PEMA: EXTERNAL ADAPTATION OF FROZEN 3D MEDICAL MLLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained multimodal large language models (MLLMs) perform poorly on specialized medical question-answering tasks, while task-specific fine-tuning can substantially improve accuracy at high computational cost. We investigate whether similar performance gains can be obtained by training a small adapter without updating the pretrained MLLM. We introduce Performance-Enhancing Multimodal Adapter for MLLMs (PEMA-MLLM, abbreviated PEMA), an adapter placed between the multimodal projector and the language model. PEMA modifies the projected visual tokens conditioned on the question, while the vision encoder, multimodal projector, and language model remain frozen. We validate our proposed concept on M3D-LaMed models with Phi-3-4B and LLaMA-2-7B backbones on closed-ended 3D-RAD Tasks 4–6. On Phi-3, PEMA achieves 67.40% mean subtask-macro accuracy, compared with 29.86% for the frozen model and 68.79% for the released fine-tuned M3D-RAD checkpoint, while training only 12.08M parameters, approximately 0.30% of the model. PEMA also substantially improves the LLaMA backbone and is competitive with rank-8 LoRA, which introduces low-rank updates within the language model. In our implementations, PEMA uses fewer trainable parameters and achieves about 1.71 to 1.90$\times$ training-time speedup over LoRA, although it does not reduce peak GPU memory. These results show that much of the task-specific performance improvement of a pretrained 3D medical MLLM can be recovered through a compact external adapter while leaving the underlying MLLM unchanged. Post-hoc controls show that similar Task 6 accuracy can be obtained without meaningful image information or with substantially simpler adaptation: zero-image training reaches 73.73% accuracy and a 3,072-parameter constant residual reaches 72.18%, compared with 73.74% for standard PEMA. These results show that high benchmark accuracy does not by itself establish strong use of the medical image.

1 INTRODUCTION

Multimodal large language models (MLLMs) combine images and text to answer questions. In medical visual question answering, these models must relate a clinical question to information in a medical image (Lin et al., 2021). M3D-LaMed supports this task for three-dimensional CT volumes by combining a 3D vision encoder, a multimodal projector, and a language model (Bai et al., 2024). We study its Phi-3-4B and LLaMA-2-7B versions (Abdin et al., 2024; Touvron et al., 2023).

Pretraining does not guarantee good performance on a specific medical task. The original M3D-LaMed models have low accuracy on the closed-ended tasks in 3D-RAD, while the fine-tuned M3D-RAD models achieve much higher scores (Gai et al., 2025). However, updating a large model requires gradients and optimizer states for the parameters being trained. Keeping a separate fine-tuned copy for each task also increases storage costs. This raises our main question: can a small adapter obtain similar accuracy improvements without changing the pretrained MLLM?

We introduce the *Performance-Enhancing Multimodal Adapter for MLLMs* (PEMA-MLLM, abbreviated PEMA) to study this question. PEMA is placed between the multimodal projector and the language model (see Fig. 1). It uses the question to learn a correction to the projected visual tokens, then passes the modified tokens to the language model. Only the adapter is trained; the vision encoder,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

projector, text embeddings, and language model remain frozen. Different tasks can therefore share one pretrained MLLM and store only their own adapter parameters.

PEMA is a sequential adapter: it modifies the visual-token sequence before the language model processes it. The adapter itself uses a residual connection to preserve the original tokens. We also compare PEMA with Low-Rank Adaptation (LoRA), a widely used parameter-efficient method (Hu et al., 2022). LoRA adds trainable low-rank branches in parallel with selected linear transformations inside the language model. This comparison tests PEMA against an established alternative that also leaves the original model parameters unchanged.

We evaluate the original models, PEMA, and rank-8 LoRA on closed-ended 3D-RAD Tasks 4–6. PEMA and LoRA use the same training examples, answer-scoring objective, and checkpoint-selection rule. We evaluate the released fine-tuned Phi-3 checkpoint as a reference and report published fine-tuned LLaMA results separately. The released models were trained by their authors, so the comparison measures achieved accuracy rather than the effect of different methods under identical fine-tuning conditions.

On Phi-3, PEMA raises mean benchmark accuracy from 29.86% to 67.40%, close to the released fine-tuned model's 68.79%. It trains 12.08M parameters, approximately 0.30% of the combined model. PEMA also gives large improvements on LLaMA and is competitive with LoRA. In our implementations, it uses fewer trainable parameters and trains 1.71–1.90 times faster than LoRA, although its peak GPU memory use is higher.

The contribution of this work is a visual-token adapter for frozen 3D medical MLLMs and an evaluation of its accuracy and training cost. We also test whether the accuracy gains depend on the image and whether simpler adapters can achieve similar results. These tests reveal important limits: frequent answers and question patterns account for much of the benchmark performance. We therefore present PEMA as a way to improve task accuracy, not as evidence of reliable medical image understanding.

2 Related Work

Medical visual question answering. Visual question answering predicts an answer from an image and a question (Antol et al., 2015). Medical VQA adds specialized terminology and the need to interpret medical images (Lin et al., 2021). For volumetric imaging, the model must process information across a 3D scan rather than a single two-dimensional image.

M3D and M3D-LaMed. M3D studies multimodal learning for 3D medical image analysis (Bai et al., 2024). Its M3D-LaMed models combine a volumetric vision encoder, a multimodal projector, and a language model. The projector converts image features into visual tokens that the language model processes together with text. Our work builds on the pretrained Phi-3-4B and LLaMA-2-7B versions of M3D-LaMed. Rather than change their existing weights, PEMA learns a correction to the projected visual tokens before they enter the language model.

3D-RAD. 3D-RAD provides a benchmark for question answering over 3D radiology images, with tasks covering image observations, medical computation, existence detection, and temporal diagnosis (Gai et al., 2025). Its authors also report M3D-RAD models obtained by adapting M3D-LaMed, with substantial accuracy improvements over the original models. We use the benchmark's closed-ended tasks to investigate whether training PEMA can obtain similar improvements while keeping M3D-LaMed frozen. The released fine-tuned model provides an accuracy reference, not a model trained under our experimental settings. Section 4 describes the selected tasks, data splits, and checkpoints.

Parameter-efficient training. Parameter-efficient fine-tuning (PEFT) trains a small number of parameters instead of updating the whole pretrained model. Examples include bottleneck adapters (Houlsby et al., 2019), prompt and prefix tuning (Lester et al., 2021; Li & Liang, 2021), and side networks (Zhang et al., 2020). Visual Prompt Tuning learns additional inputs to a vision transformer (Jia et al., 2022); VL-Adapter studies adapters for vision-language tasks (Sung et al., 2022b). Ladder Side-Tuning trains a separate network using features from a frozen backbone (Sung et al., 2022a). With PEMA, the adapter's output is passed through the frozen language model. Training must therefore backpropagate through the language model even though its weights do not change.

Multimodal adapters. LLaMA-Adapter V2 introduces visual information through additions to a language model (Gao et al., 2023). CROME combines image and text features in an adapter before a frozen LLM (Ebrahimi et al., 2024), which is closely related to PEMA's placement. Flamingo and BLIP-2 also connect vision and frozen language components through trainable attention modules (Alayrac et al., 2022; Li et al., 2023). PEMA applies a question-conditioned residual adapter to the existing visual tokens of a pretrained 3D medical MLLM. Our focus is task-specific adaptation of an existing medical model, rather than learning its initial image–language connection.

LoRA. LoRA represents an update to a frozen weight matrix as the product of two low-rank matrices (Hu et al., 2022). Its trainable branch runs in parallel with the original linear transformation. It is therefore a useful comparison for PEMA: LoRA changes selected transformations within the language model, whereas PEMA changes the visual tokens before they enter it.

Answer imbalance and image use. VQA models can exploit answer frequencies or patterns in question text (Goyal et al., 2017; Agrawal et al., 2018). High accuracy alone therefore does not establish that a model uses the image correctly. In addition to benchmark accuracy, we report balanced accuracy, predicted-answer counts, and tests with shuffled images or all-zero inputs. These tests show whether predictions depend on image information, but do not establish that the model identifies the correct anatomy or is clinically reliable.

3 METHOD

PEMA modifies the visual tokens that M3D-LaMed supplies to its language model. The adapter uses the question to compute a residual update, which is added to the original tokens. The language model then scores the possible answers. This section describes the adapter and the training objective. Figure 1 shows the system at inference, the training workflow, and the adapter's internal structure.

3.1 INPUTS AND FROZEN MODEL

Each example contains a CT volume I, a question q, an ordered set of candidate answers $\mathcal{C} = \{c_1, \ldots, c_K\}$, and the correct answer index $y \in \{1, \ldots, K\}$. Here K is the number of candidate answers and c_k is the answer at index k. The prompt p contains the model's image placeholders, the question, all answer choices, and "Answer:". It does not include the candidate answer being scored. Let f_{vision} be the frozen vision encoder, P_{mm} the frozen multimodal projector, and E_{text} the frozen text-embedding layer. For a batch of inputs, the projected visual tokens V_i and prompt features $\boldsymbol{T}$ are

$$V_i = P_{\text{mm}}(f_{\text{vision}}(I)) \in \mathbb{R}^{B \times N_v \times d_v} \tag{1}$$

and

$$\boldsymbol{T} = E_{\text{text}}(p) \in \mathbb{R}^{B \times N_t \times d_t}. \tag{2}$$

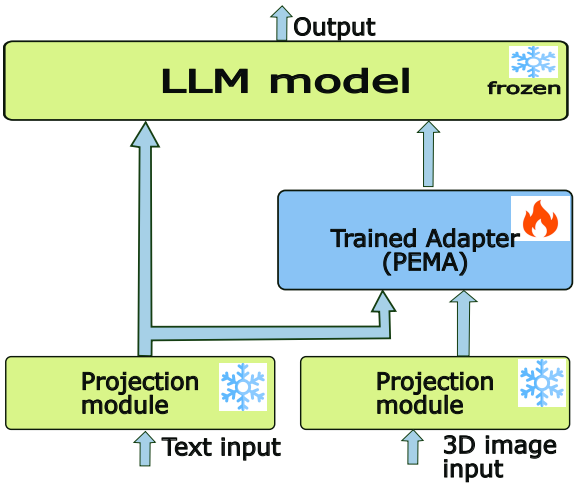
Here B is the batch size, N_v and N_t are the numbers of visual and prompt tokens, and d_v and d_t are their embedding dimensions. $\mathbb{R}$ denotes the real numbers. We follow the figures in writing the visual-token tensors as V_i, V_p, and V_a, where the subscripts denote input, predicted residual, and adapted tokens, respectively; i is not an example index here. Other vectors, matrices, and feature tensors are shown in bold. Both input features are computed without gradients and detached before entering PEMA. We denote the frozen model parameters by Φ and the trainable adaptation parameters by Θ. For PEMA, Θ includes all adapter weights, biases, and the scalar gate parameter.

3.2 SEQUENTIAL VISUAL-TOKEN ADAPTER

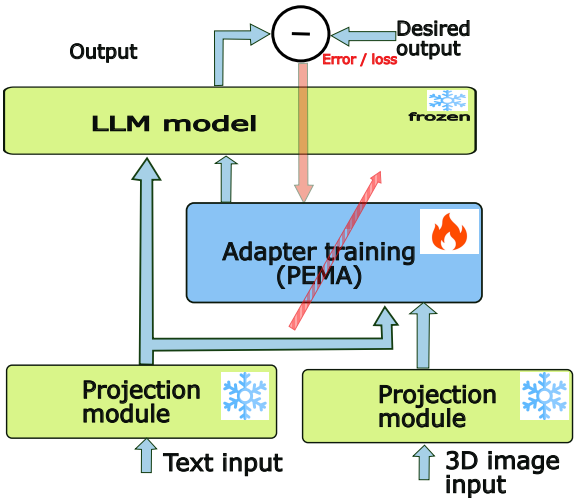
PEMA first maps the visual and prompt features to a shared hidden dimension h:

$$\boldsymbol{H}_v = V_i \boldsymbol{W}_v^\top + \boldsymbol{b}_v, \qquad \boldsymbol{H}_t = \boldsymbol{T} \boldsymbol{W}_t^\top + \boldsymbol{b}_t. \tag{3}$$

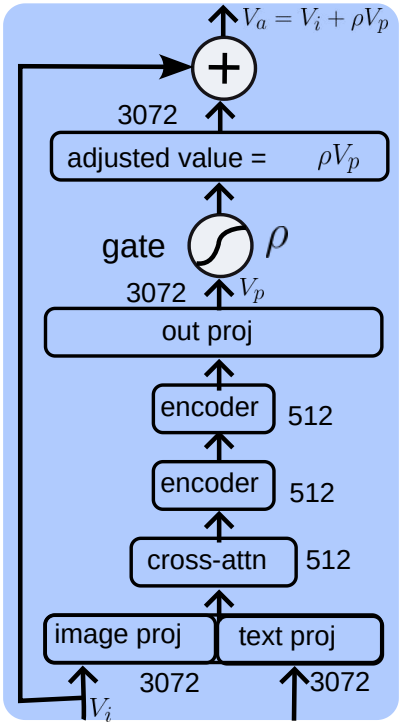
Here $\boldsymbol{W}_v \in \mathbb{R}^{h \times d_v}$ and $\boldsymbol{W}_t \in \mathbb{R}^{h \times d_t}$ are learned projection matrices, and $\boldsymbol{b}_v, \boldsymbol{b}_t \in \mathbb{R}^h$ are their learned bias vectors. The superscript $\top$ denotes matrix transpose. Each projection acts on the last feature dimension, and the same bias is added at every token position in every example. The



(a) Inference with the trained adapter.



(b) Training the adapter with the MLLM frozen.



(c) PEMA residual adapter.

Figure 1: PEMA system and adapter. The system diagrams summarize the frozen text-embedding and visual-encoding pathways as projection modules. Training updates only PEMA, including its gate; gradients still pass through the frozen LLM. Inside PEMA, the input tokens V_i receive the gated residual ρV_p, giving $V_a = V_i + \rho V_p$. Panel (c) shows Phi-3 dimensions: 3072-dimensional inputs and outputs with hidden dimension 512. LLaMA uses 4096-dimensional inputs and outputs.

resulting visual features $\boldsymbol{H}_v \in \mathbb{R}^{B\times N_v\times h}$ and prompt features $\boldsymbol{H}_t \in \mathbb{R}^{B\times N_t\times h}$ enter multi-head cross-attention (MHA) (Vaswani et al., 2017):

$$\boldsymbol{H}_c = \mathrm{MHA}(\boldsymbol{H}_v, \boldsymbol{H}_t, \boldsymbol{H}_t). \tag{4}$$

The three arguments supply queries, keys, and values, respectively. Thus, visual features query the prompt features, producing the question-conditioned output $\boldsymbol{H}_c \in \mathbb{R}^{B\times N_v\times h}$. Padding tokens in the prompt are masked. The function $\mathrm{Encoder}_L$ is a stack of L transformer encoder layers with pre-layer normalization. It processes the sum of the visual and cross-attention features, after which an output projection returns to the original visual-token dimension:

$$\boldsymbol{H} = \mathrm{Encoder}_L(\boldsymbol{H}_v + \boldsymbol{H}_c), \qquad V_p = \boldsymbol{H}\boldsymbol{W}_o^\top + \boldsymbol{b}_o. \tag{5}$$

Here $\boldsymbol{H} \in \mathbb{R}^{B\times N_v\times h}$ is the encoder output, $\boldsymbol{W}_o \in \mathbb{R}^{d_v\times h}$ is the learned output-projection matrix, and $\boldsymbol{b}_o \in \mathbb{R}^{d_v}$ is its learned bias vector, shared across examples and token positions. The residual $V_p \in \mathbb{R}^{B\times N_v\times d_v}$ is the correction predicted by PEMA, matching the output of the "out proj" block in

Figure 1(c). The default configuration uses $h = 512$, $L = 2$, eight attention heads, a $4h$ feed-forward dimension, GELU activations (Hendrycks & Gimpel, 2016), and dropout 0.1.

A learned scalar logit $\alpha \in \mathbb{R}$ sets the gate that scales the residual update:

$$\rho = \sigma(\alpha), \qquad V_a = V_i + \rho V_p. \tag{6}$$

The function σ is the sigmoid, so $\rho = 1/(1+\exp(-\alpha)) \in (0, 1)$ is a scalar shared across all examples and visual tokens. The adapted tokens $V_a \in \mathbb{R}^{B\times N_v\times d_v}$ have the same shape as the input tokens V_i. The diagram's "adjusted value" is the gated residual ρV_p, which is added to V_i rather than replacing it. Only the adapter weights and α are optimized. We initialize $\alpha = -3$, giving $\rho \approx 0.047$. A small gate does not necessarily mean a small change in the tokens, because the adapter can produce a large residual. We therefore measure the actual update ρV_p relative to the input tokens V_i, rather than interpret the gate alone.

The adapted tokens V_a replace the original visual tokens in the language-model input. All base-model weights remain frozen. However, training still requires gradients through the language model to update PEMA. Freezing the model removes its parameter gradients and optimizer states, but not the memory needed for intermediate activations.

3.3 Answer scoring and training

We append each candidate answer c_k to the prompt p and compute its average negative log likelihood (NLL), denoted by ℓ_k:

$$\ell_k = -\frac{1}{|\mathcal{A}_k|}\sum_{j\in\mathcal{A}_k} \log p_{\Phi,\Theta}(x_{k,j} \mid x_{k,:<j}, I). \tag{7}$$

Here x_k contains the prompt and candidate c_k; j is a token position, $x_{k,j}$ its token, and $x_{k,<j}$ the preceding tokens. The function $p_{\Phi,\Theta}$ denotes conditional next-token probability, distinct from prompt p, and $\log$ is the natural logarithm. The set $\mathcal{A}_k$ contains candidate-answer positions and the appended end token when present; $|\mathcal{A}_k|$ counts these scored tokens. Prompt positions are excluded. Averaging accounts for different candidate lengths. We define $z_k = -\ell_k$ and train with cross-entropy over the K scores and target y, as shown in Figure 1(b). Evaluation ranks all supplied candidates by average NLL and selects the lowest. Each score uses the candidate's own preceding tokens, not tokens generated by the model. This is complete-candidate likelihood ranking, not token-by-token greedy decoding, beam search, or sampling. All adaptation methods use the same objective and candidate formatting.

4 Experimental Setup

We also compare PEMA with LoRA using the same task subsets, base learning rate, batch and accumulation schedule, number of epochs, objective, and checkpoint-selection rule, with method-specific settings described below.

4.1 Dataset and preprocessing

We evaluate three closed-ended 3D-RAD tasks: Task 4 – Existence Detection; Task 5 – Static Temporal Diagnosis; and Task 6 – Longitudinal Temporal Diagnosis (Gai et al., 2025). Task 4 has two answer choices. Tasks 5 and 6 have up to four choices describing lesion status. Task 5 asks about the current CT without a history in the question. Task 6 supplies a textual sequence of previous presence/absence observations and asks about the current CT. Its choices distinguish refractory, resolved, new, and always-absent findings. The model receives one current volume, not a sequence of prior image volumes.

CT intensities are clipped to $[-1000, 1000]$ Hounsfield units, mapped to $[0, 1]$, and resized to single-channel $32 \times 256 \times 256$ arrays. We reconstructed 1,774 training arrays from CT-RATE NIfTI files: the 1,000-volume Task 4 pool and the 774-volume Task 5/6 pool (Hamamci et al., 2024). All methods use the same prepared official test arrays.

We split the training data by patient so that no patient appears in both training and validation. The split contains 16,992 training and 1,008 validation rows for Task 4, and 12,155 training and 1,003

validation rows for each of Tasks 5 and 6. To limit training cost, our main PEMA and LoRA experiments use fixed subsets: 2,000 training and 400 validation rows for Task 4, and 6,000 training and 1,000 validation rows for each of Tasks 5 and 6. We do not claim that these are the minimum data needed for adaptation. Testing uses all 23,472 Task 4 rows and all 2,873 rows for each of Tasks 5 and 6.

4.2 Models and baselines

Original and fine-tuned models. We use M3D-LaMed-Phi-3-4B (Abdin et al., 2024) and M3D-LaMed-LLaMA-2-7B (Touvron et al., 2023). The original models receive no task-specific training. PEMA and LoRA are trained separately for each task. For the fine-tuned reference, we evaluate the released M3D-RAD Phi-3 checkpoint, which is one model trained across tasks by its authors. Published results for both fine-tuned backbones are also listed in Appendix Table 5; published scores are not mixed with locally evaluated results.

The released model was trained with different data and optimization settings. Its public code supports both LoRA and updates to selected model weights, but the checkpoint does not specify exactly which parameters were trained. We use it as an accuracy reference, not as a controlled full-model fine-tuning experiment. We have not measured its training cost.

LoRA baseline. For a frozen LLM weight matrix $\boldsymbol{W}_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and an input vector $\boldsymbol{u} \in \mathbb{R}^{d_{\text{in}}}$, LoRA adds the update $(\alpha_{\text{LoRA}}/r)\boldsymbol{B}_{\text{L}}\boldsymbol{A}_{\text{L}}\boldsymbol{u}$ to $\boldsymbol{W}_0\boldsymbol{u}$ (Hu et al., 2022). Here d_{in} and d_{out} are the layer's input and output dimensions, r is the LoRA rank, and α_{LoRA} is its scaling parameter. The learned matrices are $\boldsymbol{A}_{\text{L}} \in \mathbb{R}^{r \times d_{\text{in}}}$ and $\boldsymbol{B}_{\text{L}} \in \mathbb{R}^{d_{\text{out}} \times r}$; the subscript L distinguishes these LoRA matrices from the batch size B. The low-rank branch is added in parallel with the original transformation, while $\boldsymbol{W}_0$ remains frozen. We use $r = 8$ and $\alpha_{\text{LoRA}} = 16$, which is separate from PEMA's gate logit α. LoRA is applied to attention and feed-forward projections inside the language model; its trainable matrices form Θ for this baseline. The target-module lists are given in the appendix.

The difference is where the learned update is applied: PEMA changes the visual-token input to the LLM, while LoRA changes selected transformations inside it. Both keep the original MLLM weights frozen.

Shared evaluation. All main comparisons use the answer-scoring rule in Section 3.3 and the same official test rows. PEMA and LoRA also use the training objective defined in that section. The additional image tests score candidates individually, whereas the main reference evaluation batches them together. These implementations can produce different predictions in bfloat16, so we report their reference scores separately in Appendix Table 4.

4.3 Training

PEMA and LoRA use AdamW (Loshchilov & Hutter, 2019) with learning rate 2×10^{-4}, batch size one, four-step gradient accumulation, no weight decay, gradient clipping at 1.0, bfloat16 precision, and three epochs. PEMA's gate uses learning rate 10^{-3}, fixed across tasks. LoRA uses dropout 0.05 and gradient checkpointing. Appendix A lists the complete settings.

We select the checkpoint with the highest validation row accuracy, using a rule fixed before test evaluation. This can favor frequent answers when classes are imbalanced. We therefore also report balanced accuracy and answer counts, and examine an alternative checkpoint rule in Appendix Table 27. All runs use one NVIDIA RTX 6000 Ada GPU, without concurrent training or evaluation jobs.

4.4 Evaluation metrics

Our main measure is *subtask-macro accuracy*: we calculate accuracy for each subtask, then take their unweighted mean, following the benchmark's presentation. We also call this *benchmark accuracy*. When reporting a mean across Tasks 4–6, we give each task equal weight. Row accuracy instead gives each question equal weight. Balanced accuracy averages recall over answer classes, making poor performance on less frequent answers easier to identify. Predicted-answer counts show whether a model ever selects each available answer.

Table 1: Benchmark accuracy (%), the unweighted mean of subtask accuracies. PEMA and LoRA values are mean ± sample SD. Original denotes unadapted M3D-LaMed; Our validation denotes the locally evaluated fine-tuned M3D-RAD checkpoint. A corresponding LLaMA checkpoint was not evaluated locally. Published fine-tuned results are listed separately in Table 5.

Model	Task	Original	PEMA	LoRA	Our validation
M3D (Phi3-4B)	4	40.43	81.46 ± 0.64	80.56 ± 2.52	82.42
M3D (Phi3-4B)	5	25.11	47.01 ± 1.33	47.16 ± 0.99	49.59
M3D (Phi3-4B)	6	24.04	73.74 ± 2.74	72.38 ± 0.54	74.36
M3D (LLaMA2-7B)	4	18.00	81.95 ± 0.10	82.01 ± 0.00	–
M3D (LLaMA2-7B)	5	26.72	48.78 ± 2.18	46.09 ± 0.00	–
M3D (LLaMA2-7B)	6	25.50	72.64 ± 0.90	61.59 ± 13.43	–

For trained methods, tables report mean accuracy and sample standard deviation. We report paired confidence intervals for PEMA–LoRA differences and use patient-level bootstrap intervals to assess test-sampling uncertainty. Statistical details are given in Appendix M.

Appendix Table 6 reports the fraction of the fine-tuned model's improvement obtained by PEMA. We also compare with simple predictors based on answer frequency and question text. Their definitions and data sources are given in the appendix.

4.5 Additional tests

We also test PEMA without question conditioning, training only the existing projector, and combining PEMA with LoRA. These additional comparisons are reported in the appendix.

We test whether predictions depend on the image by evaluating trained models with the correct volume, a shuffled volume from another patient, or an all-zero volume. Questions and target answers stay unchanged. Image shuffling does not use answer labels. Zeroing takes place before the vision encoder; the resulting features need not be zero, but cannot contain information about the example's actual image.

For Phi-3 Tasks 5 and 6, we also train, select, and test PEMA using only zero volumes, while retaining the questions and answer choices. Separate tests replace PEMA's residual with its training-set mean or learn a constant update directly. These follow-up experiments were designed after the main comparison, with their evaluation settings fixed before testing. Details are in Appendices F and G.

5 Results

Table 1 compares the original models, PEMA, LoRA, and the released fine-tuned Phi-3 checkpoint. We first examine the accuracy improvement from training PEMA, then compare it with LoRA and report the training cost.

5.1 Accuracy with PEMA

The original Phi-3 model obtains 29.86% mean benchmark accuracy across Tasks 4–6. The released fine-tuned model reaches 68.79%, showing the large improvement possible with task-specific training. Training only PEMA raises accuracy to 67.40%, close to the fine-tuned reference while keeping the original MLLM unchanged.

PEMA obtains 81.46 ± 0.64, 47.01 ± 1.33, and $73.74 \pm 2.74\%$ on Tasks 4–6, respectively. These scores are 0.96, 2.57, and 0.63 percentage points below the released reference, calculated from unrounded values. The improvement is therefore present on each task, rather than being driven by a single task in the average.

The LLaMA results show a similar improvement over the original model. PEMA reaches 81.95 ± 0.10, 48.78 ± 2.18, and $72.64 \pm 0.90\%$ on Tasks 4–6. For context, the 3D-RAD paper reports 81.09%, 51.20%, and 74.78% for its fine-tuned LLaMA model (Gai et al., 2025). These published scores are

not a locally evaluated reference under our scoring implementation, so we report them separately from the main comparison.

These results show that a small adapter can obtain much of the accuracy improvement associated with fine-tuning. They do not establish statistical equivalence, and the released models were not trained under the same conditions as PEMA. Appendix Table 6 reports the fraction of the improvement obtained by PEMA for both benchmark and balanced accuracy.

5.2 Comparison with LoRA

On Phi-3, PEMA and rank-8 LoRA differ by at most 1.36 percentage points in mean benchmark accuracy. PEMA is higher on LLaMA Tasks 5 and 6, but the Task 6 LoRA results vary substantially: $61.59 \pm 13.43\%$, compared with $72.64 \pm 0.90\%$ for PEMA. All six paired confidence intervals for benchmark accuracy include zero. Thus, the observed differences do not show a consistent advantage over LoRA. The balanced-accuracy comparison does favor PEMA on LLaMA Task 5 (Appendix Table 9).

Additional LLaMA Task 6 experiments show that both methods can be unstable (Appendix Table 28). Overall, PEMA is competitive with LoRA, but the results do not establish that sequential adapters are generally better than parallel adapters.

5.3 Parameters, training time, and memory

PEMA trains 12.08M parameters with Phi-3 and 13.65M with LLaMA, or 0.297% and 0.195% of the combined models. LoRA trains 12.58M and 19.99M parameters, respectively (Appendix Table 7). Both methods allow each task to store only its added parameters while sharing the pretrained model.

PEMA trains 1.71–1.90 times faster than LoRA in our implementations (Appendix Table 10). This result depends on the training configuration: LoRA uses gradient checkpointing and smaller candidate batches. PEMA's peak GPU memory use is higher, despite its smaller trainable parameter count (Appendix Table 15). The timing comparison is with LoRA; we did not measure the training cost of the released fine-tuned model.

5.4 Answer imbalance and image use

The accuracy gains need to be interpreted alongside the answer distribution. For Task 4, always predicting B gives 82.01% accuracy. PEMA's Phi-3 balanced accuracy is only $51.76 \pm 1.55\%$, and its accuracy on a balanced test subset remains near chance. The high Task 4 score therefore does not demonstrate reliable detection of findings.

On Phi-3 Tasks 5 and 6, PEMA trained and tested with zero volumes obtains 47.84% and 73.73% accuracy, close to the 47.01% and 73.74% obtained with normal images. Task 5 shows a larger difference in balanced accuracy: normal PEMA is 5.51 points higher on average, although the 95% interval is wide, $[-0.05, 11.08]$. Image information may help distinguish answer classes even when its effect on benchmark accuracy is small.

The similarity on Task 6 is not only in aggregate accuracy. For two of the three training seeds, normal PEMA and zero-trained PEMA agree on whether a test row is correct for approximately 98% and 91% of rows, respectively (Appendix Table 23). Shuffling the images changes only 2.87% of PEMA's Task 6 predictions (Appendix F).

A simpler adapter also performs well on Task 6. A separately trained constant residual with 3,072 parameters reaches 72.18% accuracy, compared with 73.74% for PEMA. Replacing PEMA's example-specific residual with its training-set mean gives $74.44 \pm 2.29\%$ accuracy and 97.54% prediction agreement with PEMA. Keeping only the example-varying component after subtracting this mean reduces accuracy to $30.20 \pm 2.56\%$. The shared residual accounts for 92.83–96.65% of residual energy across seeds. These results show that much of PEMA's Task 6 behavior can be reproduced by an update shared across examples, although the underlying frozen visual tokens remain image-dependent.

6 Discussion and Limitations

The main finding is that training a small adapter can substantially improve question answering without changing the pretrained MLLM. PEMA approaches the released fine-tuned Phi-3 model's accuracy and also improves the LLaMA model. This is useful when a shared model needs to support several tasks: each task can have its own adapter without storing another copy of the base weights.

The comparison with LoRA shows that changing visual tokens before the language model is a useful alternative to adding low-rank updates inside it. PEMA is smaller and faster in the tested configurations, but it is not consistently more accurate and uses more peak GPU memory. Fewer trainable parameters should therefore not be taken to mean lower memory use or better performance in every setting.

There are two main limits to the accuracy comparison. First, the released M3D-RAD checkpoint is a single model trained across tasks by its authors, whereas PEMA uses separate task-specific adapters and smaller data subsets. We cannot attribute every score difference to the adaptation method or use the reference to claim a measured saving over full fine-tuning. Second, high accuracy on these tasks is not the same as reliable image interpretation. Answer imbalance and strong results with zero volumes show that models can answer many questions without information from the actual scan.

The simpler Task 6 adapters also affect how we interpret PEMA's design. A constant update can approach PEMA's accuracy, so these results do not show that its attention layers or full parameter count are necessary. However, a constant residual is still added to image-dependent base tokens. Its success alone does not show that the complete model ignores images; the separate zero-volume tests address that question.

The study covers two backbones and three closed-ended tasks from one benchmark. The detailed Task 5/6 image tests use Phi-3, and the residual decomposition is limited to Phi-3 Task 6. Some historical epoch checkpoints were not retained, preventing complete test evaluation under alternative checkpoint-selection rules. Further work should test other datasets, open-ended answers, and localized image changes before drawing broader conclusions about medical image understanding.

7 Conclusion

We introduced PEMA, an adapter that improves 3D medical question answering while keeping the pretrained MLLM frozen. Placed between the multimodal projector and the language model, it uses the question to modify the projected visual tokens. On Phi-3, training 12.08M parameters raises mean benchmark accuracy from 29.86% to 67.40%, close to the released fine-tuned model's 68.79%. PEMA also improves LLaMA and is competitive with rank-8 LoRA, with fewer trainable parameters and faster training in our implementations, but no peak-memory advantage.

These results support training a small adapter as an alternative to updating the pretrained MLLM for the evaluated tasks. Answer imbalance, strong performance without images, and the success of simpler adapters limit the claim to benchmark accuracy rather than reliable medical image understanding.

Reproducibility Statement

Section 4 describes the data split, preprocessing, training, checkpoint selection, and evaluation metrics. Appendices A and M give the full settings and statistical procedures. Appendix N describes the saved file hashes, training arguments, software versions, and predictions used to check the results. The anonymized supplement will include the training, evaluation, and analysis code needed to reproduce the experiments.

Ethics Statement

This work uses de-identified public research datasets and released model checkpoints. We do not collect new patient data or use model outputs for patient care. The experiments measure performance on closed-ended benchmark questions, not clinical safety or readiness for deployment. We report

answer imbalance and results without images because high accuracy can otherwise give a misleading impression of medical image understanding.

AI USE STATEMENT

Generative AI tools assisted with experimental setup implementation, test development, review of experiment scripts and results, and creating an initial draft and critiquing the manuscript. They were not used to generate or modify medical images, questions, answer labels, or model predictions. The authors reviewed the assisted work, ran automated and end-to-end checks, and verified the results against saved predictions. The authors take responsibility for the paper, code, and results.

REFERENCES

Marah Abdin et al. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4971–4980, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Agrawal_Dont_Just_Assume_CVPR_2018_paper.html.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/960a172bc7fbf0177ccccbb411a7d800-Abstract-Conference.html.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2425–2433, 2015. URL https://arxiv.org/abs/1505.00468.

Fan Bai, Yuxin Du, Tiejun Huang, Max Q.-H. Meng, and Bo Zhao. M3D: Advancing 3d medical image analysis with multi-modal large language models, 2024. URL https://arxiv.org/abs/2404.00578.

Sayna Ebrahimi, Sercan O. Arik, Tejas Nama, and Tomas Pfister. CROME: Cross-modal adapters for efficient multimodal LLM, 2024. URL https://arxiv.org/abs/2408.06610.

Xiaotang Gai, Jiaxiang Liu, Yichen Li, Zijie Meng, Jian Wu, and Zuozhu Liu. 3D-RAD: A comprehensive 3d radiology med-vqa dataset with multi-temporal analysis and diverse diagnostic tasks, 2025. URL https://arxiv.org/abs/2506.11147.

Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. LLaMA-Adapter V2: Parameter-efficient visual instruction model, 2023. URL https://arxiv.org/abs/2304.15010.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Goyal_Making_the_v_CVPR_2017_paper.html.

Ibrahim Ethem Hamamci, Sezgin Er, Furkan Almas, Ayse Gulnihan Simsek, Sevval Nil Esirgun, Irem Dogan, Muhammed Furkan Dasdelen, Omer Faruk Durugol, Bastian Wittmann, Tamaz Amiranashvili, et al. Developing generalist foundation models from a multimodal dataset for 3d computed tomography, 2024. URL https://arxiv.org/abs/2403.17834.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs), 2016. URL https://arxiv.org/abs/1606.08415.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2790–2799, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

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

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision – ECCV 2022*, 2022. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/4175_ECCV_2022_paper.php.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021. URL https://aclanthology.org/2021.emnlp-main.243.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 19730–19742, 2023. URL https://proceedings.mlr.press/v202/li23q.html.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 4582–4597, 2021. URL https://aclanthology.org/2021.acl-long.353.

Zhihong Lin, Donghao Zhang, Qingyi Tao, Danli Shi, Gholamreza Haffari, Qi Wu, Mingguang He, and Zongyuan Ge. Medical visual question answering: A survey, 2021. URL https://arxiv.org/abs/2111.10056.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. LST: Ladder side-tuning for parameter and memory efficient transfer learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022a. URL https://papers.nips.cc/paper_files/paper/2022/hash/54801e196796134a2b0ae5e8adef502f-Abstract-Conference.html.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. VL-Adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5227–5237, 2022b. URL https://openaccess.thecvf.com/content/CVPR2022/html/Sung_VL-Adapter_Parameter-Efficient_Transfer_Learning_for_Vision-and-Language_Tasks_CVPR_2022_paper.html.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Jeffrey O. Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. Side-tuning: A baseline for network adaptation via additive side networks. In *Computer Vision – ECCV 2020*, volume 12348, pp. 698–714, 2020. URL https://arxiv.org/abs/1912.13503.

A Detailed Configuration

Table 2 lists the settings used in the main training experiments. The patient-disjoint split uses seed 2027 and remains fixed across methods. Training configurations use seeds 2027, 2028, and 2029 to vary initialization and training randomness without changing the training or validation examples. The fivefold gate learning-rate multiplier offsets the small sigmoid gradient near initialization; it was fixed across tasks, with no gate weight decay.

Table 2: Training settings for the main comparison.

Setting	PEMA / visual adapter	LoRA
Epochs	3	3
Physical batch / accumulation	1 / 4	1 / 4
Learning rate	2×10^{-4}	2×10^{-4}
Gate learning rate	10^{-3}	–
Weight decay	0	0
Maximum sequence length	512	512
Candidate score	normalized NLL	normalized NLL
Answer style	label and text	label and text
Class weighting / sampling	none / none	none / none
Checkpoint criterion	validation accuracy	validation accuracy
Precision	bfloat16	bfloat16
Hidden dimension / layers / heads	512 / 2 / 8	–
Gate initialization	$\alpha = -3$	–
Rank / scale / dropout	–	8 / 16 / 0.05
Training seeds	2027, 2028, 2029	2027, 2028, 2029

B Frozen and Released Checkpoint References

Table 3 reports the exact locally evaluated references used to construct Table 6. For external context, Table 5 reproduces the closed-ended accuracies reported by the 3D-RAD authors. Published values are not mixed with our paired uncertainty analyses.

B.1 Fraction of the Fine-Tuned Model's Improvement

For completeness, we report the fraction of the released checkpoint's improvement over the original model obtained by PEMA:

$$R_{\text{gain}} = \frac{A_{\text{PEMA}} - A_{\text{original}}}{A_{\text{released}} - A_{\text{original}}}. \tag{8}$$

Here A is benchmark accuracy. The same calculation using balanced accuracy shows how the comparison changes when answer classes receive equal weight. This is a descriptive ratio, not a test of equivalence. We hold the original and released scores fixed, calculate the ratio for each PEMA training run, and report a Student-t interval across the three runs.

The constant-answer reference in Table 6 uses the most frequent answer in the test labels. It describes the test distribution and is not a predictor selected from training data. The majority and question-lookup predictors in Appendix F use only training labels. Both majority rules select B for Task 4 and D for Tasks 5 and 6, but their data sources differ.

C Detailed PEFT Comparisons

This appendix gives the detailed PEMA–LoRA results, confidence intervals, training times, and comparisons with other adapter configurations.

Table 3: Locally evaluated references under the same candidate-NLL scoring rule and official test manifests as the adaptation runs.

Reference	Task	Subtask-macro	Row	Balanced
M3D-LaMed-Phi-3-4B (frozen)	Task 4	40.43	40.43	46.56
M3D-LaMed-Phi-3-4B (frozen)	Task 5	25.11	14.27	25.10
M3D-LaMed-Phi-3-4B (frozen)	Task 6	24.04	15.42	26.55
M3D-LaMed-LLaMA-2-7B (frozen)	Task 4	18.00	18.00	50.01
M3D-LaMed-LLaMA-2-7B (frozen)	Task 5	26.72	14.90	25.94
M3D-LaMed-LLaMA-2-7B (frozen)	Task 6	25.50	14.72	26.28
M3D-RAD-Phi-3-4B (released)	Task 4	82.42	82.42	56.12
M3D-RAD-Phi-3-4B (released)	Task 5	49.59	59.97	35.62
M3D-RAD-Phi-3-4B (released)	Task 6	74.36	80.89	55.31

Table 4: Primary archived reference evaluation versus the later single-image audit. Both use normalized candidate NLL, but different reference forward paths yield different scores. The primary three-task recovery table retains its original denominators; mechanism and image-audit tables use only the latter references. Values must not be interchanged.

Task	Model	Primary macro	Primary bal.	Audit macro	Audit bal.
5	Frozen	25.11	25.10	25.45	24.90
5	Released	49.59	35.62	47.97	35.50
6	Frozen	24.04	26.55	24.10	26.05
6	Released	74.36	55.31	75.46	55.82

D Question Conditioning and Adapter Components

LoRA targets fused `qkv_proj`, `o_proj`, `gate_up_proj`, and `down_proj` layers on Phi-3, and separate query, key, value, output, gate, up, and down projections on LLaMA. The visual-only adapter removes PEMA's prompt cross-attention. A separate test keeps a trained PEMA fixed and changes only the prompt features supplied to its cross-attention layer. The LLM still receives the correct question.

E Projector Training and GPU Memory

The projector-only Task 6 experiment trains the existing projector's 11.80M parameters using the primary subsets, objective, three epochs, and validation-accuracy rule. The original seed-2027 run was replicated with seeds 2028/2029 and all three are reported in Table 19, superseding the exploratory one-seed estimate.

The one-step profiles use the same four training rows, batch and accumulation, objective, and bfloat16 precision. Primary LoRA retains one-candidate chunks and gradient checkpointing; the matched variants checkpoint PEMA and score all four LoRA candidates together. PEMA checkpointing calls Hugging Face `gradient_checkpointing_enable` on the frozen `LamedPhi3ForCausalLM`, which activates recomputation in the inherited Phi-3 decoder stack. It does not checkpoint PEMA or the no-gradient vision path. Consequently, it reduces saved decoder activations but not frozen weights, adapter tensors, candidate representations, logits, or transient kernel workspace, consistent with the measured 19.74-to-18.27 GiB reduction. The profile does not estimate full-run variance or full-fine-tuning memory.

The additional 32-row profile starts from trained checkpoints after four warm-up rows. It complements the older one-step profile; differing candidate chunk sizes and checkpointing settings remain explicit rather than pooled.

F Additional Adapter and Image Tests

These experiments investigate the trained adapter and its dependence on images. They use the same saved Phi-3 checkpoints and test rows, with candidates scored individually through one common

Table 5: Accuracy (%) reported in the 3D-RAD paper. M3D-RAD denotes the authors' adapted model. These are external published values, not results from our local evaluation.

Published model	Task 4	Task 5	Task 6
M3D (LLaMA2-7B)	18.00	25.47	24.17
M3D-RAD (LLaMA2-7B)	81.09	51.20	74.78
M3D (Phi3-4B)	40.25	25.40	24.31
M3D-RAD (Phi3-4B)	82.43	49.30	74.77

Table 6: Phi-3 recovery in the primary archived candidate-NLL evaluation. "Majority" is the constant global-majority predictor; PEMA is mean ± SD over three seeds. Recovery is $(\mathrm{PEMA} - \mathrm{Frozen})/(\mathrm{Released\ FT} - \mathrm{Frozen})$ with 95% training-seed intervals. The released model used its authors' all-task training protocol. Later single-image audit references are reported separately in Table 4.

Subtask-macro accuracy (%)

Task	Frozen	Majority	PEMA	Released FT	Recovery [95% CI]
Task 4	40.43	82.01	81.46 ± 0.64	82.42	97.7 [94.0, 101.5]
Task 5	25.11	46.09	47.01 ± 1.33	49.59	89.5 [76.0, 103.0]
Task 6	24.04	46.09	73.74 ± 2.74	74.36	98.8 [85.2, 112.3]
Mean	29.86	58.06	67.40 ± 1.04	68.79	96.4 [89.8, 103.1]

Balanced accuracy (%)

Task	Frozen	Majority	PEMA	Released FT	Recovery [95% CI]
Task 4	46.56	50.00	51.76 ± 1.55	56.12	54.4 [14.2, 94.6]
Task 5	25.10	25.00	33.55 ± 0.58	35.62	80.3 [66.7, 93.9]
Task 6	26.55	25.00	53.91 ± 1.34	55.31	95.1 [83.6, 106.7]
Mean	32.74	33.33	46.41 ± 0.83	49.02	84.0 [71.3, 96.6]

evaluator. They provide additional analysis of the main results and are not used to select checkpoints for that comparison.

Image shuffling assigns a volume from a different patient to each row without using answer labels. Each source row's image is used once in the permutation. Question text and target answers remain fixed. No-image models are trained, validated, and tested with zero volumes, with the earliest epoch retained when validation row accuracy ties. The original question remains available to the language model even when PEMA's question input is changed or removed. The separate structured lookup groups the same 6,000 training rows by finding and history. We also examine the 838 test rows it answers incorrectly, but do not use them for model selection. The full-data lookup fits 12,155 training rows. Historical image-shuffling experiments with a different volume permutation are reported separately.

F.1 Replacing the Task 6 residual with its training mean

Table 16 compares the Phi-3 Task 6 shared-residual controls. Let $\boldsymbol{D}_j = \rho V_{p,j}$ be PEMA's effective residual for example j and $\overline{\boldsymbol{D}}$ its mean over the 6,000 training rows. The shared matrix accounts for 92.83–96.65% of total squared residual energy across seeds (Appendix Table 18). More importantly, replacing $\boldsymbol{D}_j$ with $\overline{\boldsymbol{D}}$ at inference yields $74.44 \pm 2.29\%$ macro accuracy versus PEMA's $73.74 \pm 2.74\%$, with 97.54% prediction agreement. Keeping only $\boldsymbol{D}_j - \overline{\boldsymbol{D}}$ reduces accuracy to $30.20 \pm 2.56\%$. Thus, the mean residual reproduces most of PEMA's predictions, while removing it causes a large accuracy drop. Subtraction changes the tokens passed to the LLM and can produce inputs unlike those seen during training. The result does not mean that the mean residual contains everything PEMA has learned.

A separately optimized shared vector, broadcast over visual-token positions, reaches $72.18 \pm 0.88\%$ macro and $52.57 \pm 0.60\%$ balanced accuracy with only 3,072 trainable values. Its 83.62% prediction agreement with PEMA is lower than the training-mean replacement's: comparable scores do not imply identical behavior. The extracted mean residual is not a training-free adapter, since it requires training PEMA first. Training only the existing projector, which has a parameter count similar to

Table 7: Trainable parameter counts for PEMA and the LoRA reference. Percentages use the total number of base-model and adapter parameters.

Frozen backbone	Method	Parameters	Fraction
M3D-LaMed-Phi-3-4B	PEMA	12.08M	0.297%
M3D-LaMed-Phi-3-4B	LoRA	12.58M	0.310%
M3D-LaMed-LLaMA-2-7B	PEMA	13.65M	0.195%
M3D-LaMed-LLaMA-2-7B	LoRA	19.99M	0.285%

Table 8: Fixed-protocol accuracy (%). Values are mean ± sample standard deviation across completed seeds. n reports completed PEMA/LoRA seeds. Δ is PEMA minus LoRA in mean subtask-macro accuracy, the primary metric; balanced accuracy is diagnostic.

Backbone	Task	Subtask-macro			Balanced		n
		PEMA	LoRA	Δ	PEMA	LoRA	P/L
M3D-LaMed-Phi-3-4B	Task 4	81.46 ± 0.64	80.56 ± 2.52	+0.91	51.76 ± 1.55	53.46 ± 5.99	3/3
M3D-LaMed-Phi-3-4B	Task 5	47.01 ± 1.33	47.16 ± 0.99	-0.15	33.55 ± 0.58	28.89 ± 3.38	3/3
M3D-LaMed-Phi-3-4B	Task 6	73.74 ± 2.74	72.38 ± 0.54	+1.36	53.91 ± 1.34	54.91 ± 0.55	3/3
M3D-LaMed-LLaMA-2-7B	Task 4	81.95 ± 0.10	82.01 ± 0.00	-0.06	50.63 ± 1.09	50.00 ± 0.00	3/3
M3D-LaMed-LLaMA-2-7B	Task 5	48.78 ± 2.18	46.09 ± 0.00	+2.69	30.35 ± 0.57	25.00 ± 0.00	3/3
M3D-LaMed-LLaMA-2-7B	Task 6	72.64 ± 0.90	61.59 ± 13.43	+11.06	53.56 ± 1.25	42.72 ± 15.44	3/3

PEMA, reaches $54.87 \pm 3.50\%$ macro accuracy over three seeds (Table 19). These results show that an added residual can be useful without requiring PEMA's full network or cross-attention. A constant residual still modifies image-dependent base tokens, so its success alone does not show that the complete model works without images.

F.2 Tests with shuffled images and zero volumes

Table 17 compares learned and simple no-image controls; Appendix Table 25 gives all image-intervention scores.

Task 5. A training-set majority predictor attains 46.09% macro accuracy versus 47.01% for normal PEMA; exact-prompt lookup obtains 47.18%. PEMA trained, selected, and tested entirely with zero volumes reaches $47.84 \pm 1.51\%$. However, its balanced accuracy is $28.04 \pm 2.63\%$, below normal PEMA's $33.55 \pm 0.58\%$: a 5.51-point difference. One zero-trained seed predicts only D and the others only A/D. Normal PEMA recovers some B recall in two seeds, although no PEMA seed predicts C (Table 21).

Changing the images while keeping each trained model fixed gives a similar pattern: normal PEMA exceeds shuffled and zero-image evaluation by 2.14 and 2.84 balanced-accuracy points, versus only 1.10 and 0.63 macro points. The paired patient-bootstrap intervals for the balanced effects are [0.96, 3.38] and [1.84, 3.84]. These intervals describe uncertainty from the test patients for the saved models. For the 5.51-point difference between normal training and zero-volume training, the patient interval is [4.38, 6.67], but the Student-t interval across training runs is [-0.05, 11.08] (Table 22). The wider interval reflects variation from training that the patient bootstrap does not measure. Images appear to help distinguish answer classes, but the evidence does not establish reliable image interpretation.

Task 6. Global majority and exact-prompt lookup obtain only 46.09% macro and 25.00% balanced accuracy. Learned no-image PEMA reaches 73.73% and 52.59%, close to normal PEMA's 73.74% and 53.91%. Normal-minus-shuffle and normal-minus-zero macro effects are -0.45 and +0.05 points. The learned model therefore does more than predict the most frequent answer. These results do not show that PEMA's cross-attention is necessary or identify which parts of the question the model uses. The LLM receives the original question, including textual history, in every condition. A distinct, structured finding/history lookup reaches 59.44% using the same training subset (Table 24). Even this simple lookup can use some of the information in the question without training a neural model.

Table 9: Paired PEMA-minus-LoRA differences across the three matched training seeds. Intervals are 95% Student-t intervals; positive values favor PEMA.

Backbone	Task	Δ macro	95% interval	Δ balanced	95% interval
M3D-LaMed-Phi-3-4B	Task 4	+0.91	[-6.59, +8.41]	-1.69	[-20.38, +16.99]
M3D-LaMed-Phi-3-4B	Task 5	-0.15	[-5.92, +5.62]	+4.66	[-3.67, +12.98]
M3D-LaMed-Phi-3-4B	Task 6	+1.36	[-6.76, +9.48]	-1.00	[-5.03, +3.03]
M3D-LaMed-LLaMA-2-7B	Task 4	-0.06	[-0.30, +0.19]	+0.63	[-2.08, +3.35]
M3D-LaMed-LLaMA-2-7B	Task 5	+2.69	[-2.73, +8.12]	+5.35	[+3.94, +6.75]
M3D-LaMed-LLaMA-2-7B	Task 6	+11.06	[-22.82, +44.94]	+10.84	[-26.76, +48.45]

Table 10: Mean wall-clock training time across three fixed-protocol seeds on one RTX 6000 Ada GPU. Speedup is LoRA time divided by PEMA time.

Backbone	Task	PEMA (min)	LoRA (min)	Speedup
Phi-3	4	22.06	38.67	1.75×
Phi-3	5	121.90	231.12	1.90×
Phi-3	6	123.87	232.67	1.88×
LLaMA	4	33.78	57.85	1.71×
LLaMA	5	193.97	344.68	1.78×
LLaMA	6	195.41	348.74	1.78×

Changes in individual predictions. The released checkpoint also has modest macro changes: normal-minus-shuffle and normal-minus-zero are -0.68/+2.61 points on Task 5 and +0.06/+1.86 on Task 6. Its Task 5 balanced normal-over-shuffle effect is +2.70 points. PEMA changes approximately 20.16% of Task 5 predictions under shuffling, versus 2.87% on Task 6; the released Task 6 model changes 6.47%. Changes from correct to incorrect answers can be offset by changes in the opposite direction, leaving overall accuracy nearly unchanged. Task 4 gives another example of why accuracy alone is insufficient: predicting only B obtains 82.01%, near PEMA's 81.46%, while its balanced diagnostic remains near chance (Appendix Figure 2). Together, these tests show that higher benchmark accuracy does not establish that the model interprets the image correctly.

G Shared and Constant Residuals

Write the effective update for training row j as $\boldsymbol{D}_j = \rho V_{p,j} \in \mathbb{R}^{N_v \times d_v}$, where $V_{p,j}$ is that example's predicted residual. Using only the n training rows, define

$$\overline{\boldsymbol{D}} = \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{D}_j, \qquad \boldsymbol{D}_j^{\text{vary}} = \boldsymbol{D}_j - \overline{\boldsymbol{D}}. \tag{9}$$

We either replace the residual with this mean or subtract the mean from it, leaving the base visual tokens unchanged. We do not rescale the replacement. Averaging over token positions also gives a single vector that can be added at every position. Computing these means uses no test rows or further training, but requires a trained PEMA model. Separate constant-vector and constant-matrix experiments test whether the update can be learned directly.

All Task 6 prototypes are constructed from effective residuals on the same 6,000 training rows, without validation or test examples. The matrix mean preserves token position; the broadcast vector additionally averages across positions. We apply replacement and subtraction at scale one, without test-selected scaling. Energy fractions use squared Frobenius norms, not mean token norms. They describe how much of the residual magnitude is shared across examples. The replacement and subtraction tests show how changing that shared component affects predictions.

Independently trained constant vectors and matrices use three epochs and the same candidate objective and data subsets. For each shape, learning rates $\{2 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}\}$ were compared using seed-2027 validation row accuracy only. Both selected 5×10^{-3}, then replicated that rate on seeds 2028 and 2029; the seed-2027 sweep winner is included in the three-seed summary. Epoch ties keep the earliest checkpoint and rate ties prefer the smaller rate. This tuning budget differs from primary PEMA's fixed rate. These tests examine simpler adapters; they do not compare methods under equal

Table 11: Adaptation-location controls under the fixed protocol. Values are mean subtask-macro accuracy (%) across three seeds. Δ_{cond} is PEMA minus the visual-only adapter; Δ_{hyb} is the hybrid minus the better of PEMA and LoRA.

Backbone	Task	Visual	PEMA	Δ_{cond}	LoRA	Hybrid	Δ_{hyb}
M3D-LaMed-Phi-3-4B	Task 4	81.98	81.46	-0.52	80.56	81.84	+0.37
M3D-LaMed-Phi-3-4B	Task 5	46.59	47.01	+0.42	47.16	49.17	+2.00
M3D-LaMed-Phi-3-4B	Task 6	73.37	73.74	+0.36	72.38	71.89	-1.84
M3D-LaMed-LLaMA-2-7B	Task 4	81.47	81.95	+0.49	82.01	82.01	+0.00
M3D-LaMed-LLaMA-2-7B	Task 5	46.82	48.78	+1.96	46.09	47.60	-1.18
M3D-LaMed-LLaMA-2-7B	Task 6	72.49	72.64	+0.15	61.59	54.48	-18.16

Table 12: Adapter component control. Values are subtask-macro accuracy (%), mean $\pm$ sample standard deviation over three seeds. Δ is PEMA minus the unconditioned visual-token adapter.

Backbone	Task	Visual-only	PEMA	Δ
M3D-LaMed-Phi-3-4B	Task 4	81.98 ± 0.04	81.46 ± 0.64	-0.52
M3D-LaMed-Phi-3-4B	Task 5	46.59 ± 1.12	47.01 ± 1.33	+0.42
M3D-LaMed-Phi-3-4B	Task 6	73.37 ± 1.71	73.74 ± 2.74	+0.36
M3D-LaMed-LLaMA-2-7B	Task 4	81.47 ± 0.92	81.95 ± 0.10	+0.49
M3D-LaMed-LLaMA-2-7B	Task 5	46.82 ± 1.90	48.78 ± 2.18	+1.96
M3D-LaMed-LLaMA-2-7B	Task 6	72.49 ± 0.54	72.64 ± 0.90	+0.15

training budgets. The vector has 3,072 parameters and the matrix 786,432. Extracted means store the same numbers of values, but require training the 12.08M-parameter PEMA first.

H Gate and Residual Measurements

Table 20 reports values from the training-log epoch corresponding to each validation-selected checkpoint. They characterize the size of the adapter update, not where the model attends or whether it uses the correct image features. For Phi-3 Task 6, the mean gate is about 0.044 but the effective residual/base norm ratio is 1.97. A small gate therefore does not imply a small change in the visual tokens: the residual magnitude can compensate for it.

I Training Without Image Information

This experiment replaces every volume with zeros during training, checkpoint selection, and testing. It keeps the same prompts, labels, answer scoring, data subsets, and random seeds.

The Task 5 control completes all three epochs before selecting epochs 1, 2, and 3 for seeds 2027, 2028, and 2029, respectively. Test results never select epochs. The identical Task 5 predictions of seeds 2028/2029 arise from distinct trained weight tensors, not reuse of a checkpoint. Table 21 reports counts and recall because similar overall accuracy can hide differences between answer classes. Table 22 separates test-patient uncertainty from variation across training runs.

I.1 Exact prompts versus structured textual history

The exact-prompt lookup matches the entire normalized question/choice prompt. The separate structured lookup extracts the named finding and binary history sequence, pooling differently worded prompts that share that key. Its 59.44% matched-subset and 63.23% full-training macro scores therefore do not contradict the exact-prompt lookup's 46.09%. The lookup-residual table uses the original archived scorer and a label-defined descriptive subset. On that subset, adapted models have similar macro scores to frozen inference but much higher balanced accuracy; the macro comparison alone cannot imply that adaptation is useless there.

Table 13: Inference-time intervention on the prompt features supplied only to PEMA cross-attention. The frozen LLM always receives the correct prompt. Values are subtask-macro accuracy (%) across three training seeds; Δ is the paired normal-minus-intervention mean with a 95% Student-t interval. Positive values indicate greater accuracy with the original prompt features.

Backbone	Task	Normal	Shuffled	Paired Δ [95% CI]	Zero	Paired Δ [95% CI]
M3D-LaMed-Phi-3-4B	Task 5	47.01 ± 1.33	43.99 ± 1.72	$+3.02$ [-4.54, $+10.58$]	43.60 ± 7.16	$+3.42$ [-17.42, $+24.25$]
M3D-LaMed-LLaMA-2-7B	Task 5	48.78 ± 2.18	43.48 ± 6.18	$+5.30$ [-8.28, $+18.87$]	46.35 ± 0.51	$+2.43$ [-3.03, $+7.89$]

Table 14: Peak memory in a one-step Phi-3 Task 6 profile. All rows use batch size one, four-way gradient accumulation, bfloat16, and the same objective. "Checkpointed" applies recomputation to the inherited Phi-3 decoder stack; "chunk 1" is LoRA's primary candidate-scoring configuration, while "unchunked" scores all four candidates together like PEMA. These measurements characterize one training step, not end-to-end runtime.

Method	CUDA allocated (GiB)	CUDA reserved (GiB)	Host RSS (GiB)
PEMA	19.74	20.29	15.05
PEMA (checkpointed)	18.27	18.79	15.05
LoRA (chunk 1)	8.82	9.21	15.05
LoRA (unchunked)	9.85	11.02	15.05

J Additional Image Test Results

Figure 2 shows historical per-seed Task 4 interventions. Table 26 reports the earlier full-data Task 5 control using unique-volume derangements and the original evaluator. These use different image permutations from the tests in Appendix F and are not pooled with them.

K Checkpoint-Selection Sensitivity

The primary protocol selects checkpoints by validation accuracy. Because that criterion may favor a frequent answer, we also identify the epoch that would have been selected by validation balanced accuracy without changing any test result or replacing the prespecified checkpoint.

Validation logs show which epoch each rule would select, even when that epoch's weights were not retained. Missing checkpoints prevent us from recomputing every test score under the alternative rule. We do not select checkpoints using test scores or treat missing evaluations as evidence of no difference.

K.1 Post-hoc LLaMA Task 6 stability

The prespecified LLaMA Task 6 runs motivated two additional seeds for both PEMA and LoRA. This post-hoc analysis measures whether the apparent method gap persists; it does not replace the main results.

L Complete Experimental Results

All 72 prespecified model–method–task–seed runs completed full-test evaluation. "Missing" in Table 29 is a strict diagnostic that counts a run when it never predicts at least one answer choice represented in the gold test labels.

M Statistical Reporting

Trained-method summaries report the mean and sample standard deviation across the three training runs. PEMA–LoRA differences are paired by training seed and reported with 95% Student-t intervals. Patient-bootstrap intervals use 2,000 resamples of patients, keeping all questions for each patient together. They measure test-sampling uncertainty for fixed trained models, not variation from training.

Table 15: Task 6 short training profiles: 32 rows after four warm-up rows, seed 2027, BF16, one RTX 6000 Ada. Temporary updates start from trained checkpoints and are discarded. These are configuration-dependent measurements, not compute-matched or full-run memory estimates.

Method	Candidate chunk	Checkpointing	Peak GiB	Rows/s
PEMA	4	No	19.87	2.71
LoRA	1	Yes	8.96	1.37
Projector	1	No	17.84	2.21
Shared vector	4	No	19.58	2.74

Table 16: Task 6 shared-residual controls under the single-image audit scorer. Scores are percentages; adapted methods use three seeds. Agreement is row-wise prediction agreement with the corresponding PEMA seed (references average over seeds). The trained vector optimizes 3,072 values. The training-mean matrix stores $256 \times 3{,}072$ values and inherits PEMA's 12.08M-parameter training; it is not a training-free method.

Method	Macro	Balanced	Agreement
Frozen	24.10	26.05	16.58
PEMA	73.74 ± 2.74	53.91 ± 1.34	100.00
Trained shared vector	72.18 ± 0.88	52.57 ± 0.60	83.62
Training-mean residual only	74.44 ± 2.29	53.75 ± 0.75	97.54
PEMA minus training mean	30.20 ± 2.56	27.03 ± 0.87	28.32
Released M3D-RAD	75.46	55.82	89.28

Table 30 reports uncertainty from resampling patients within each saved test prediction file. The training-seed intervals in Table 9 answer a different question and remain the appropriate basis for claims about optimization variability.

N Reproducibility and Artifact Integrity

The submission launcher constructs patient-disjoint manifests once, records their SHA-256 hashes, and reuses them for every method and training seed. Each checkpoint stores the resolved configuration, Git commit and dirty-tree state, package versions, source-file hashes, parameter summary, and optimizer metadata. Software random-number generators are seeded, but CUDA execution is not claimed to be bitwise reproducible. Evaluation writes row-level predictions before aggregate summaries. Statistical scripts reject duplicate prediction keys and pair methods by task, patient, question, and answer-choice identity. Internal provenance is retained unchanged. The anonymous artifact will instead use relative-path mappings and artifact-local hashes, omitting repository commit identifiers, usernames, hostnames, and absolute paths.

Table 17: Two nonvisual performance regimes on Tasks 5 and 6. Macro and balanced accuracy (%) use the single-image audit scorer; PEMA values are mean ± sample SD over three seeds. Both lookup and majority controls are fit on training labels only. Exact-prompt lookup is distinct from the structured finding/history lookup in Appendix Table 24.

Task	Method	Macro	Balanced
5	Training majority	46.09	25.00
5	Exact-prompt lookup	47.18	31.19
5	PEMA, normal images	47.01 ± 1.33	33.55 ± 0.58
5	PEMA, trained/tested zero	47.84 ± 1.51	28.04 ± 2.63
5	Released M3D-RAD	47.97	35.50
6	Training majority	46.09	25.00
6	Exact-prompt lookup	46.09	25.00
6	PEMA, normal images	73.74 ± 2.74	53.91 ± 1.34
6	PEMA, trained/tested zero	73.73 ± 1.27	52.59 ± 1.79
6	Released M3D-RAD	75.46	55.82

Table 18: Effective-residual geometry on the 6,000 training rows used to construct the prototypes. Energy fractions (%) are squared Frobenius norms of the shared component divided by total residual energy. The last column averages per-example mean-token residual/base-norm ratios.

Seed	Shared-vector energy	Shared-matrix energy	Norm ratio
2027	94.44	96.65	2.30
2028	86.23	93.64	1.75
2029	79.60	92.83	2.00

Table 19: Additional Task 6 mechanism controls (three seeds, single-image scorer, mean ± SD in %). The trained constant matrix uses 786,432 parameters; the native projector uses 11.80M. The training-mean vector stores 3,072 values but inherits source-PEMA training.

Control	Macro	Balanced
Trained constant matrix	68.27 ± 3.10	49.12 ± 1.50
Training-mean vector	74.64 ± 1.97	53.90 ± 0.30
PEMA minus mean vector	29.38 ± 4.54	27.64 ± 0.55
Native projector	54.87 ± 3.50	44.27 ± 2.19

Table 20: PEMA residual diagnostics at each validation-selected checkpoint. Values are mean ± sample standard deviation over three seeds. Norm ratios are averages recorded during the selected training epoch. Because the gate and residual are jointly optimized, neither quantity is independently identifiable.

Backbone	Task	Selected epochs	ρ	$\|\Delta V\|/\|V\|$	$\|\rho\Delta V\|/\|V\|$
M3D-LaMed-Phi-3-4B	Task 4	1,3,3	0.0470 ± 0.0026	25.63 ± 9.89	1.23 ± 0.52
M3D-LaMed-Phi-3-4B	Task 5	3,3,3	0.0428 ± 0.0045	41.53 ± 15.95	1.87 ± 0.88
M3D-LaMed-Phi-3-4B	Task 6	2,2,3	0.0436 ± 0.0006	43.79 ± 1.11	1.97 ± 0.02
M3D-LaMed-LLaMA-2-7B	Task 4	2,2,1	0.0470 ± 0.0002	24.00 ± 11.12	1.13 ± 0.52
M3D-LaMed-LLaMA-2-7B	Task 5	2,2,1	0.0451 ± 0.0017	39.26 ± 22.50	1.81 ± 1.06
M3D-LaMed-LLaMA-2-7B	Task 6	2,1,2	0.0454 ± 0.0010	47.39 ± 19.85	2.17 ± 0.89

Table 21: Task 5 prediction counts and per-class recall (%). Gold counts are A:426, B:393, C:212, D:1,842. The trained-zero control never predicts B or C; normal PEMA also never predicts C.

Model	Seed	A	B	C	D	Rec. A	Rec. B	Rec. C	Rec. D
PEMA normal	2027	480	778	0	1615	29.11	36.90	0.00	66.23
PEMA normal	2028	1044	0	0	1829	59.62	0.00	0.00	74.00
PEMA normal	2029	432	689	0	1752	29.58	35.88	0.00	71.28
Trained zero	2027	0	0	0	2873	0.00	0.00	0.00	100.00
Trained zero	2028	338	0	0	2535	24.41	0.00	0.00	93.81
Trained zero	2029	338	0	0	2535	24.41	0.00	0.00	93.81

Table 22: Paired effects in percentage points, with separate 95% intervals. Seed: Student-t over three matched training seeds. Patient: 2,000 joint patient-cluster bootstrap draws, conditional on the fitted checkpoints and averaged over seeds. These are different uncertainty targets; neither interval establishes equivalence.

Task	Contrast	Metric	Effect	Seed CI	Patient CI
5	Normal − shuffled	Macro	+1.10	[+0.41, +1.80]	[-1.49, +3.76]
5	Normal − shuffled	Balanced	+2.14	[+0.73, +3.56]	[+0.96, +3.38]
5	Normal − inference zero	Macro	+0.63	[-6.31, +7.58]	[-1.81, +2.87]
5	Normal − inference zero	Balanced	+2.84	[+0.51, +5.18]	[+1.84, +3.84]
5	Normal − trained zero	Macro	-0.82	[-3.26, +1.62]	[-5.16, +2.59]
5	Normal − trained zero	Balanced	+5.51	[-0.05, +11.08]	[+4.38, +6.67]
6	Normal − shuffled	Macro	-0.45	[-1.17, +0.26]	[-1.60, +0.64]
6	Normal − shuffled	Balanced	-0.72	[-3.03, +1.60]	[-1.42, -0.01]
6	Normal − inference zero	Macro	+0.05	[-5.31, +5.40]	[-1.22, +1.56]
6	Normal − inference zero	Balanced	-0.08	[-4.25, +4.09]	[-0.78, +0.63]
6	Normal − trained zero	Macro	+0.00	[-4.41, +4.42]	[-2.16, +1.77]
6	Normal − trained zero	Balanced	+1.32	[-1.38, +4.02]	[+0.18, +2.44]

Table 23: Paired Task 6 correctness for primary PEMA with normal images versus zero-trained PEMA with zero images. P-only and Z-only are discordant row counts. Effects are primary-minus-zero-trained row accuracy with 95% paired patient-cluster bootstrap intervals.

Seed	Both right	P-only	Z-only	Both wrong	Δ row [95% CI]
2027	2287	31	28	527	+0.10 [-0.45, +0.66]
2028	2203	137	121	412	+0.56 [-0.52, +1.71]
2029	1902	383	170	418	+7.41 [+5.46, +9.29]

Table 24: Task 6 text-prior audit. The parameter-free lookup memorizes majority labels for exact (finding, history) keys and globally defaults to D. "Residual" contains its 838 errors. PEMA is mean $\pm$ SD over three seeds.

Evaluation set	Method	Macro (%)	Row (%)	Balanced (%)
All test rows	Global training majority	46.09	64.11	25.00
All test rows	Exact lookup, 6,000 rows (28.7% coverage)	59.44	70.83	36.77
All test rows	Exact lookup, 12,155 rows (29.3% coverage)	63.23	71.28	37.60
All test rows	Primary PEMA	73.74 $\pm$ 2.74	80.55 $\pm$ 0.96	53.91 $\pm$ 1.34
All test rows	Zero-trained PEMA, zero image	73.73 $\pm$ 1.27	77.86 $\pm$ 4.98	52.59 $\pm$ 1.79
Lookup-residual rows	Frozen	40.70	40.33	27.06
Lookup-residual rows	Primary PEMA	40.38 $\pm$ 3.26	38.31 $\pm$ 1.86	50.60 $\pm$ 1.24
Lookup-residual rows	Zero-trained PEMA	40.21 $\pm$ 2.04	37.75 $\pm$ 0.07	50.25 $\pm$ 0.05
Lookup-residual rows	Released FT	40.64	39.02	51.47

Table 25: Correct-image interventions under the common single-image scorer (%). N: normal; S: cross-patient shuffled; Z: zero volume. PEMA is averaged over the same three fixed checkpoints in every condition. Small aggregate effects do not imply unchanged predictions.

Task	Method	Macro N	Macro S	Macro Z	Bal. N	Bal. S	Bal. Z
5	PEMA	47.01	45.91	46.38	33.55	31.41	30.71
5	Released	47.97	48.65	45.37	35.50	32.80	35.00
6	PEMA	73.74	74.19	73.69	53.91	54.63	53.99
6	Released	75.46	75.39	73.60	55.82	55.23	53.64

Table 26: Full-data Phi-3 PEMA Task 5 image audit. Accuracy is subtask-macro percent over three seeds; effects are normal minus intervention with 95% Student-t intervals. Training uses all 12,155 patient-disjoint rows.

Condition	Accuracy	Effect [95% CI]
Normal	47.81 $\pm$ 0.84	–
Shuffled	47.41 $\pm$ 1.36	+0.39 [-3.35, +4.14]
Zero	49.80 $\pm$ 1.80	-1.99 [-7.64, +3.66]

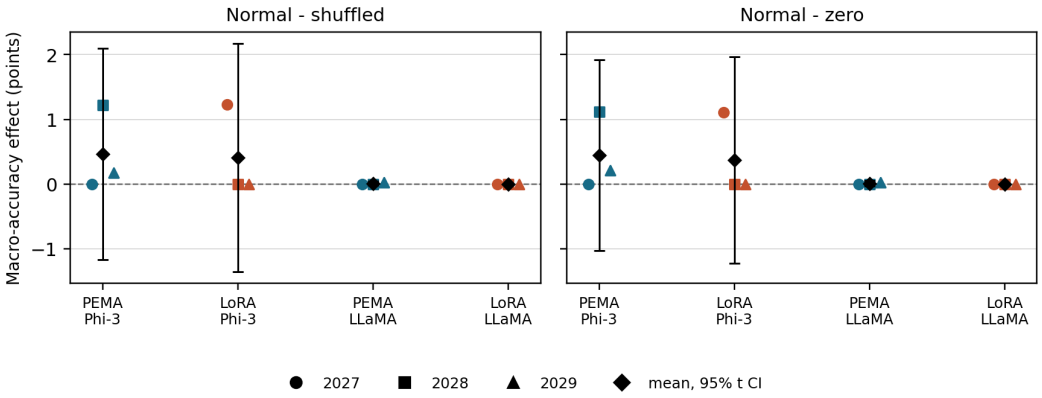


Figure 2: Task 4 image-intervention effects on a fixed test subset balanced between A and B within each included finding subtask. Points are training seeds; diamonds and bars are the mean and 95% Student-t interval across three seeds. Positive values indicate higher subtask-macro accuracy with the correctly paired image.

Table 27: Sensitivity of checkpoint identity to the validation metric. A count indicates that validation balanced accuracy selects a different epoch from the prespecified validation-accuracy rule. This is a selection audit, not a post-hoc replacement of the primary test checkpoints.

Backbone	Method	Runs	Task 4	Task 5	Task 6	Any
M3D-LaMed-Phi-3-4B	PEMA	9	0	1	0	1
M3D-LaMed-Phi-3-4B	LoRA	9	0	3	0	3
M3D-LaMed-LLaMA-2-7B	PEMA	9	2	1	0	3
M3D-LaMed-LLaMA-2-7B	LoRA	9	0	1	0	1

Table 28: Post-hoc LLaMA Task 6 stability follow-up. Values are mean $\pm$ sample standard deviation across five matched seeds. The paired PEMA-minus-LoRA subtask-macro difference is +12.32 points (95% Student-t interval: $[-3.24, +27.87]$).

Method	Subtask-macro	Row	Balanced
PEMA	67.70 ± 10.35	74.84 ± 12.39	49.89 ± 9.18
LoRA	55.39 ± 12.74	70.13 ± 8.30	35.63 ± 14.61

Table 29: Inventory of completed fixed-protocol runs. Accuracy is mean ± sample standard deviation in percent. "Missing" counts runs that never predict at least one answer choice present in the gold test labels; it is a strict missing-choice diagnostic, not only a constant-prediction flag.

Backbone	Task	Method	Seeds	Subtask-macro	Row	Balanced	Missing
M3D-LaMed-Phi-3-4B	Task 4	Visual adapter	3	81.98 ± 0.04	81.98 ± 0.04	50.02 ± 0.02	1/3
M3D-LaMed-Phi-3-4B	Task 4	PEMA	3	81.46 ± 0.64	81.46 ± 0.64	51.76 ± 1.55	1/3
M3D-LaMed-Phi-3-4B	Task 4	LoRA	3	80.56 ± 2.52	80.56 ± 2.52	53.46 ± 5.99	2/3
M3D-LaMed-Phi-3-4B	Task 4	PEMA+LoRA	3	81.84 ± 0.30	81.84 ± 0.30	51.38 ± 2.38	2/3
M3D-LaMed-Phi-3-4B	Task 5	Visual adapter	3	46.59 ± 1.12	57.90 ± 5.91	30.62 ± 4.89	3/3
M3D-LaMed-Phi-3-4B	Task 5	PEMA	3	47.01 ± 1.33	54.37 ± 2.29	33.55 ± 0.58	3/3
M3D-LaMed-Phi-3-4B	Task 5	LoRA	3	47.16 ± 0.99	62.16 ± 1.76	28.89 ± 3.38	3/3
M3D-LaMed-Phi-3-4B	Task 5	PEMA+LoRA	3	49.17 ± 0.63	60.38 ± 2.00	31.07 ± 1.31	3/3
M3D-LaMed-Phi-3-4B	Task 6	Visual adapter	3	73.37 ± 1.71	80.66 ± 0.25	54.27 ± 0.88	3/3
M3D-LaMed-Phi-3-4B	Task 6	PEMA	3	73.74 ± 2.74	80.55 ± 0.96	53.91 ± 1.34	3/3
M3D-LaMed-Phi-3-4B	Task 6	LoRA	3	72.38 ± 0.54	80.95 ± 0.36	54.91 ± 0.55	3/3
M3D-LaMed-Phi-3-4B	Task 6	PEMA+LoRA	3	71.89 ± 1.49	80.35 ± 1.22	52.99 ± 2.59	3/3
M3D-LaMed-LLaMA-2-7B	Task 4	Visual adapter	3	81.47 ± 0.92	81.47 ± 0.92	51.23 ± 2.12	1/3
M3D-LaMed-LLaMA-2-7B	Task 4	PEMA	3	81.95 ± 0.10	81.95 ± 0.10	50.63 ± 1.09	2/3
M3D-LaMed-LLaMA-2-7B	Task 4	LoRA	3	82.01 ± 0.00	82.01 ± 0.00	50.00 ± 0.00	3/3
M3D-LaMed-LLaMA-2-7B	Task 4	PEMA+LoRA	3	82.01 ± 0.00	82.01 ± 0.00	50.00 ± 0.00	3/3
M3D-LaMed-LLaMA-2-7B	Task 5	Visual adapter	3	46.82 ± 1.90	62.14 ± 2.32	27.22 ± 2.05	3/3
M3D-LaMed-LLaMA-2-7B	Task 5	PEMA	3	48.78 ± 2.18	60.89 ± 1.98	30.35 ± 0.57	3/3
M3D-LaMed-LLaMA-2-7B	Task 5	LoRA	3	46.09 ± 0.00	64.11 ± 0.00	25.00 ± 0.00	3/3
M3D-LaMed-LLaMA-2-7B	Task 5	PEMA+LoRA	3	47.60 ± 1.32	61.68 ± 2.39	27.97 ± 2.61	3/3
M3D-LaMed-LLaMA-2-7B	Task 6	Visual adapter	3	72.49 ± 0.54	80.21 ± 0.82	53.98 ± 0.77	3/3
M3D-LaMed-LLaMA-2-7B	Task 6	PEMA	3	72.64 ± 0.90	80.09 ± 1.03	53.56 ± 1.25	3/3
M3D-LaMed-LLaMA-2-7B	Task 6	LoRA	3	61.59 ± 13.43	74.14 ± 8.80	42.72 ± 15.44	3/3
M3D-LaMed-LLaMA-2-7B	Task 6	PEMA+LoRA	3	54.48 ± 13.18	65.65 ± 11.85	37.64 ± 12.71	3/3

Table 30: Within-run paired patient-cluster bootstrap differences for PEMA minus LoRA. Values are percentage points with 95% percentile intervals from 2,000 resamples. These intervals condition on a trained checkpoint and do not replace across-seed uncertainty.

Backbone	Task	Seed	Δ macro	95% interval	Δ balanced	95% interval
M3D-LaMed-Phi-3-4B	Task 4	2027	+4.36	[+3.79, +4.90]	-10.37	[-11.08, -9.64]
M3D-LaMed-Phi-3-4B	Task 4	2028	-1.24	[-1.55, -0.95]	+2.40	[+1.97, +2.85]
M3D-LaMed-Phi-3-4B	Task 4	2029	-0.39	[-0.62, -0.15]	+2.89	[+2.47, +3.32]
M3D-LaMed-Phi-3-4B	Task 5	2027	-2.21	[-8.45, +1.96]	+1.95	[-0.11, +3.94]
M3D-LaMed-Phi-3-4B	Task 5	2028	+2.37	[-1.80, +6.30]	+8.41	[+6.99, +9.89]
M3D-LaMed-Phi-3-4B	Task 5	2029	-0.61	[-3.88, +2.70]	+3.61	[+1.96, +5.41]
M3D-LaMed-Phi-3-4B	Task 6	2027	+1.87	[-2.01, +5.96]	-2.15	[-3.96, -0.38]
M3D-LaMed-Phi-3-4B	Task 6	2028	+4.35	[+0.66, +6.27]	+0.86	[-0.86, +2.61]
M3D-LaMed-Phi-3-4B	Task 6	2029	-2.13	[-5.27, +0.82]	-1.70	[-3.52, +0.16]
M3D-LaMed-LLaMA-2-7B	Task 4	2027	+0.00	[+0.00, +0.00]	+0.00	[+0.00, +0.00]
M3D-LaMed-LLaMA-2-7B	Task 4	2028	+0.00	[+0.00, +0.00]	+0.00	[+0.00, +0.00]
M3D-LaMed-LLaMA-2-7B	Task 4	2029	-0.17	[-0.35, +0.01]	+1.89	[+1.55, +2.23]
M3D-LaMed-LLaMA-2-7B	Task 5	2027	+0.35	[-3.54, +3.43]	+5.24	[+4.11, +6.36]
M3D-LaMed-LLaMA-2-7B	Task 5	2028	+3.06	[+1.33, +4.16]	+4.84	[+3.69, +6.08]
M3D-LaMed-LLaMA-2-7B	Task 5	2029	+4.67	[+1.01, +7.04]	+5.96	[+4.61, +7.41]
M3D-LaMed-LLaMA-2-7B	Task 6	2027	+2.71	[+0.13, +7.82]	+5.13	[+3.25, +6.96]
M3D-LaMed-LLaMA-2-7B	Task 6	2028	+3.67	[-0.49, +9.57]	-0.61	[-3.27, +2.04]
M3D-LaMed-LLaMA-2-7B	Task 6	2029	+26.80	[+21.94, +32.32]	+28.01	[+26.18, +29.89]