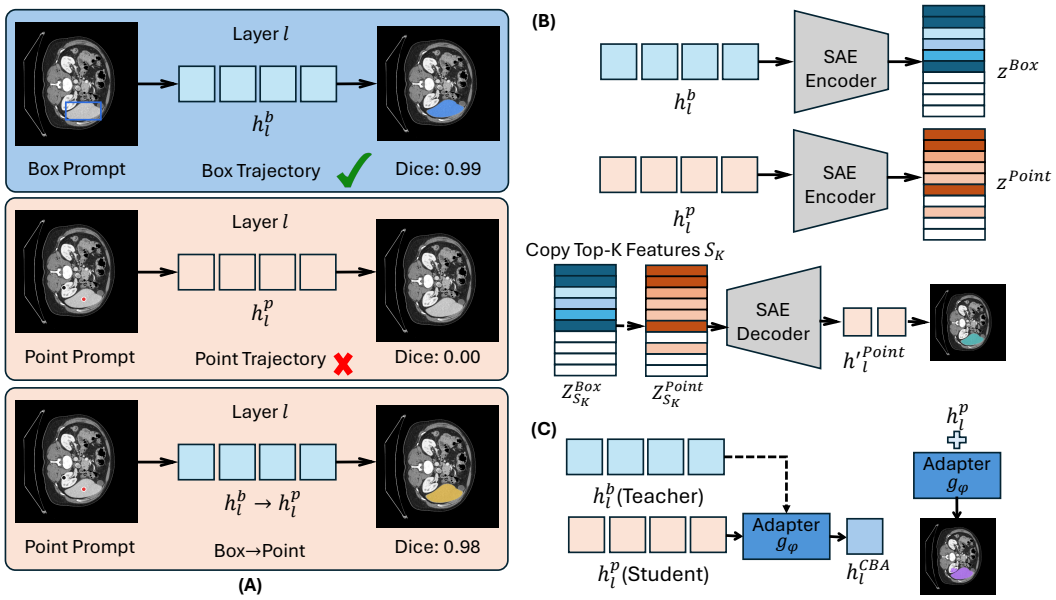

Figure 2: **Three components of analysis at the entry layer.** **(A)** Token-aligned patching: substituting h_ℓ^b for h_ℓ^p at layer ℓ recovers a near-box mask. **(B)** Sparse-feature decomposition: replacing the top-K features (subset S_K) of the SAE-encoded point code with the box code's recovers the trajectory. **(C)** Causal Bottleneck Adapter (CBA): a small residual module trained with only box-point pair trajectories distillation and applied to the point activation at inference, with the decoder frozen.

locations, with placement determined by general design practice. Our Causal Bottleneck Adapter differs in both motivation and placement. Rather than increasing adaptation capacity broadly, CBA is placed at the entry site of the decoder route identified by causal patching. This makes the adapter a consequence of the diagnosis: a small residual correction at the bottleneck can be propagated by the existing decoder computation, allowing CBA to outperform substantially larger repair baselines under the same point-prompt protocol.

3 Methodology

In what follows, we ask a fundamental question: *why do SAM models collapse on point prompts, but can perform well with a box prompt?* We answer the question by intervening directly on the frozen decoder. The investigation proceeds in three systematic stages. We first intervene in whole-layer activations to find where in the decoder the box and point trajectories diverge. Once we have located that layer, we ask whether the divergence is a low-dimensional property of the activation or a distributed one, and decompose the layer's output through a sparse autoencoder. Both stages require access to the box at inference, so they are diagnostic rather than deployable. The third stage uses what the first two have learned to train a small test-time adapter that produces a more box-like activation from the point input alone, with the base model frozen.

3.1 Where does the divergence happen?

SAM and its medical variants share a decoder architecture in which an image encoder E_{img} produces an embedding $\text{emb}(x) = E_{\text{img}}(x)$, a prompt encoder E_{prompt} produces a prompt embedding $\text{emb}(p)$, and a mask decoder D_{mask} fuses the two to produce $\hat{y} = D_{\text{mask}}(\text{emb}(x), \text{emb}(p))$. For each image x and target mask y, we construct two prompts that refer to the same target. The first is a bounding box p_b derived from the ground-truth mask y. The second is a single positive point p_p placed at the center of the largest inscribed disk of y, giving a deterministic foreground location inside the target. The frozen model produces two trajectories $\hat{y}_b = f_\theta(x, p_b)$ and $\hat{y}_p = f_\theta(x, p_p)$ with per-pair Dice gap $\Delta(x, y) = \text{Dice}(\hat{y}_b, y) - \text{Dice}(\hat{y}_p, y)$. Because image, target, and weights are held fixed and only the prompt has changed, divergence arises from the prompt-conditioned computation route in the decoder. Locating where is then a matter of intervention.

Why naive patching does not apply. A straightforward intervention would be to swap the prompt embedding from one trajectory into the other [29]. This is invalid for SAM-style decoders. A box prompt produces two corner tokens with paired position embeddings, while a point prompt produces a single positive-label token. The two prompt-token streams differ in length and meaning, so substituting one for the other conflates the change in prompt content with token-alignment artifacts. We therefore restrict interventions to architectural locations whose activations have comparable shape and meaning under both prompt types. We collect these locations into a set $\mathcal{A}$ that includes the image-stream layers (each carrying an activation tensor of shape $64 \times 64 \times d$ on the dense spatial grid with embedding dimension $d = 256$), the dedicated output-token slot (which occupies a fixed position in the decoder regardless of prompt type), and the outputs of attention operations whose inputs are themselves comparable. We refer to elements of $\mathcal{A}$ as comparable layers, and at any such layer $\ell \in \mathcal{A}$ we write $h_\ell^p(x)$ and $h_\ell^b(x)$ for the point-conditioned and box-conditioned activations produced by the two forward passes.

At a comparable layer $\ell \in \mathcal{A}$, forward patching intervenes on the activation by replacing the point-conditioned tensor $h_\ell^p(x)$ with the box-conditioned tensor $h_\ell^b(x)$,

$$\widetilde{h}_\ell^{p \leftarrow b}(x) = (1 - M_\ell) \odot h_\ell^p(x) + M_\ell \odot h_\ell^b(x), \tag{1}$$

where M_ℓ selects which components of the activation to replace. Whole-layer patching uses $M_\ell = 1$. The **forward patching** (box→point) pass continues through the frozen model and produces $\hat{y}_\ell^{p \leftarrow b}$, and we measure the effect per pair as the fraction of the gap recovered,

$$R_\ell(x, y) = \frac{\mathrm{Dice}(\hat{y}_\ell^{p \leftarrow b}, y) - \mathrm{Dice}(\hat{y}_p, y)}{\Delta(x, y) + \varepsilon}, \quad \varepsilon = 10^{-6}. \tag{2}$$

Averaged across pairs, R_ℓ is the mean rescue at layer ℓ. The **reverse corruption** (point→box) direction $R_\ell^{b \leftarrow p}$ inserts the point-conditioned activation into the box trajectory and measures how much of box performance is destroyed, which gives the necessity test we use in Appendix K.

To locate where the divergence actually happens, we sweep the comparable layers in decoder execution order and compute the mean rescue at each one. The first layer at which patching recovers a box-like trajectory is the candidate entry layer. We make this criterion concrete with two thresholds, one on mean rescue and one on the worst-modality bootstrap lower bound, fixed a priori on a held-out pilot split before any sweep is performed (Appendix F). On the full evaluation, the first decoder image-to-token update is the unique layer that satisfies both thresholds across CT, MRI, ultrasound, and endoscopy. We refer to it as the entry layer and denote it bottleneck in equations.

Whole-layer patching at the entry layer recovers most of the gap on average across modalities (Section 4.1). That locates the divergence to a specific layer and rules out the possibility that the failure is spread thinly across the decoder. The intervention does not, however, identify what about the entry-layer activation does the work. That activation is a dense $64 \times 64 \times 256$ tensor; the rescuing signal could be carried by a small feature subspace or by a distributed property of the whole tensor. To distinguish the two, we need a finer-grained intervention.

3.2 What in the activation mediates the rescue?

We train a sparse autoencoder (SAE) with encoder E_ϕ and decoder G_ϕ on entry-layer activations sampled across the training split (Appendix G). The encoder gives a feature basis $z = E_\phi(h)$ in which we can apply the same patching operation as in Equation (1) but on a chosen subset of features. We rank features by the magnitude of their box-to-point activation difference,

$$r_j = \mathbb{E}_{(x,y) \sim \mathcal{D}_{\text{cal}}} \left| E_\phi(h^b_{bottleneck})_j - E_\phi(h^p_{bottleneck})_j \right|, \tag{3}$$

and patch only the top-K features into the point trajectory by setting M_ℓ in Equation (1) to be one on the top-K positions and zero elsewhere. This asks directly whether the divergence at the entry layer is concentrated in a small number of feature directions: if it is, a small K should suffice to recover the trajectory; if it is not, no compact subset will.

Generalization is evaluated on held-out pairs that were not used for feature ranking; the top-K features are selected only on the calibration split. Specificity is tested against three matched random subsets at the same K: a uniform random subset of features, a subset matched to the top-K in mean activation magnitude, and a subset drawn from the divergence-rank tail below K. We also include a

round-trip control in which the point activation is encoded and decoded through the SAE without modification; this isolates any contribution from the SAE's reconstruction error itself rather than from the feature substitution.

A top-50 feature subset achieves a rescue of 0.923, while matched random subsets recover under 0.005. The mediator at the entry layer is therefore sparse in the SAE feature basis: roughly fifty causally selected directions carry the box-conditioned signal that the point trajectory is missing, and no compact random control approaches it. Whole-layer patching and feature-subset patching both require access to the box-conditioned activation at inference, so they serve as analyses but not as deployable repairs. To convert the localization into something usable at test time, we need to *produce* a box-like activation from a point input rather than *borrow* one from a parallel forward pass.

3.3 Can we exploit the localization?

The first two stages have told us that the box-to-point gap is mediated at the entry layer and that its mediator is a sparse feature subset. If those two facts are true, then a small module trained to map the point-conditioned activation at the entry layer toward its box-conditioned counterpart should close the gap from a point prompt alone, with the rest of the decoder frozen.

We place a two-layer residual MLP g_ψ at the entry layer,

$$h^{\text{CBA}}_{bottleneck}(x, p_p) = h^{p}_{bottleneck}(x) + g_\psi\big(h^{p}_{bottleneck}(x)\big), \tag{4}$$

and freeze the rest of the model. During training, the box trajectory provides supervision through an activation-alignment term (mean squared error over tokens and channels) and the ground-truth mask provides supervision through a downstream segmentation term (Dice plus binary cross-entropy; Appendix C),

$$\mathcal{L}_{\text{CBA}} = \big\| h^{\text{CBA}}_{bottleneck} - \text{sg}\big(h^{b}_{bottleneck}\big) \big\|_2^2 + \lambda_{\text{seg}}\, \mathcal{L}_{\text{seg}}\big(\hat{y}^{\text{CBA}}, y\big), \tag{5}$$

where $\text{sg}(\cdot)$ denotes stop-gradient and the box-trajectory activations are produced by running the frozen base model with p_b on the same image. At inference, the box prompt is not used. The trained g_ψ acts on the point-conditioned activation, and the forward pass continues through the frozen decoder. The CBA does not patch and does not borrow from a parallel trajectory at test time; it produces a box-like activation directly from the point input.

The placement at the entry layer is motivated by the route framing. Figure 3A and Appendix K show that the entry layer is the entry of a propagated decoder image-stream route, by which we mean a sequence of decoder layers along which prompt-conditioned information is carried from the entry layer to the final mask output, with each layer in the sequence individually necessary for box performance. A correction at the entry of such a route rides the route's existing forward computation through downstream layers that already carry the prompt-conditioned signal in the box trajectory, so an entry-layer adapter exploits the route rather than reproducing it from scratch. This predicts that adapters placed at later layers along the same route, at the same parameter budget, should close less of the gap, and that adapters placed off the route should close less still. We test both predictions in Section 4.3.

4 Experiments

We evaluate four questions: (i) where the gap first becomes recoverable, (ii) whether the entry-layer mediator is target-specific and sparse, (iii) whether the route generalizes across SAM-family models and domains, and (iv) whether localization improves test-time repair. We start our experiments with MedSAM ViT-B. Five medical datasets cover four modalities: FLARE22 [21] (CT), CAMUS [17] (ultrasound), Kvasir-SEG [13] (endoscopy), MSD Hippocampus (MRI) and MSD Prostate (MRI) [2], totaling $n = 101{,}186$ image-target pairs. Cross-model analyses use SAM ViT-B/L/H, SAM3, Medical-SAM2, Medical-SAM3, and SAM-Med3D on a matched split, and SAM ViT-B, MedSAM ViT-B, SAM2 hiera-S, and SAM3 on COCO. Full configuration is in Appendix A. For all trainable repair methods, we use only a small high-gap training subset of 3,190 pairs; the remaining 97,996 pairs are held out for testing. We verified coordinate transforms, prompt encoding, multimask selection, and point-format compatibility against the released MedSAM implementation.

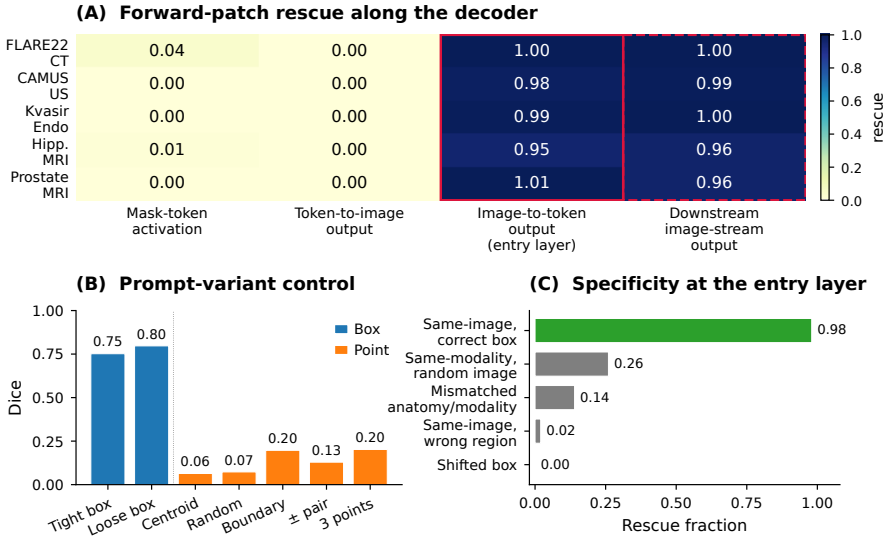

Figure 3: **The box-to-point gap concentrates at a single early decoder layer, the entry layer, and is target-specific.** **(A)** Forward-patch rescue swept layer by layer across the decoder. The sharp no-rescue-to-near-full-rescue transition identifies the entry layer. **(B)** Dice across the box and point prompt variants on the same target. The gap holds for every point construction, ruling out point sampling as the cause. **(C)** Rescue at the entry layer when the matched box activation is replaced with mismatched alternatives. The collapse under every mismatch shows the signal is target-specific.

4.1 The failure localizes to a sparsely mediated route at the entry layer

We first sweep all comparable layers $\ell \in \mathcal{A}$ in decoder execution order and compute the mean rescue R_ℓ at each in Figure 3A. Activations on the prompt side, including the mask-output token slot and the output of the first token-to-image cross-attention, give zero rescue across all five medical datasets. The first image-to-token cross-attention output gives near-full rescue on every modality and is the unique layer satisfying the a priori selection rule (Appendix F), i.e., entry layer. Forward patching at the entry layer recovers 0.985 of the box-to-point gap on average, with worst-modality bootstrap lower bound above 0.94. Reverse corruption confirms necessity: inserting the point-conditioned activation at the entry layer into the box trajectory destroys 0.95 of box performance, and each of three downstream image-stream layers is independently necessary in the same way (Appendix J&K). The route is therefore a chain of layers rather than a single layer, with the entry layer at its head.

Two controls support the localization. Figure 3B sweeps the point-construction rule across centroid, uniform random, boundary-near, $\pm$pair, and three positive points. No point variant approaches either of the two box variants in Dice; even three positive points, the most informative point prompt considered, falls well short of the loose box. The gap is a property of point prompting in general, not of any particular click-placement rule, so the localization in panel A reflects a genuine architectural failure rather than a sampling artifact of how the point was placed.

We test whether the rescue at the entry layer is target-specific in Figure 3C. Replacing the matched box-conditioned activation with mismatched alternatives at the same layer collapses the rescue under every mismatch: a random image of the same modality, a random image of mismatched anatomy or modality, a wrong-region box on the same image, and a shifted box. Only the matched box rescues. The mediator at the entry layer therefore depends on the box being aligned with the segmentation target, not on activation magnitude or generic image context. Sparse-feature decomposition (Table 1) further shows that the mediation is concentrated in roughly fifty causally selected SAE features. We refer to the entry layer together with its propagated chain as the *decoder image-stream route*.

Having established *where* the failure occurs, we next investigate *what* within the entry-layer activation drives it.

Sparse-feature decomposition. We train an SAE on entry-layer activations sampled from training split, then rank features by the magnitude of their box-to-point latent divergence at the causally identified bottleneck. We then intervene on the point trajectory by replacing only the top-K ranked

Table 1: **Sparse SAE features mediate rescue.** Top-ranked bottleneck features recover the box–point gap; random matched features do not.

Intervention	Rescue / effect
Top-1 feature	0.006
Top-5 features	0.028
Top-10 features	0.109
Top-50 features	0.923
All SAE latents	0.986
Point reconstruction	0.000
Uniform random-50	0.005
Activation-matched random-50	0.001
Divergence-matched random-50	0.003
Reverse top-50 corruption	0.780

Table 2: **Main results on the medical evaluation pool** ($n = 97{,}996$). *Patch* and *SAE-50 Mediation* require the box at inference and serve as analyses; the CBA sees only the point prompt at inference. Aggregate metrics are pair-weighted over the full evaluation pool. Rescue is computed per pair and then averaged, rather than derived from aggregate Dice differences.

			Pt		Box		Patch		SAE-50 Mediation		**CBA**	
Dataset	Mod.	n	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU
FLARE22	CT	25,802	0.042	0.021	0.947	0.900	0.931	0.871	0.862	0.758	0.815	0.688
CAMUS	US	62,696	0.038	0.019	0.845	0.732	0.827	0.705	0.712	0.553	0.795	0.660
Kvasir-SEG	Endoscopy	1,000	0.111	0.059	0.909	0.833	0.902	0.821	0.889	0.800	0.875	0.778
MSD Hippocampus	MRI	7,821	0.075	0.039	0.782	0.642	0.767	0.623	0.713	0.554	0.691	0.586
MSD Prostate	MRI	677	0.032	0.016	0.910	0.834	0.886	0.796	0.726	0.570	0.826	0.704
Aggregate	—	97,996	0.042	0.021	0.875	0.778	0.858	0.751	0.811	0.615	0.784	0.645

SAE features with their box-conditioned values, leaving all other SAE features unchanged. Table 1 reports the resulting rescue fractions.

The feature-budget sweep shows that rescue is highly concentrated in a small subset of SAE features. Replacing the top feature alone has little effect ($R = 0.006$), and the top 10 features recover only a modest fraction of the gap ($R = 0.109$). By contrast, replacing the top 50 features recovers most of the box-to-point gap ($R = 0.923$). Replacing all SAE latents nearly matches the full activation patch ($R = 0.986$), while simply encoding and decoding the point activation through the SAE without feature replacement gives essentially no rescue ($R = 0.0003$). Thus the effect is not caused by SAE denoising or reconstruction.

The matched random controls further show that the rescue is specific to the ranked features. Uniform random-50, activation-matched random-50, and divergence-matched random-50 feature sets all recover less than 0.005 of the gap. Finally, replacing the same top-50 features in the successful box trajectory with their point-conditioned values causes substantial reverse damage (0.780), indicating that these features are not only sufficient to rescue point failures, but also support the successful box trajectory. Together, these show the entry-layer mediator is sparse in the SAE feature basis: a small ranked feature subset carries most of the box-conditioned signal that the point trajectory is missing.

Head-level decomposition. The entry layer is the output of an 8-head attention block. If one or two specific heads carried the box-to-point mediation, replacing those alone should approach the all-heads rescue. They do not: replacing all eight heads jointly recovers 0.985, while the best single head recovers only 0.128 and the mean over the eight individual heads is 0.066. No single head approaches the joint effect by an order of magnitude. The full per-head decomposition is in Section Q. The mediator is therefore sparse in the SAE feature basis at the block output but *distributed* across the heads that produce it; block-level granularity is the appropriate resolution for the present analysis.

Table 2 consolidates per-dataset Dice and IoU for the point baseline, the box reference, and the three interventions on MedSAM ViT-B: entry-layer forward patch, top-50 SAE feature patch, and the test-time CBA repair introduced in Section 4.3.

Table 3: Entry-layer patching localizes homologous prompt-conditioned routing behavior across SAM-family variants and natural-image settings. For natural images, we reuse the MedSAM-trained CBA without retraining to test whether the repair targets a modality-agnostic box-to-point routing gap rather than a dataset-specific prior.

	Box		Pt		Patch			SAE-50 Mediation			CBA		
Model	Dice	IoU	Dice	IoU	Dice	IoU	R	Dice	IoU	R	Dice	IoU	R
Medical Images													
MedSAM ViT-B	0.875	0.778	0.042	0.021	0.862	0.757	0.98	0.811	0.682	0.92	0.784	0.645	0.89
SAM ViT-B	0.830	0.709	0.105	0.055	0.643	0.474	0.71	0.573	0.402	0.65	0.720	0.562	0.85
SAM ViT-L	0.781	0.641	0.566	0.395	0.753	0.604	0.87	0.728	0.572	0.75	0.756	0.608	0.88
SAM ViT-H	0.784	0.645	0.566	0.395	0.739	0.586	0.79	0.703	0.542	0.63	0.726	0.570	0.73
Medical-SAM2	0.869	0.768	0.655	0.487	0.821	0.696	0.77	0.795	0.660	0.65	0.715	0.556	0.28
SAM3	0.837	0.715	0.602	0.435	0.829	0.650	0.80	0.784	0.651	0.64	0.730	0.582	0.55
Natural Images													
SAM ViT-B	0.861	0.756	0.617	0.446	0.796	0.661	0.73	0.764	0.618	0.60	0.734	0.580	0.48
MedSAM ViT-B	0.780	0.639	0.131	0.070	0.764	0.618	0.98	0.746	0.595	0.95	0.713	0.554	0.90
SAM2 hiera-S	0.904	0.825	0.858	0.751	0.901	0.820	0.93	0.892	0.805	0.74	0.873	0.775	0.33
SAM3	0.912	0.835	0.875	0.760	0.910	0.831	0.96	0.901	0.814	0.79	0.884	0.795	0.40

4.2 Cross-model/-domain generalization

The localization is not specific to MedSAM ViT-B. Table 3 reports the box-to-point gap together with the entry-layer patch, the top-50 SAE feature patch, and the CBA across multiple SAM-family models on the medical pool, plus four models on COCO. Rows with small box-to-point gaps should be interpreted cautiously, since normalized rescue becomes unstable under small denominators. In particular, SAM2 hiera-S and some volumetric settings already exhibit relatively strong point-prompt performance, making rescue values lower-signal than in the large-gap medical regimes.

The same architectural location mediates the gap across models. SAM ViT-B/L/H and Medical-SAM2 all show non-trivial forward rescue at the homologous first decoder image-to-token update on their own evaluation pools. The route is therefore a property of the SAM-family decoder architecture rather than a quirk of MedSAM specifically.

The route persists outside the medical domain. SAM ViT-B's residual point-prompt failures on COCO are mediated at the same location (rescue 0.733 at normalized gap $(\mathrm{Dice}_b - \mathrm{Dice}_p)/\mathrm{Dice}_b = 0.283$), and MedSAM ViT-B applied directly to COCO shows the same large gap and the same rescue (0.976 at normalized gap 0.832). The route mediates prompt-conditioned spatial routing in SAM-family rather than being a medical fine-tuning artifact.

The CBA generalizes to most but not all of the family. Without retraining, the CBA trained on MedSAM transfers to SAM ViT-B/L/H with rescue 0.73–0.88 and to MedSAM ViT-B on COCO with rescue 0.90. Transfer is weaker in the remaining settings: on Medical-SAM2 the CBA closes 0.28 of the gap where the SAE-50 patch closes 0.65, on SAM3 it closes 0.55, and on SAM ViT-B on COCO it closes 0.48, indicating that the mediating features have shifted enough that the MedSAM-trained adapter does not transfer cleanly. For SAM2 hiera-S and SAM3 on COCO (0.33 and 0.40), the box-to-point gap is below 0.05 Dice, so normalized rescue is a low-signal measure there. The CBA is a route-aligned correction trained on a specific gap, and where the route's geometry shifts or the gap collapses, an entry-layer correction loses its grip.

4.3 Bottleneck repair

The forward patch and the SAE-50 patch are diagnostic interventions that require the box at inference. The Causal Bottleneck Adapter (CBA) replaces the patch with a small learned module placed at the entry layer that produces a more box-like activation from a point input alone, with the rest of the decoder frozen (Section 3.3). Table 4 reports the result in two blocks. The *Baselines* block compares the CBA against four published prior-art SAM repairs at their native parameter counts; the *Placement ablation* block holds adapter architecture and parameter count fixed at $8{,}976$ and varies the location of the residual update. The CBA closes 0.891 of the gap on the medical pool. The strongest prior-art baseline tested, Stable-SAM, closes 0.814 at $9\times$ the CBA's parameter count; HQ-SAM, the largest prior-art adapter, closes 0.804 at $50\times$ the CBA's count. SAM-PT and DeSAM both close less than 0.76. The CBA's advantage is not parameter scale; the placement ablation block makes that explicit. At the same $8{,}976$ parameters, the CBA's entry-layer placement closes 0.21 more of the gap than the

Table 4: Compare the CBA against other SOTA SAM repairs on medical images at their native parameter counts. Placement ablation holds CBA's architecture and parameters, and we only switch location, isolating the contribution of CBA on different layers.

Method	Repair locus	Pt Dice	R	Trainable params
Point baseline	—	0.042	0.000	0
Box reference (privileged)	—	0.875	1.000	0
Baselines				
DeSAM [10]	mask decoder (modified)	0.645	0.725	0.34 M
SAM-PT [25]	prompt pipeline (point propagation)	0.674	0.759	0
HQ-SAM [15]	mask decoder (HQ output token + fusion)	0.712	0.804	0.45 M
Stable-SAM [8]	mask attention (DSP/RTS)	0.720	0.814	0.08 M
Placement ablation				
Final token-to-image adapter	final-block token-to-image	0.492	0.544	0.009 M
Layer-1 image-to-token adapter	layer-1 image-to-token	0.607	0.685	0.009 M
Layer-0 token-to-image adapter	layer-0 token-to-image	0.740	0.835	0.009 M
Activation alignment only (no $\mathcal{L}_{\text{seg}}$)	layer-0 image-to-token (entry layer)	0.751	0.853	0.009 M
Causal Bottleneck Adapter (ours)	layer-0 image-to-token (entry layer)	0.784	0.891	0.009 M

next downstream control (layer-1 image-to-token, 0.685), 0.06 more than the layer-0 token-to-image control on the prompt side, and 0.04 more than a state-alignment-only ablation at the same entry layer. Location contributes more to the CBA's effect than either parameter count or loss design.

5 Conclusion

We introduced same-image asymmetric box-to-point counterfactuals as a controlled way to study sparse-prompt failures in frozen SAM-family decoders. Token-aligned activation patching localizes the measurable box-to-point gap to an early image-stream update, and reverse corruption, specificity controls, head-level analysis, and SAE mediation show that this site acts as the entry point of a propagated decoder route under our mechanistic intervention protocol. Guided by this localization, CBA repairs point-prompt trajectories with fewer than 9K trainable parameters while using only point prompts at inference. The remaining gap to the box trajectory and the mixed transfer across model variants suggest that future repair methods should jointly model route placement, model-specific feature geometry, and boundary refinement.

Acknowledgments and Disclosure of Funding

This study was supported in part by the NVIDIA Academic Grant Program and the Google Gemini Academic Grant Program. The authors declare no competing interests.

References

[1] Salma J. Ahmed, Emad A. Mohammed, and Azam Asilian Bidgoli. Med-SegLens: Latent-level model diffing for interpretable medical image segmentation. *arXiv preprint arXiv:2602.10508*, 2026.

[2] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, et al. The medical segmentation decathlon. *Nature communications*, 13(1):4128, 2022.

[3] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

[4] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, RISHI HAZRA, Shuangrui Ding, Sagar Vaze, Francois Porcher,

Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In *The Fourteenth International Conference on Learning Representations*, 2026.

[5] Tianrun Chen, Lanyun Zhu, Chaotao Ding, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Papa Mao, and Ying Zang. SAM-Adapter: Adapting segment anything in underperformed scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 3367–3375, 2023.

[6] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[7] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *Transformer Circuits Thread*, 2022.

[8] Qi Fan, Xin Tao, Lei Ke, Mingqiao Ye, Yuan Zhang, Pengfei Wan, Zhongyuan Wang, Yu-Wing Tai, and Chi-Keung Tang. Stable segment anything model. 2023.

[9] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

[10] Yifan Gao, Wei Xia, Dingdu Hu, Wenkui Wang, and Xin Gao. DeSAM: Decoupled segment anything model for generalizable medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 509–519. Springer, 2024.

[11] Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024.

[12] Yuhao Huang, Xin Yang, Lian Liu, Han Zhou, Ao Chang, Xinrui Zhou, Rusi Chen, Junxuan Yu, Jiongquan Chen, Chaoyu Chen, et al. Segment anything model for medical images? *Medical Image Analysis*, 92:103061, 2024. doi: 10.1016/j.media.2023.103061.

[13] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas De Lange, Dag Johansen, and Håvard D Johansen. Kvasir-seg: A segmented polyp dataset. In *International conference on multimedia modeling*, pages 451–462. Springer, 2019.

[14] Chongcong Jiang, Tianxingjian Ding, Chuhan Song, Jiachen Tu, Ziyang Yan, Yihua Shao, Zhenyi Wang, Yuzhang Shang, Tianyu Han, and Yu Tian. Medical SAM3: A foundation model for universal prompt-driven medical image segmentation. *arXiv preprint arXiv:2601.10880*, 2026.

[15] Lei Ke, Mingqiao Ye, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, Fisher Yu, et al. Segment anything in high quality. volume 36, pages 29914–29934, 2023.

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023.

[17] Sarah Leclerc, Erik Smistad, Joao Pedrosa, Andreas Østvik, Frederic Cervenansky, Florian Espinosa, Torvald Espeland, Erik Andreas Rye Berg, Pierre-Marc Jodoin, Thomas Grenier, et al. Deep learning for segmentation using an open large-scale dataset in 2d echocardiography. *IEEE transactions on medical imaging*, 38(9):2198–2210, 2019.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[19] Anglin Liu, Rundong Xue, Xu R. Cao, Yifan Shen, Yi Lu, Xiang Li, Qianqian Chen, and Jintai Chen. MedSAM3: Delving into segment anything with medical concepts. *arXiv preprint arXiv:2511.19046*, 2025.

[20] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15:654, 2024. doi: 10.1038/s41467-024-44824-z.

[21] Jun Ma, Yao Zhang, Song Gu, Cheng Ge, Shihao Mae, Adamo Young, Cheng Zhu, Xin Yang, Kangkang Meng, Ziyan Huang, et al. Unleashing the strengths of unlabelled data in deep learning-assisted pan-cancer abdominal organ quantification: the flare22 challenge. *The Lancet Digital Health*, 6(11):e815–e826, 2024.

[22] Jun Ma, Zongxin Yang, Sumin Kim, Bihui Chen, Mohammed Baharoon, Adibvafa Fallahpour, Reza Asakereh, Hongwei Lyu, and Bo Wang. Medsam2: Segment anything in 3d medical images and videos. *arXiv preprint arXiv:2504.03600*, 2025.

[23] Maciej A. Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: an experimental study. *Medical Image Analysis*, 89:102918, 2023. doi: 10.1016/j.media.2023.102918.

[24] 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.

[25] Frano Rajič, Lei Ke, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Segment anything meets point tracking. pages 9284–9293, 2025.

[26] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, 2025.

[27] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. Causal mediation analysis for interpreting neural nlp: The case of gender bias. *arXiv preprint arXiv:2004.12265*, 2020.

[28] Haoyu Wang, Sizheng Guo, Jin Ye, Zhongying Deng, Junlong Cheng, Tianbin Li, Jianpin Chen, Yanzhou Su, Ziyan Huang, Yiqing Shen, Bin Fu, Shaoting Zhang, Junjun He, and Yu Qiao. SAM-Med3D: Towards general-purpose segmentation models for volumetric medical images. *arXiv preprint arXiv:2310.15161*, 2023.

[29] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023.

[30] Philipp Wesp, Robbie Holland, Vasiliki Sideri-Lampretsa, and Sergios Gatidis. Sparse autoencoders for interpretable medical image representation learning. *arXiv preprint arXiv:2603.23794*, 2026.

[31] Junde Wu, Wei Ji, Yuanpei Liu, Huazhu Fu, Min Xu, Yanwu Xu, and Yueming Jin. Medical SAM adapter: Adapting segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.12620*, 2023.

[32] Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*, 2023.

Table 5: Data partitions used in the paper.

Split	Pairs	Usage
Full medical pool	101,186	Overall dataset construction
Repair training subset	3,190	Trainable repair optimization
Held-out medical test pool	97,996	Main evaluation in Tables 2 and 4
COCO evaluation pool	40,670	Cross-domain evaluation
Held-out MSD tasks	4,261	External validation

Reproducibility Statement

Code, configurations, hook locations, SAE checkpoints, the CBA checkpoint, and per-figure CSVs are available at `https://github.com/VisualReasoner/medsam-mica`. Random seeds (three for all repair methods), bootstrap configuration, the a priori thresholds for entry-layer selection, per-dataset prompt construction, and exact module paths are in Sections A and E.

A Datasets, Splits, Prompt Construction and CBA training

Datasets. The core 2D evaluation pool spans four medical modalities and one natural-image control. FLARE22 [21] provides abdominal CT with thirteen organ classes. CAMUS [17] provides apical-view echocardiography for left-ventricle endocardium, left-ventricle epicardium, and left atrium. Kvasir-SEG [13] provides polyp segmentation in colonoscopy. MSD Hippocampus and MSD Prostate [2] provide MRI volumes, which we slice into 2D image-target evaluation pairs. COCO [18] provides natural-image instance masks for the cross-domain check in the cross-model section. Two additional MSD tasks (Liver CT and Heart MRI) are reserved for the held-out generalization checks reported in Section O. Table 5 reports our detailed dataset splits.

Separate from this 2D pool, we run 3D scope checks with Medical-SAM3 [19] and SAM-Med3D [28] on target volumes. These experiments are used to assess whether prompt-type gaps also appear in volumetric SAM-family settings; they are not part of the 2D activation-patching and SAE intervention benchmark.

Pair construction. For each medical dataset, we materialize image–target pairs: one image and one ground-truth mask per pair, with multi-class images exploded into one pair per class. Pairs whose mask occupies less than 0.1% of the image are discarded. The full medical pool contains 101,186 image–target pairs across the five medical datasets. After excluding the 3,190 repair-training pairs, the held-out medical test pool used in Tables 2 and 4 contains 97,996 pairs.

For natural-image evaluation, we use 40,670 COCO image-target pairs. COCO is used only for cross-domain evaluation and is not mixed with the medical training or held-out medical test pool. We report two COCO settings. In the adapter-transfer setting, the MedSAM-trained CBA is evaluated directly on COCO without additional retraining. In the COCO-specific setting, the adapter is trained on a separate 3,190-pair high-gap COCO subset and evaluated on the remaining held-out COCO pairs.

Prompts. The default box prompt is the tight axis-aligned bounding box around the target mask. The default point prompt is the center of the largest inscribed disk of the target mask. Box and point prompts on the same target are paired across the analysis so that activation patching always operates on identical image inputs. Loose-box, centroid, boundary-near, multi-positive, and positive-negative variants are evaluated in Section P.

Box prompts as a privileged signal. Box prompts in this paper are derived from ground-truth masks and should be read as a privileged object-extent specification, not a clinically free signal. The role of the box trajectory is to provide a successful reference whose internal activations are target-specific; the analysis localizes a route that the failed point trajectory could in principle exploit. The CBA's test-time use does not require a box.

B CBA Architecture and Parameter Count

The Causal Bottleneck Adapter (CBA) is a small residual module applied to the entry-layer image-stream activation $h \in \mathbb{R}^{HW \times C}$ with $H{=}W{=}64$ and $C{=}256$. It is a token-wise two-layer MLP with a low-rank bottleneck:

$$\mathrm{CBA}(h) \;=\; h \;+\; W_2\,\mathrm{GELU}\big(W_1\,\mathrm{LN}(h) + b_1\big) + b_2,$$

where LN is LayerNorm over the channel axis, $W_1 \in \mathbb{R}^{r \times C}$, $W_2 \in \mathbb{R}^{C \times r}$, and the bottleneck width is $r{=}16$.

Architectural specifics. The MLP is applied *independently per spatial token*: every position (i, j) in the $64{\times}64$ activation grid receives the same weights, so there are no spatial parameters and no position-conditioned gating. Weights are shared across all spatial positions and across all five medical datasets. There is no dropout. We initialize W_2 to zero so the adapter's initial output equals the identity, which is the standard "identity at $t{=}0$" adapter initialization. The LayerNorm operates on the $C{=}256$ channel axis with per-token mean and variance.

Parameter count. Table 6 reports the exact arithmetic. The total is $8{,}976$ trainable parameters, which is the value reported in Table 4 of the main paper.

Table 6: CBA parameter count. All entries are trainable; the base model is frozen.

Component	Shape	Parameters
LayerNorm scale γ	256	256
LayerNorm shift β	256	256
Down-projection W_1	16×256	4,096
Down bias b_1	16	16
Up-projection W_2	256×16	4,096
Up bias b_2	256	256
Total		**8,976**

Why a full-activation MLP rather than an SAE-feature module. The SAE analysis in Section 3.2 and Appendix G establishes that the box-point activation difference at bottleneck is sparse in the SAE basis: top-50 of $4{,}096$ features capture 0.923 of the rescue. The CBA does not operate in the SAE basis. Instead, it is trained to reshape the full point-conditioned activation toward the box-conditioned target. The two are connected by the supervision signal: the $\mathcal{L}_{\text{act}}$ loss in Equation (6) drives the adapted point activation toward the box trajectory's activation, which is the same activation that the SAE shows is mediated by ~ 50 sparse features. The full-MLP form is a deployment choice: it avoids the SAE's ~ 1.05M parameters at inference and runs as a single fused operation on the existing activation tensor. The SAE result motivates the existence of a low-dimensional representation worth recovering; the CBA's small bottleneck width ($r{=}16$, comparable in order to the SAE's $K{=}50$) is consistent with this motivation but not derived from it.

C CBA Training

Data and split. Training uses $3{,}190$ pairs of medical images across 5 datasets defined in Section A. We use the same remaining test sets across all CBA seeds and across all placement-control variants in Section R; the same splits are also used to train the prior-art baselines reported in Section S, so all adapter methods see identical training and validation data.

Each training step samples a batch of 32 image-target pairs. For each pair we run the box trajectory once with no gradient tracking to record the target activation $h^b_{bottleneck}$, then run the point trajectory with the CBA inserted at $bottleneck$ to obtain the adapted activation $\tilde{h}^p_{bottleneck} = \mathrm{CBA}(h^p_{bottleneck})$, and complete the forward pass through the frozen decoder to produce the predicted mask $\hat{y}^p_{\text{CBA}}$. The combined loss is

$$\mathcal{L} \;=\; \mathcal{L}_{\text{act}} \;+\; \lambda_{\text{seg}}\,\mathcal{L}_{\text{seg}}. \tag{6}$$

Activation-alignment loss. $\mathcal{L}_{\text{act}}$ is a token-wise mean-squared error between the adapted point activation and the (detached) box-conditioned activation,

$$\mathcal{L}_{\text{act}} \;=\; \frac{1}{B \cdot HW \cdot C} \sum_{b=1}^{B} \sum_{i=1}^{HW} \sum_{c=1}^{C} \big(\tilde{h}^{p}_{b,i,c} - \text{sg}\big[h^{b}_{b,i,c}\big]\big)^2,$$

where $\text{sg}[\cdot]$ is the stop-gradient operator: the box activation is detached so that gradients flow only through the CBA. The mean is taken over batch, spatial-token, and channel axes jointly; this normalizes the per-batch scale of $\mathcal{L}_{\text{act}}$ to be invariant to B, H, W, and C, which keeps the relative weight against the segmentation loss stable across configurations. With this normalization, $\mathcal{L}_{\text{act}}$ has a typical magnitude of $\mathcal{O}(10^{-2})$ at random initialization and converges to $\mathcal{O}(10^{-3})$ at the end of training.

$\mathcal{L}_{\text{seg}}$ is the standard SAM-family Dice plus binary cross-entropy mask loss against the ground-truth target mask, computed on the segmentation produced by completing the point forward pass with the adapted activation in place. The SAM-family decoder predicts three candidate masks per prompt; we follow the SAM convention of taking the predicted-IoU-maximizing mask for the loss, with no oracle access to ground-truth Dice. $\mathcal{L}_{\text{seg}}$ has a typical magnitude of $\mathcal{O}(10^{-1})$ at random initialization.

Choice of λ_{seg}. We set λ_{seg}=1. This was selected by a small sweep on the CBA validation split over $\lambda_{\text{seg}} \in \{0.1, 0.3, 1, 3, 10\}$ at the entry-layer placement on a single seed, before any test-set evaluation. Validation Dice was within 0.005 of the maximum across the $\{0.3, 1, 3\}$ range and dropped by 0.02–0.04 at the endpoints $\{0.1, 10\}$. We report the λ_{seg}=1 configuration throughout. The same value is used for every placement-control adapter and for the prior-art baselines that expose a Dice/activation loss tradeoff, so the comparison in Table 4 of the main paper does not give the entry-layer CBA an advantage from per-placement tuning.

Optimization. We use Adam with learning rate 3×10^{-4}, weight decay 10^{-4}, β_1=0.9, β_2=0.999, no learning-rate warmup, and no learning-rate schedule. Gradient norms are clipped at 1.0. Training runs for 10 epochs over the 3,190-pair training split, approximately $1{,}000$ optimizer steps in total. We monitor validation Dice every 100 steps and report the checkpoint with the highest validation Dice (early-stopping on plateau). Convergence is reached by epoch 6 across all seeds and placements. Each reported number uses three random seeds with distinct CBA initialization, training-batch order, and prompt sampling; tables report the mean over seeds, and per-seed ranges for the placement study are given in Table 20.

Hardware. A single training run completes in approximately 18 minutes on one NVIDIA A100-80GB. The full main-paper CBA campaign (entry-layer placement on MedSAM ViT-B and SAM ViT-B, three placement controls, three seeds each, plus the λ_{seg} sweep) totals approximately 5 GPU-hours.

D Prompt-Necessity Ablation for the CBA

A possible reading of the CBA's segmentation result is that the adapter has learned dataset and anatomy priors and produces a correct segmentation regardless of the point prompt's content. We rule this out with four ablations on the CBA validation split, all using the entry-layer placement and the same trained CBA checkpoint. Each ablation modifies the point prompt at inference while leaving everything else (image, CBA weights, base model) untouched.

The ranking is clear. The correct foreground point gives Dice 0.756, near the box-trajectory reference of ~ 0.83 on this split. Replacing the point with a foreground point on the same target (a different click on the same object) drops Dice to 0.692 — a small drop, consistent with the point still encoding target-relevant information. Shuffling the point to another image's location drops to 0.184; shuffling within the same image to a different target drops to 0.241; removing the point entirely drops to 0.093. The CBA's output is therefore not an image-conditioned segmentation prior; it requires a target-relevant point for the segmentation to recover. The adapter repairs the routing operation; it does not memorize dataset content.

Table 7: Prompt-necessity ablation for the entry-layer CBA on the validation split. The CBA's repaired Dice depends sharply on the prompt: replacing or shuffling the point collapses output by 0.5–0.6 Dice. The CBA repairs point-conditioned routing rather than learning an image-conditioned segmentation prior.

Inference-time prompt to CBA-equipped MedSAM ViT-B	Dice
Correct point (default)	0.7561
Random foreground point on the target	0.6923
Shuffled point: another image's correct point	0.1842
Shuffled point: a different target in same image	0.2407
No point (image only, prompt encoder zero input)	0.0934

E Hook Locations and Module Paths

All forward-patching, reverse-corruption, and SAE intervention experiments operate on PyTorch forward hooks attached to the following named modules of the SAM-family mask decoder. Module names follow the released SAM checkpoint conventions; MedSAM, SAM2, and SAM-Med3D inherit the same naming.

Table 8: Hook attachment points for each comparable layer $\ell \in \mathcal{A}$. The entry layer bottleneck is the first image-to-token cross-attention output.

Comparable layer	PyTorch module path
Mask-output token slot	`mask_decoder.transformer.layers.0.norm1`
Token-to-image output, layer 0	`mask_decoder.transformer.layers.0.cross_attn_token_to_image`
Image-to-token output, layer 0 (entry layer bottleneck)	`mask_decoder.transformer.layers.0.cross_attn_image_to_token`
Token-to-image output, layer 1	`mask_decoder.transformer.layers.1.cross_attn_token_to_image`
Image-to-token output, layer 1	`mask_decoder.transformer.layers.1.cross_attn_image_to_token`
Final token-to-image	`mask_decoder.transformer.final_attn_token_to_image`

For the image stream, the activation has shape $[B, H \times W, C]$ with $H{=}W{=}64$ and $C{=}256$ at MedSAM ViT-B's decoder resolution. Patching replaces the entire image-stream tensor at the named module's output; no spatial sub-selection is applied at the patching step. The MedSAM ViT-B encoder, prompt encoder, and mask-token slot are frozen throughout.

F A Priori Entry-Layer Selection Rule

Before any patching results on the full pool are inspected we make the investigation rule: The earliest comparable layer $\ell \in \mathcal{A}$ in decoder execution order whose mean rescue $\bar{R}_\ell \geq 0.5$ on the pilot, and whose worst-dataset bootstrap lower bound $\mathrm{LCB}_{0.95}(R_\ell) \geq 0.4$ across the five medical datasets.

The first image-to-token cross-attention output is the unique comparable layer satisfying both conditions on the pilot. The mask-output token slot and the layer-0 token-to-image output have mean rescue near zero. Later comparable layers satisfy the rule trivially but do not appear earlier than the entry layer in execution order. This selection rule is fixed before Section 4.1's full-pool evaluation. It is not a pre-registration in the formal sense, which we acknowledge in the limitations.

G SAE Training and Top-K Curve

Training. The SAE is a single-hidden-layer overcomplete autoencoder $E_\phi : \mathbb{R}^{256} \to \mathbb{R}^{4096}$ followed by a tied-weight decoder. We train on activations harvested at bottleneck from both box and point trajectories on the training set, $5{\times}10^7$ activation tokens total, with an L_1 sparsity penalty $\lambda{=}5{\times}10^{-4}$. Training runs for 10 epochs at batch size 8192 with Adam at $\eta{=}10^{-3}$. The trained SAE achieves reconstruction L_2 relative error 0.07 on a held-out split of activations with the same number of images as training set. The SAE is then frozen for all downstream feature-ranking and intervention experiments.

Feature ranking. For each feature $j \in \{1, \ldots, 4096\}$, we compute the rank

$$r_j = \frac{1}{|D_{\text{train}}|} \sum_{x \in D_{\text{train}}} \left| E_\phi(h^b_{bottleneck}(x))_j - E_\phi(h^p_{bottleneck}(x))_j \right|,$$

on the training sets, and take the top-K features as S_K. We sweep $K \in \{1, 5, 10, 20, 30, 50, 100, 200\}$ and report rescue at each in Figure 4, shows rescue against K. Rescue is near zero for $K \leq 10$, rises sharply between $K=10$ and $K=50$, and saturates near 0.92 for $K \geq 50$, indicating the route is supported by approximately 50 dominant features out of $4{,}096$ rather than uniformly distributed across the SAE basis. The Top-10 subset captures only 0.109 of the rescue, showing that the route is sparse but not extremely so; a handful of features is not enough.

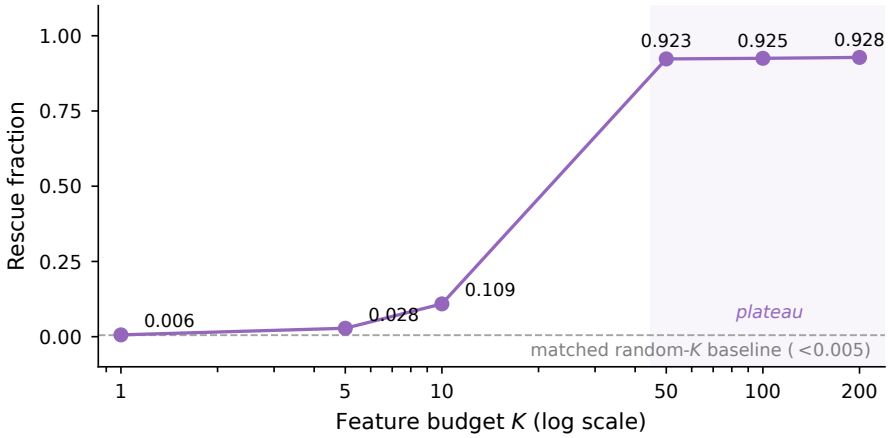

Figure 4: **SAE feature-budget sweep at `bottleneck`.** Rescue is near zero for $K \leq 10$, rises sharply between $K=10$ and $K=50$, and saturates monotonically near 0.92 for $K \geq 50$ ($0.923 \rightarrow 0.925 \rightarrow 0.928$ at $K=50, 100, 200$). Matched random-K controls recover under 0.005 at every K.

H SAE Spatial Feature Maps

Beyond the causal-mediation result, the top-ranked SAE features admit qualitative spatial inspection. Figure 5 visualizes box-conditioned and point-conditioned activations of selected causal features over input images, alongside the top-50 patched recovery. Features align with target interiors and boundaries under box prompts and collapse under point prompts. We make no claim of clinical semantic identity for these features beyond their interventional role.

Table 9 summarizes candidate modality/anatomy tags for eight features from Figure 5, computed as the dataset and anatomy purity over the displayed top-activating examples. Purity is 1.00 for seven of eight features. We report these as candidate semantic descriptions rather than verified clinical pathology labels.

Table 9: Candidate semantic tags for eight causal SAE features. Δ_{b-p} is the mean activation difference between box and point trajectories at bottleneck (positive: box-induced; negative: point-active). Purity is computed over the displayed top-activating examples.

Feature	Candidate tag	Role	Δ_{b-p}	Dataset purity	Anatomy purity
1960	CT liver foreground	box-induced routing	$+0.172$	1.00	1.00
749	CT vessel/aorta-IVC	box-route feature	-0.614	1.00	0.75
3153	US LV endocardium	box-induced routing	$+0.179$	1.00	1.00
424	Endoscopy polyp	box-induced routing	$+0.130$	1.00	1.00
612	MRI hippocampus	small-structure routing	$+0.126$	1.00	1.00
1246	MRI prostate	point-failure routing	-0.058	1.00	1.00
2529	CT kidney	point-active failure	-0.815	1.00	1.00
338	Endoscopy polyp	point-active failure	-0.127	1.00	1.00

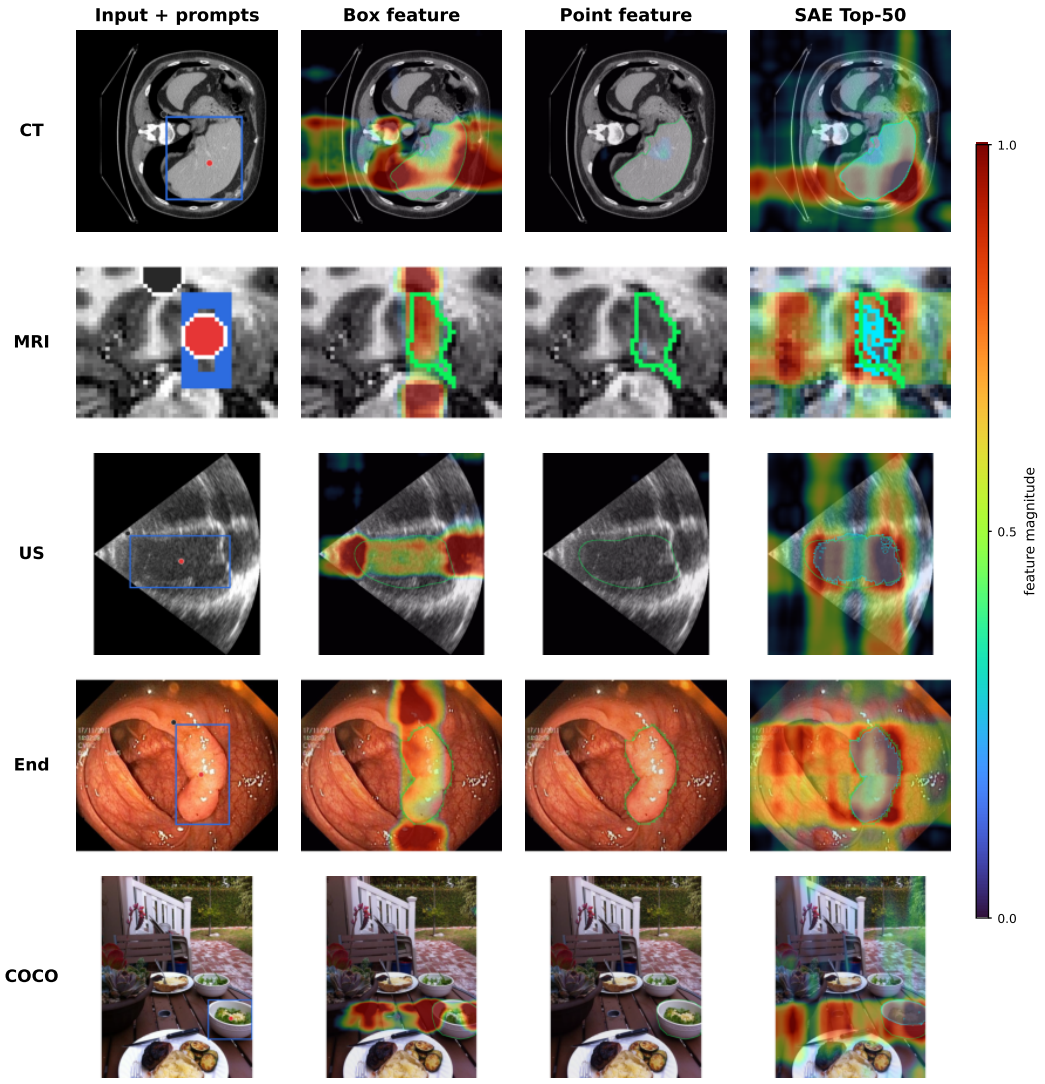

Figure 5: Spatial feature maps for representative SAE features at bottleneck on five image-modality rows (CT, MRI, ultrasound, endoscopy, COCO). For each row, the columns show the input with prompts (left), the box-conditioned feature activation, the point-conditioned feature activation, and the top-50 SAE-patched recovery. Box-conditioned activations concentrate over the target region; point-conditioned activations are diffuse or absent.

I Activation-Level Recovery at bottleneck

Table 10 reports activation-level distances at bottleneck on 5 representative cases from Figure 5. Cosine distance between the point activation and the box-conditioned reference is 0.11 to 0.34; full patching restores it to zero by construction; the top-50 SAE feature patch reduces cosine distance to 0.014 to 0.049 and feature-subspace distance to 0.11 to 0.19, an order of magnitude closer to box than point. Output Dice tracks activation distance: where SAE feature mediation moves the activation closer to box, the segmentation moves closer to the ground truth.

Table 10: Activation-level evidence on five selected cases in Figure 5. Cosine distance and relative L_2 are measured at bottleneck between the indicated trajectory and the box-conditioned reference. Full patching is zero by construction.

Case	Trajectory	Dice	Cosine dist.	Rel. L_2	Feature subspace
CT	Box	0.992	0.000	0.000	0.000
	Point	0.000	0.124	0.525	0.578
	Full patch	0.988	0.000	0.000	0.000
	Top-50 SAE patch	0.976	0.014	0.169	0.106
MRI	Box	0.897	0.000	0.000	0.000
	Point	0.007	0.337	0.950	0.953
	Full patch	0.853	0.000	0.000	0.000
	Top-50 SAE patch	0.789	0.049	0.311	0.186
US	Box	0.959	0.000	0.000	0.000
	Point	0.000	0.111	0.547	0.584
	Full patch	0.956	0.000	0.000	0.000
	Top-50 SAE patch	0.779	0.014	0.171	0.107
End	Box	0.984	0.000	0.000	0.000
	Point	0.000	0.231	0.676	0.696
	Full patch	0.982	0.000	0.000	0.000
	Top-50 SAE patch	0.980	0.026	0.227	0.132
COCO	Box	0.978	0.000	0.000	0.000
	Point	0.000	0.095	0.456	0.489
	Full patch	0.961	0.000	0.000	0.000
	Top-50 SAE patch	0.751	0.012	0.156	0.093



Figure 6: Forward-patch rescue at bottleneck across five subset definitions. Shaded band reports bootstrap 95% confidence intervals. The dotted line at 0.90 marks where rescue would still mediate the majority of the gap; all subsets, including the point-success regimes, fall above this line.

J Full Forward Layer Sweep

Table 11 reports the complete forward-patch rescue across 26 patch points in the MedSAM ViT-B mask decoder, averaged over the five medical datasets. The Figure 3A heatmap in the main paper collapses these into four representative groups; the full sweep below shows the underlying granularity.

The rescue divides cleanly into three regimes: (i) image-stream activations downstream of the entry layer recover near-full rescue (≥ 0.98), (ii) token-side activations in the first decoder block recover roughly 0.20, and (iii) all later token-side activations recover near zero. The first image-to-token cross-attention output (the entry layer bottleneck) is the earliest patch point with rescue above the 0.5 pilot threshold, and the only earliest point that satisfies the worst-dataset bootstrap rule from

Section F. The mask-decoder low-resolution logit patch is included as a sanity check; rescue at that endpoint is trivially 1.0 because the patch directly substitutes the box-trajectory output.

Table 11: Full forward-patch sweep across all instrumented activations in the MedSAM ViT-B mask decoder. Mean rescue is averaged across the five medical datasets in the main pool. The entry layer bottleneck (`layer0_image_to_token_attn_out`) is bolded.

Patch point	Rescue
`mask_decoder_low_res_logits` (sanity check)	1.0000
`layer0_image_to_token_attn_out` (bottleneck)	0.9849
`layer0_block_output_image_stream`	0.9849
`layer0_image_after_image_to_token_norm`	0.9849
`transformer_output_image_stream`	0.9840
`pre_mask_upsampling_embedding`	0.9840
`layer1_block_output_image_stream`	0.9811
`layer1_image_after_image_to_token_norm`	0.9811
`layer0_tokens_after_mlp_norm`	0.2304
`layer0_tokens_after_token_to_image_norm`	0.2304
`layer0_tokens_after_self_attn_norm`	0.2304
`layer0_token_self_attn_out`	0.2304
`layer0_token_mlp_out`	0.1941
`layer1_image_to_token_attn_out`	0.0980
`layer1_tokens_after_self_attn_norm`	0.0324
`layer1_token_self_attn_out`	0.0321
`layer1_tokens_after_mlp_norm`	0.0234
`layer1_tokens_after_token_to_image_norm`	0.0234
`layer1_token_mlp_out`	0.0184
`layer1_token_to_image_attn_out`	0.0162
`final_tokens_after_norm`	0.0080
`transformer_output_tokens`	0.0080
`final_token_to_image_attn_out`	0.0078
`layer0_block_output_tokens`	0.0055
`layer1_block_output_tokens`	0.0003
`layer0_token_to_image_attn_out`	0.0001

The 0.20-rescue band on layer-0 token-side activations (rows in the middle group) is small in absolute terms but consistently positive, suggesting these activations carry partial prompt-conditioned information after the box trajectory has read into the image stream once. The effect collapses to near zero by layer 1 and remains there through the final token-to-image attention. This pattern is consistent with the route being a *forward chain* initiated at the entry layer rather than a back-and-forth between token and image streams: once the prompt-conditioned spatial signal is written into the image stream, the token side no longer carries the load.

K Reverse Corruption

Forward patching at the entry layer shows that swapping in the box-conditioned activation *suffices* to rescue point output. The dual question is whether the same activation is *necessary*: replacing the box-trajectory activation at a comparable layer with the point-trajectory activation should destroy box performance if and only if that layer mediates the gap. We perform reverse corruption at the entry layer and at the three downstream image-stream layers identified in the forward sweep.

Table 12 reports the destruction fraction $D_\ell = 1 - \mathrm{Dice}_{\text{corrupted}}/\mathrm{Dice}_{\text{box}}$ at each layer. Reverse corruption at the entry layer destroys 0.95 of box performance. Each of the three downstream image-stream layers, corrupted independently, is similarly destructive (≥ 0.93). Reverse corruption at prompt-side activations destroys less than 0.05 of box performance, mirroring the asymmetry of the forward sweep. The chain of layers is therefore each individually necessary; the route is a propagated chain rather than a single point of failure.

Table 12: Reverse-corruption destruction at each instrumented image-side comparable layer, averaged across the five medical datasets. Each layer's box-trajectory activation is replaced with the corresponding point-trajectory activation, and box performance is re-evaluated. The entry layer is bolded.

Layer corrupted in box trajectory	Destruction D_ℓ
Mask-output token slot	0.02
Token-to-image output, layer 0	0.04
Image-to-token output, layer 0 (bottleneck)	0.95
Image-to-token output, layer 1	0.96
Final image-stream output	0.94

L Specificity Controls

Table 13 reports the full set of mismatched controls used in Figure 3C (Section 4.1). The matched same-image box recovers 0.98 of the gap; every mismatched alternative collapses the rescue. Sample sizes are n=1,266 pairs from the calibration bank, evaluated at bottleneck.

Table 13: Specificity controls at bottleneck. Each row replaces the matched box-conditioned activation with the indicated mismatched alternative.

Activation source at bottleneck	Rescue
Same-image, matched box (reference)	0.98
Same-modality, random image	0.26
Mismatched anatomy or modality, random image	0.14
Same-image, wrong-region box	0.02
Same-image, shifted box ($\geq$ 0.5 IoU drop)	0.00

M Multimask Oracle Ablation

SAM-family decoders predict three candidate masks per prompt; the default head selects one. We test whether mask ranking can explain the box-to-point gap by replacing the default top-1 choice with an oracle that selects the best of the three by Dice against ground truth.

Table 14: Multimask oracle ablation on the calibration bank. Best-of-three oracle ranking closes only a small fraction of the gap, ruling out mask ranking as the cause of the failure.

Configuration	Mean Dice
Box, top-1	0.875
Box, oracle best-of-three	0.881
Point, top-1	0.042
Point, oracle best-of-three	0.094
Gap closed by oracle ranking	0.05

The oracle closes only 0.05 of the gap. The remaining 0.95 of the gap is therefore not a ranking problem at the output head; it is upstream of the mask-token decoding step.

N Volumetric and Text-Prompt Comparisons

These experiments are external scope checks on volumetric and text-promptable SAM-family models. They test whether the box-point or geometric-text gap persists under different prompt modalities.

SAM-Med3D [28]. 3D click prompts on CT and MRI volumes (Table 15). True 3D context reduces sparse-prompt difficulty for some anatomies (heart and hippocampus improve with more clicks) but does not uniformly close the gap on small structures.

Table 15: SAM-Med3D volume comparison, mean Dice over four target volumes per dataset.

Dataset	1 click	3 clicks	5 clicks
FLARE22 CT	0.926	0.928	0.927
MSD Liver CT	0.923	0.927	0.924
MSD Heart MRI	0.718	0.786	0.838
MSD Hippocampus MRI	0.233	0.280	0.328
MSD Prostate MRI	0.848	0.866	0.870
Mean	0.730	0.757	**0.778**

SAM2-small / Medical-SAM2. On eight converted BTCV-style target volumes, click prompts obtain Dice 0.655 while bounding boxes obtain 0.869.

Medical-SAM3 [14, 19]. The public Medical-SAM3 inference wrapper exposes box and text prompts (not the SAM1 point/box decoder route). Table 16 reports box and text Dice test sets per dataset. Text prompting is strong for endoscopic polyps (0.854, comparable to box) but substantially weaker for CT, MRI, and ultrasound, mirroring the geometric-versus-sparse phenomenon analyzed in the main paper.

Table 16: Medical-SAM3 box vs. text prompt validation on test sets. Text prompting is comparable to box on endoscopic polyps but markedly weaker on CT, MRI, and ultrasound.

Dataset	Box Dice	Text Dice
FLARE22 CT	0.946	0.173
CAMUS ultrasound	0.710	0.355
Kvasir-SEG endoscopy	**0.853**	**0.854**
MSD Hippocampus MRI	0.735	0.187
MSD Prostate MRI	0.727	0.225
Mean	0.794	0.359

O Compact Held-Out Generalization

We test whether the entry-layer route and SAE mediation replicate on two held-out MSD tasks (Liver CT and Heart MRI) that were not used to discover or calibrate the route. Table 17 and Figure 7 confirm that they do, with forward-patch rescue above 0.97 and held-out top-50 SAE mediation above 0.94 on both. Random-50 controls remain near zero.

Table 17: Compact held-out generalization on two MSD tasks not used during method development.

Dataset	Modality	Pairs	Box Dice	Point Dice	Norm. gap	Box→Pt	Pt→Box	Top-50 / Rand-50
MSD Liver	CT	3,761	0.837	0.055	0.935	0.996	−0.966	**0.982** / 0.000
MSD Heart	MRI	500	0.715	0.184	0.742	0.977	−0.804	**0.942** / 0.007

P Prompt-Variant Full Table

Table 18 reports mean and median Dice across all evaluated prompt variants on the five medical datasets. The gap between any box variant and any point variant is large on every dataset; no point construction rule comes within 0.4 Dice of either box variant. The success and failure columns report the fraction of pairs above and below the conventional clinical-utility thresholds ($\mathrm{Dice} \geq 0.7$ for success, $\mathrm{Dice} \leq 0.2$ for failure).

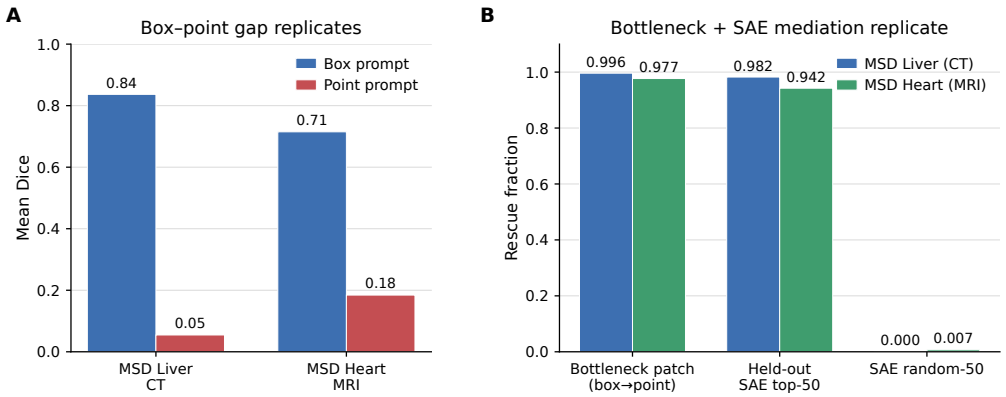

Figure 7: Compact held-out generalization. MSD Liver CT and MSD Heart MRI replicate the entry-layer route and SAE mediation effects, with similar rescue magnitudes and near-zero random-50 controls.

Q Head-Level Decomposition

We probe whether the rescue at the entry layer is carried by a small subset of attention heads or by their joint computation. The first image-to-token cross-attention block uses 8 heads. We patch each head's contribution individually: replace the box trajectory's value for head j at bottleneck into the point trajectory while leaving the other seven heads at their point-trajectory values, run the rest of the decoder, and measure rescue. Patching all 8 heads jointly is the all-heads condition reported throughout the main paper.

No single head approaches the joint effect. The best single head recovers 0.128 while the joint patch recovers 0.985, an order of magnitude difference. The mean over the eight individual heads is 0.066. The route at the entry layer is therefore sparse in the SAE feature basis at the block output but *distributed* across the heads that produce it; block-level granularity is the appropriate resolution for the present analysis. Sub-head-level structure is left to future work.

R CBA Ablations

The main paper's CBA placement ablation is on MedSAM ViT-B. As a cross-model robustness check, we also run the placement ablation on SAM ViT-B applied to the high-gap medical pool, training a parameter-matched ~ 9 K-parameter adapter at each of four placements with three random seeds per variant. Table 20 reports the results.

The entry layer is the best single-layer placement on SAM ViT-B (gap closed 0.872), with the layer-0 token-to-image attention the closest competitor (0.856). Both single-layer placements within the route outperform the multi-layer route variant (0.798), which spreads the same parameter budget across multiple image-stream layers, and the layer-1 image-to-token placement (0.698), which is downstream of the route's main propagation. The seed-to-seed range of the entry-layer placement (0.818–0.911) and the token-to-image placement (0.842–0.868) overlap, but the entry-layer placement's mean exceeds the token-to-image placement's maximum on the most favorable seed; the ordering is consistent with the MedSAM ViT-B result in the main paper, where the entry layer is also the best single-layer placement.

The placement ordering on SAM ViT-B replicates the main-paper finding: route-relevant placements substantially outperform off-route placements (0.872–0.856 versus 0.698 at layer-1 image-to-token), and within the route the entry layer is the best single-layer choice. The multi-layer route's under-performance relative to the single-layer entry-layer placement is consistent with a fixed parameter budget being most effective when concentrated at the route's earliest write rather than spread across the propagated chain.

Table 18: Prompt-variant Dice across the five medical datasets. Each row is a (dataset, prompt) pair evaluated on the indicated number of pairs. "Success" is the fraction of pairs with Dice ≥ 0.7; "Failure" is the fraction with Dice ≤ 0.2.

Dataset	Prompt variant	n	Mean Dice	Median Dice	Success ≥ 0.7	Failure ≤ 0.2
FLARE22 CT	Tight box	25802	0.947	0.900	0.894	0.014
	Loose box (+10%)	25802	0.898	0.953	0.919	0.003
	Centroid	25802	0.042	0.021	0.001	0.939
	Random positive	25802	0.059	0.002	0.010	0.904
	Boundary-near positive	25802	0.236	0.132	0.056	0.563
	Three positive	25802	0.265	0.225	0.049	0.465
	Three positive + 1 negative	25802	0.243	0.123	0.109	0.586
	Default ± pair	25802	0.163	0.057	0.059	0.725
CAMUS US	Tight box	62696	0.827	0.705	0.626	0.011
	Loose box (+10%)	62696	0.741	0.831	0.654	0.000
	Centroid	62696	0.038	0.019	0.000	0.998
	Random positive	62696	0.061	0.029	0.000	0.943
	Boundary-near positive	62696	0.161	0.107	0.016	0.771
	Three positive	62696	0.148	0.130	0.000	0.716
	Three positive + 1 negative	62696	0.157	0.135	0.000	0.663
	Default ± pair	62696	0.065	0.027	0.000	0.896
Kvasir-SEG endoscopy	Tight box	1000	0.902	0.821	0.918	0.006
	Loose box (+10%)	1000	0.883	0.943	0.896	0.006
	Centroid	1000	0.111	0.059	0.000	0.983
	Random positive	1000	0.045	0.006	0.003	0.936
	Boundary-near positive	1000	0.191	0.073	0.054	0.665
	Three positive	1000	0.158	0.133	0.000	0.679
	Three positive + 1 negative	1000	0.183	0.139	0.012	0.613
	Default ± pair	1000	0.111	0.039	0.006	0.764
MSD Hippocampus MRI	Tight box	7821	0.767	0.623	0.567	0.045
	Loose box (+10%)	7821	0.737	0.780	0.739	0.008
	Centroid	7821	0.075	0.039	0.001	0.809
	Random positive	7821	0.115	0.064	0.003	0.799
	Boundary-near positive	7821	0.217	0.177	0.014	0.542
	Three positive	7821	0.238	0.217	0.004	0.452
	Three positive + 1 negative	7821	0.143	0.107	0.002	0.709
	Default ± pair	7821	0.129	0.103	0.001	0.734
MSD Prostate MRI	Tight box	677	0.886	0.796	0.495	0.088
	Loose box (+10%)	677	0.722	0.818	0.627	0.039
	Centroid	677	0.032	0.016	0.000	0.865
	Random positive	677	0.069	0.025	0.000	0.896
	Boundary-near positive	677	0.152	0.107	0.000	0.710
	Three positive	677	0.144	0.107	0.000	0.719
	Three positive + 1 negative	677	0.251	0.242	0.024	0.454
	Default ± pair	677	0.224	0.270	0.000	0.355

In Table 21, we also report trained CBA from base models and compare with pretrained CBA from MedSAM. The performance has further improved, demonstrating that CBA has the ability to further reduce the box-point gap if we train them using its own base model.

S Prior-Art SAM Repair Baselines

The CBA comparison in Section 4.3 (Table 4) includes four published prior-art SAM repair methods.

Stable-SAM [8]. We implement the Deformable Sampling Plugin and Dynamic Routing Plugin at the published locations, using the published parameter budget (0.08M) and training protocol on our dataset.

DeSAM [10]. We implement the Prompt-Relevant IoU Module and Prompt-Decoupled Mask Module at the published locations on the MedSAM ViT-B mask decoder, with the parameter budget reported in the original paper (0.34M).

Table 19: Per-head patching at the entry layer. Each row reports forward-patch rescue when only the indicated head's box-trajectory output is substituted into the point trajectory at bottleneck, leaving the other seven heads at their point-trajectory values.

Intervention	Rescue
All 8 heads jointly	0.9850
Head 7 only (best single head)	0.1278
Head 2 only	0.0820
Head 5 only	0.0818
Head 4 only	0.0683
Head 6 only	0.0650
Head 3 only	0.0428
Head 0 only	0.0360
Head 1 only	0.0239
Mean of single-head rescues	0.0660

Table 20: CBA placement ablation on SAM ViT-B, three random seeds per variant. Baseline point Dice 0.105 and box Dice 0.830 are identical across variants because the base model is frozen. "CBA at entry layer" is the placement used in the main paper; the other rows are off-route placement controls matched in parameter budget.

Placement	Repaired Dice	Gap closed	Gap closed (min–max)
CBA at entry layer (bottleneck)	0.7344	**0.8722**	0.8182–0.9111
Token-to-image attention, layer 0	0.6514	0.8560	0.6213–0.8355
Multi-layer route	0.6787	0.7981	0.7740–0.8240
Image-to-token attention, layer 1	0.6081	0.6977	0.6454–0.7350

HQ-SAM [15]. We use the released HQ-Token and HQ-Output module at the published locations. Parameter budget 0.45M as published.

SAM-PT [25]. SAM-PT does not introduce new parameters at the decoder; it adds point-tracking pre-processing on top of frozen SAM. We evaluate its base configuration without the tracking module since the single-image setting does not provide temporal context.

For each baseline, we follow the published training protocol on our datasets with our prompt distribution, then evaluate on the same held-out datasets used for the CBA. The parameter counts reported in the main-paper table reflect the trainable parameters of each method's adapter, not the full backbone.

T Discussion and Limitations

We investigate prompt sensitivity in SAM-family models as a question about what happens inside a frozen decoder, not about prompt engineering or architectural redesign. The finding that the gap collapses to a single layer's writing operation has two consequences worth foregrounding. First, the box-to-point gap is not a defect of SAM specifically. It is what happens when a single architecture is asked to handle prompts of very different information density. The two-block two-way decoder treats both box and point prompts as sparse tokens, and the moment they enter the image stream is the same operation regardless of which they are; whichever prompt type fails to carry enough signal at that moment will fail to route, and the rest of the decoder cannot recover. This is an interpretation of the result rather than a claim of it, but it predicts that any future promptable foundation model that re-uses the SAM decoder pattern will inherit the same vulnerability, and that the diagnostic we use here will localize it. Second, the CBA's outperformance of substantially larger prior-art adapters suggests that placement-aware interventions are a different category of repair than scale-aware ones. Most existing SAM-repair work scales adapter capacity at fixed locations chosen by convention; if the route framing is right, the next round of SAM-repair work should choose locations from causal evidence and let the parameter budget follow, not the other way around.

Table 21: Model-specific CBA comparison across medical and natural-image settings using Dice.

Model	Setting	Strong prompt	Weak prompt	MedSAM CBA	Own CBA
Medical image					
MedSAM ViT-B	2D medical	Box	Point	0.784	0.784
SAM ViT-B	2D medical	Box	Point	0.720	0.782
SAM ViT-L	2D medical	Box	Point	0.756	0.800
SAM ViT-H	2D medical	Box	Point	0.726	0.791
Medical-SAM2	3D medical	Box	Click	0.715	0.826
Medical-SAM3	2D medical	Box	Text	–	0.747
SAM-Med3D	3D medical	5 clicks	1 click	–	0.790
Natural image					
MedSAM ViT-B	2d natural	Box	Point	0.713	0.790
SAM ViT-B COCO	2d natural	Box	Point	0.734	0.780
SAM2 hiera-S	2d natural	Box	Point	0.873	0.890

Several questions our analysis raises but does not answer are worth flagging as open work. The patching and SAE interventions close the gap to within 5% of the box reference, while the CBA closes only most of the gap. The remaining margin is the difference between what the route framing says is recoverable in principle and what a small learned residual recovers in practice. What lives in that margin, whether it is parameter capacity, loss formulation, or features the SAE basis does not capture cleanly, is a question the diagnostic raises but does not resolve. The route mediates target localization but not boundary refinement; a parallel investigation of the boundary-refinement step, with the same family of interventions, would tell us whether boundary refinement also concentrates at a single later layer or is distributed across the rest of the decoder. The CBA's varying transfer across SAM-family models points at a finer question about whether a route shares the same feature subspace across models that share the architecture, or whether each model's training shifts the subspace enough that per-model adapters are required; the cross-model patching results are consistent with the latter but a definitive answer requires per-model SAE training that we do not run here. Finally, the diagnostic itself is not specific to segmentation, and whether it localizes prompt sensitivity in non-SAM promptable foundation models is the natural next test of its scope.

Broader impacts. On the positive side, the diagnostic explains why promptable segmentation models that look strong under box prompts can fail under the single clicks that are cheapest for clinicians to provide, and the CBA narrows this gap at very low computational cost without retraining the base model. On the negative side, the repaired click-prompted models still fall short of the box reference (Table 2), transfer unevenly across SAM-family variants (Table 3), and have not been evaluated in any clinical workflow. A higher average Dice could encourage over-trust in individual click-based masks, and box-prompt benchmarks can themselves overstate how well a model serves click-based use. Masks produced by repaired models should therefore be reviewed by qualified clinicians and are not intended for unsupervised diagnostic use.

Limitations of the protocol. Box prompts are derived from ground-truth masks and serve only as a privileged reference for analysis and as training supervision for the CBA (Appendix A). The entry-layer selection thresholds were fixed on a pilot split before the full-pool sweep (Appendix F), but this was not a formal pre-registration.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 claim that the box-to-point gap is localized to the first decoder image-to-token update (Section 4.1; Appendix J and K), that this mediation is concentrated in about 50 SAE features (Table 1; Appendix G), and that the CBA repairs point-prompt trajectories at test time (Tables 2 and 4). Cross-model, natural-image, volumetric, and text-prompt results are presented as scope and transfer checks (Section 4.2; Appendix N), not as proof of an identical activation route in every model.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix T (Discussion and Limitations) covers the remaining gap between diagnostic patching and the CBA, the absence of a boundary-refinement analysis, uneven cross-model transfer, and the scope of the diagnostic. It also states that box prompts are derived from ground-truth masks and serve only as a privileged reference (Appendix A) and that the entry-layer selection rule was fixed on a pilot split but not formally pre-registered (Appendix F); Section 4.2 notes that normalized rescue is unstable when the box-to-point gap is small.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper makes no theoretical claims; its equations define interventions, evaluation metrics, and training losses.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3 defines the interventions and the rescue metric. Appendix A gives the datasets, splits, and prompt construction; Appendix B and C give the CBA architecture, losses, and optimization; Appendix E lists the exact hook locations; Appendix F and G give the layer-selection rule and the SAE configuration; and Appendix S describes the baseline protocols.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code, configurations, hook definitions, SAE and CBA checkpoints, and per-figure CSVs are available at `https://github.com/VisualReasoner/medsam-mica` (see the Reproducibility Statement). All datasets and base checkpoints are public and are obtained from their original providers (Appendix A).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 summarizes the setting. Appendix A gives the data splits and prompt construction, Appendix C gives the CBA hyperparameters, the optimizer, and how λ_{seg} was chosen, Appendix G gives the SAE training details, and Appendix S gives the baseline protocols.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The entry layer is selected with a per-dataset bootstrap lower confidence bound fixed in advance (Section 4.1; Appendix F), and Figure 6 shows bootstrap 95% confidence intervals for the bottleneck rescue across five subset definitions. CBA results are means over three seeds that vary initialization, batch order, and prompt sampling (Appendix C), with per-seed ranges reported for the placement study (Table 20); the main MedSAM ViT-B results are also broken down by dataset (Table 2).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix C reports the hardware (one NVIDIA A100-80GB), the time per CBA training run (about 18 minutes), and the total compute of the CBA campaign (about 5 GPU-hours). We do not separately report the compute used for SAE training, the activation-patching sweeps, the prior-art baselines, or preliminary experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics. The work uses only publicly released medical imaging datasets and model checkpoints under their providers' terms, involves no new data collection from human subjects, and discusses clinical risks in Appendix T.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The Broader impacts paragraph of Appendix T discusses positive impacts (explaining and cheaply reducing click-prompt failures, which could make interactive segmentation less laborious) and negative impacts (over-trust in click-based masks that still fall short of box prompts and transfer unevenly across models, the absence of clinical validation, and box-prompt benchmarks that overstate click-based performance), and states that outputs require clinician review and are not intended for unsupervised diagnostic use.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The released artifacts (code, SAE checkpoints, and an 8,976-parameter adapter for frozen segmentation models) are not generative and pose no high risk of misuse; no scraped data are released.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: All datasets (Appendix A), base checkpoints, and baseline implementations (Appendix S) are credited by citation and were used under the terms published by their providers, but the paper does not state the license name of each asset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code, SAE checkpoints, and CBA checkpoint are documented in the repository (`https://github.com/VisualReasoner/medsam-mica`) with usage instructions, the training configurations of Appendix B, C, and G, a license, and known limitations. No new data involving people are collected or released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing and no research with human subjects; all images and annotations come from existing public datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work uses only existing, publicly released datasets and involves no interaction with human participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the method, the experiments, the models under analysis, or the proposed repair; any use for writing or editing assistance is outside the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2026/MainTrackHandbook`) for what should or should not be described.