Budgeted Counterfactual Reasoning Adaptation via UCB-Guided Layer Selection

Anonymous ACL submission

Abstract

Large language models frequently fail to adhere to counterfactual premises, as pretrained parametric knowledge overrides explicit contextual instructions. We propose budgeted counterfactual adaptation, a framework that mitigates conflicts between context and memory by modifying a restricted subset of algorithmically identified high leverage layers. The pipeline trains a MLP probe to estimate counterfactual fidelity from frozen hidden states. An Upper Confidence Bound search utilizes this probe as a reward signal to navigate the combinatorial layer space and select an optimal subset under a fixed parameter budget. We then apply Low Rank Adaptation exclusively to these targeted layers. Experiments across four open source models, including dense and mixture of experts architectures, demonstrate that this localized approach matches the counterfactual accuracy of global adaptation while reducing trainable parameters by over 82 percent. When applied to a 30B sparse model, the method reduces peak memory usage by 40 percent and decreases training time by a factor of 2.4, enabling adaptation on consumer hardware. Ablation studies confirm that structural localization is critical, as expanding the intervention scope beyond the algorithmically selected layers degrades reasoning performance. This methodology demonstrates that effective network adaptation depends on identifying optimal structural placement rather than scaling the magnitude of parameter updates.

1 Introduction

Large language models frequently encounter hypothetical scenarios that contradict established real world facts, such as being asked to reason in a world where gravity pushes objects upward. Despite explicit instructions to adopt these counterfactual premises, modern models often revert to pretrained priors and generate responses based on factual reality. This failure mode, where parametric memory overrides contradictory context, presents a practical obstacle for deploying these models in settings that require strict instruction following under altered premises and hypothetical reasoning (Zhao et al., 2023).

Standard approaches address this issue by strengthening prompting techniques or adding longer reasoning rationales such as Chain of Thought. However, preliminary experiments indicate that these safeguards act unreliably in the presence of strong factual priors. Instead of suspending the prior, extended rationales can amplify it, resulting in fluent responses that ultimately fail to adhere to the hypothetical premise. This observation motivates an optimization perspective. Correcting counterfactual fidelity requires parameter level adjustments, yet global adaptation introduces broad interference with established general capabilities.

We formulate this challenge as budgeted counterfactual adaptation. Under a strict update budget, the objective is to modify the minimum number of parameters necessary to correct conflicts between context and memory. We hypothesize that counterfactual alignment relies on a small set of high leverage layers, making global network updates unnecessary and prone to capability degradation. Our research objective centers strictly on computational efficiency and adaptation stability rather than mechanistic interpretability. While uncovering the exact internal representations and functional roles of these identified high leverage layers constitutes a valuable direction for future research (Olah et al., 2020; Elhage et al., 2021), the present framework focuses entirely on algorithmically exploiting the observed empirical sparsity. We aim to improve counterfactual fidelity and reduce training costs by localizing parameter updates without committing to causal claims regarding the emergence of these specific layers.

To operationalize this hypothesis, we propose an automated framework that trains a lightweight

probe as a proxy signal for counterfactual fidelity. The framework applies a search guided by an Upper Confidence Bound(UCB) strategy over layer subsets to efficiently identify candidate high leverage layers within the combinatorial space. Guided by the discovered subset, we apply Low Rank Adaptation (Hu et al., 2021) exclusively to the selected layers. This targeted approach seeks to improve counterfactual fidelity while preserving broad network competence under the same training budget.

We present three primary contributions. First, we show that the parameter updates necessary to correct conflicts between provided context and parametric memory exhibit structural sparsity. We demonstrate that counterfactual adaptation functions as a constrained optimization problem where parameter modifications target a restricted subset of effective layers rather than the global network. Second, by utilizing this observed structural sparsity, we build an automated search framework that couples a Gated MLP probe with an Upper Confidence Bound strategy. We integrate this search mechanism with Low Rank Adaptation to apply updates exclusively to the algorithmically identified layers, removing the need for manual or heuristic layer selection. Third, through experiments with Llama 3.1 and Qwen architectures up to 30B parameters, we demonstrate that this targeted approach reduces trainable parameters by over 80 percent and enables model adaptation on consumer hardware. We show that the method increases accuracy on ConJNLI and CounterFact benchmarks while strictly preserving baseline performance on general mathematical and logical reasoning tasks.

2 Related Work

Our work targets *Budgeted Counterfactual Adaptation*: improving premise-following under counterfactual or contradictory context while strictly controlling the parameter-update budget. We position this as filling a practical gap in Parameter-Efficient Fine-Tuning (PEFT): *automatic, budget-controllable layer selection* (High-Leverage Layers).

2.1 Counterfactual Reasoning and Premise Following under Knowledge Conflicts

Counterfactual reasoning concerns drawing inferences under hypothetical premises that contradict reality (Pearl, 2009; Byrne, 2007). In LLMs, a closely related difficulty arises when instructions conflict with pre-trained priors, leading to knowledge-conflict failures where parametric memory overrides contextual premises (Longpre et al., 2021; Xie et al., 2023). Benchmarks such as CounterFact (Meng et al., 2022) and CLOMO (Huang et al., 2024) operationalize these settings, while datasets like CRASS provide additional counterfactual evaluation signals (Frohberg and Binder, 2022). Prior attempts to improve robustness often rely on prompting or scaling fine-tuning data (Wu et al., 2023). In contrast, we focus on *parameter-efficient* adaptation under a strict budget, aiming to improve counterfactual premise adherence without broad model interference.

2.2 PEFT Lineage: LoRA, Budget Allocation, and Layer/Module Selection

PEFT methods reduce adaptation cost by updating only a small subset of parameters. LoRA (Hu et al., 2021) is a widely used baseline that injects low-rank adapters, typically applied broadly across layers and modules. AdaLoRA (Zhang et al., 2023) further refines this paradigm by *allocating low-rank budget adaptively* (e.g., varying effective ranks) based on training dynamics, improving budget utilization without necessarily answering *where* to adapt under a fixed layer budget. LISA (Rui et al., 2024) explores *layer/module-level selectivity* via sampling schedules to save memory, demonstrating that not all layers need to be updated at every step. Our work complements these approaches by targeting a distinct missing piece: *automated discovery of a task-relevant, controllable-budget layer subset* (High-Leverage Layers) for counterfactual premise following, which can be viewed as an upstream selection step for PEFT.

2.3 Automated Selection/Search: UCB as an Efficient Engineering Selector

Selecting which layers to adapt is inherently combinatorial, and exhaustive evaluation is infeasible at realistic budgets. Multi-armed bandit methods provide principled tools for sequential decision-making under limited trials, with Upper Confidence Bound (UCB) offering a classic exploration–exploitation trade-off (Auer et al., 2002; Slivkins, 2024). We use UCB *as an engineering-efficient selector* to navigate the layer-subset space: iteratively proposing candidate subsets based on a proxy reward signal and updating selection statistics over time. Importantly, this use of UCB is not positioned as a mechanistic explanation tool; rather,

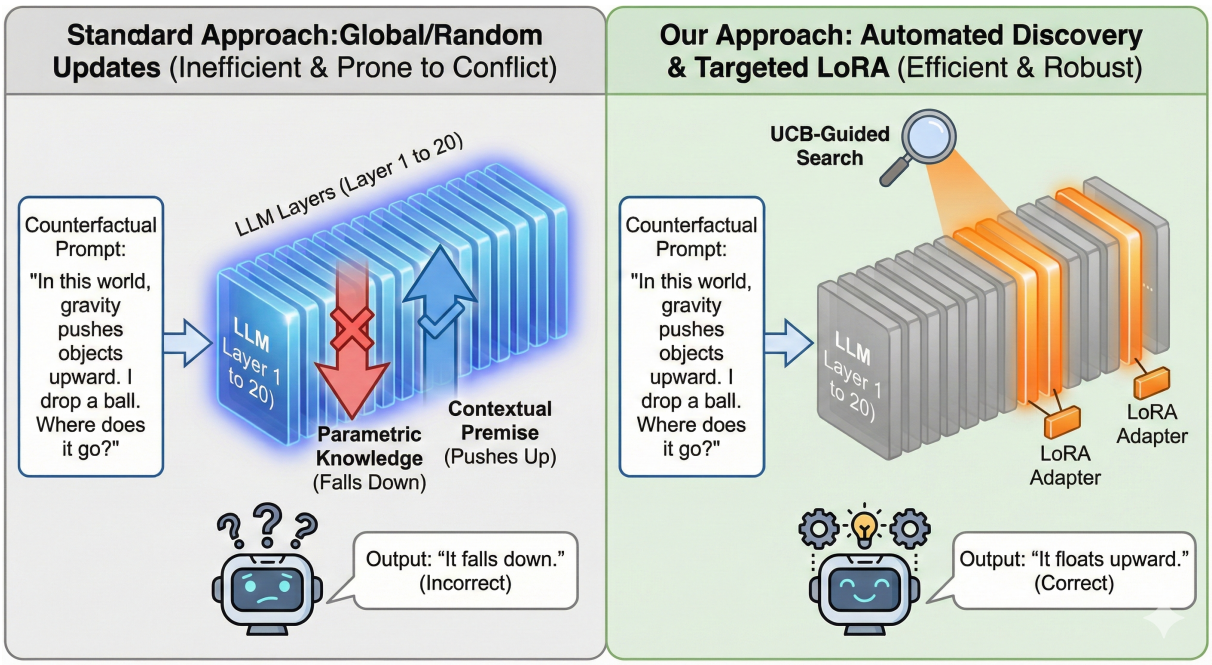

Figure 1: Overview of the budgeted counterfactual adaptation framework. The left panel illustrates how standard approaches struggle to resolve the conflict between pretrained parametric knowledge and counterfactual context, resulting in factual overrides. The right panel demonstrates the proposed framework, which uses an automated probe guided by an Upper Confidence Bound strategy to identify high leverage layers responsible for context integration. Applying Low Rank Adaptation strictly to these layers improves counterfactual reasoning accuracy while controlling the parameter update budget and minimizing interference.

it is a practical mechanism to achieve *automatic, budget-controllable* layer selection for PEFT in counterfactual adaptation.

3 Methodology

This section details the framework for budgeted counterfactual adaptation. The methodology consists of four sequential phases designed to improve instruction following under contradictory contexts while maintaining strict control over the parameter budget. First, we construct a specialized counterfactual benchmark to induce conflicts between parametric memory and provided premises, establishing a baseline evaluation signal. Next, we train a lightweight probe to act as a proxy reward for measuring counterfactual fidelity. We then employ a search mechanism guided by an Upper Confidence Bound strategy to automatically identify a restricted subset of high leverage layers. Finally, we apply Low Rank Adaptation exclusively to these selected layers to execute targeted model updates.

3.1 Counterfactual Benchmark Construction

We construct counterfactual (Passage, Question, Reference) tuples from American Stories (Dell et al., 2023) to induce conflicts between parametric memory and contextual premises. We filter this corpus to identify passages rich in multi-hop reasoning potential, specifically targeting narratives containing multiple causal factors and distinct entity interactions that allow for plausible hypothetical interventions. Using Kimi-K2-Instruct(Team et al., 2025) as the generative engine, we transform these raw texts into a standardized format of 2,000 tuples. We generate questions that require accepting a hypothetical intervention (e.g., a change in a character's decision or a physical law) and deducing downstream consequences, and we provide graded reference answers (0–10) as anchors for consistent scoring and calibration. To facilitate precise evaluation, for each problem, we generate a spectrum of six reference answers ranging in quality from 0 (completely irrelevant or hallucinated) to 10 (perfect counterfactual deduction). These reference answers serve as few-shot anchors for the judging phase. The detailed prompt templates and dataset examples are provided in Appendix A.

3.2 Grading and Evaluation Protocol

We adopt a dual-judge setup using two strong, distinct LLMs (Llama-3.3-70B-Instruct and Qwen-2.5-72B-Instruct) to score responses on a 0 to 10

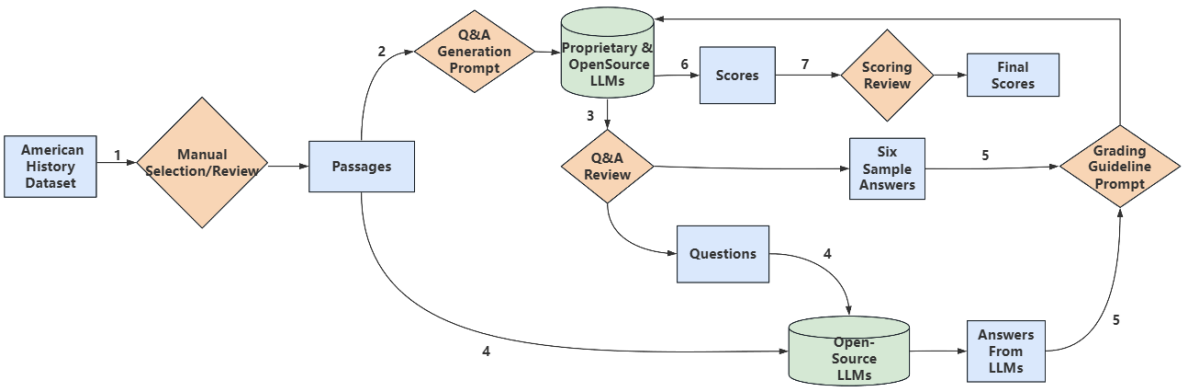

Figure 2: Overview of the Data Construction and Evaluation Pipeline. The framework consists of four phases spanning seven steps

scale along: (1) adherence to the counterfactual premise, (2) consistency with the modified context, and (3) completeness. If the two judges differ by ≤ 1, we average the scores; otherwise the example is flagged for manual adjudication.

To train the probe with a high signal-to-noise ratio, we binarize scores with strict filtering: examples scored 7 or above by both judges are labeled positive, and examples scored 6 or less by both judges are labeled negative; boundary-straddle cases are discarded. This yields a high-confidence dataset of approximately 1,500 labeled examples per LLM, maintained at a negative-to-positive ratio between 1:1 and 2:1, used solely for probe training and layer-subset selection.

3.3 The Conflict-Aware MLP Probe

To efficiently localize the "Effective Layers" without exhaustive manual intervention, we require a lightweight proxy capable of estimating the reasoning fidelity of internal representations. We design a gated MLP probe to discern between context-adherent signals and memory-reliant hallucinations within the frozen model.

Formally, let $\mathbf{h}_l \in \mathbb{R}^d$ denote the mean-pooled hidden state of the entire prompt context extracted from layer l. This representation aggregates the model's internal processing of the counterfactual premise prior to generation. We first project $\mathbf{h}_l$ into a unified latent space via a linear transformation and LayerNorm, yielding a compacted embedding $\mathbf{e}_l \in \mathbb{R}^{256}$.

To model the inherent tension between the counterfactual context and parametric memory, we process $\mathbf{e}_l$ through two parallel, non-linear branches, producing latent feature vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ (each $\in \mathbb{R}^{64}$). These branches are designed to capture competing semantic evidences, such as the conflict between the "factual" world state and the "counterfactual" instruction, without imposing rigid constraints.

We employ a dynamic gating mechanism to weigh these conflicting signals. A gate vector $\mathbf{g} \in [0,1]^{64}$ is computed via a sigmoid activation over the concatenated features:

$$\mathbf{g} = \sigma\left(\mathbf{W}_g[\mathbf{v}_1; \mathbf{v}_2] + \mathbf{b}_g\right),$$

where $[;]$ denotes concatenation. The final fused representation $\mathbf{z}$ is obtained via element-wise interpolation, allowing the probe to selectively amplify features that correlate with correct reasoning while suppressing noise:

$$\mathbf{z} = \mathbf{g} \odot \mathbf{v}_1 + (1 - \mathbf{g}) \odot \mathbf{v}_2.$$

Finally, $\mathbf{z}$ is passed through a binary classification head to predict the probability $P(y = 1 \mid \mathbf{h}_l)$ that the given layer's state will lead to a correct counterfactual response (Score ≥ 7).

The probe is trained on a frozen LLM using a balanced subset of the constructed benchmark (approx. 2000 samples), ensuring an equal distribution of positive and negative examples. We optimize the model using binary Cross-Entropy loss. To ensure the reliability of this reward signal for the subsequent search phase, we employ stratified 5-fold cross-validation, optimizing the decision threshold to maximize the F_1 score on the validation set. Detailed hyperparameters are provided in Appendix B.

3.4 Automated Layer Selection

Identifying the optimal layers for adaptation presents a combinatorial challenge where exhaus-

tive search remains computationally infeasible under standard resource constraints. We formulate this selection process as a multi-armed bandit problem. Let $\mathcal{S}$ denote a candidate layer subset of fixed size K drawn from the total available layers N. We treat each candidate subset as an independent arm and apply an Upper Confidence Bound heuristic (Auer et al., 2002; Slivkins, 2024) to allocate a restricted number of evaluation trials across these subsets. This approach provides a sample efficient mechanism for automated layer selection while strictly controlling the computational budget.

During each trial, the framework evaluates a candidate subset by training the probe on representations extracted from the layers within $\mathcal{S}$. We measure the validation F_1 score to serve as the reward signal. The algorithm subsequently updates the layer statistics and proposes the next subset to evaluate by balancing exploration and exploitation. This process continues until the search converges or exhausts the fixed trial budget. The resulting optimal subset $\mathcal{S}^*$ defines the high leverage layers utilized for the subsequent targeted adaptation phase.

To establish a reliable statistical baseline, we initiate the process with a bootstrap phase that samples an initial set of $M = 200$ random layer combinations. We impose a coverage constraint ensuring that every unique layer l from the set of all N layers appears in at least two distinct combinations. For each sampled subset $\mathcal{S}_i$, we train an independent instance of the probe and record its corresponding validation performance. At step t, the algorithm assigns an Upper Confidence Bound score to each layer l based on the average performance of all previously evaluated subsets containing that layer:

$$\text{Score}(l) = \bar{\mu}_l + c\sqrt{\frac{\ln t}{n_l}},$$

where $\bar{\mu}_l$ represents the mean reward of all evaluated subsets that include layer l, n_l denotes the total selection count of layer l, and c acts as a scalar hyperparameter controlling the exploration rate.

For all subsequent iterations, rather than sampling randomly, the algorithm constructs a new candidate subset $\mathcal{S}_{new}$ by selecting the highest ranking K layers according to their current computed scores. We train the probe on this newly synthesized configuration, observe the resulting validation score, and propagate this reward backward to update the running statistics for all participating layers. This iterative feedback loop drives the search algorithm to converge efficiently on the structural components most critical for processing counterfactual premises.

3.5 Targeted Low Rank Adaptation

Once the optimal subset of High-Leverage Layers $\mathcal{S}^*$ is identified, we proceed to the intervention phase. Given $\mathcal{S}^*$, we apply LoRA (Hu et al., 2021) only to modules within the selected layers, keeping the rest of the network frozen. This implements budgeted adaptation directly at the layer-selection level: the update budget is controlled by K and the adapter configuration, rather than by uniformly adapting the entire network.

To generalize the model's logical robustness beyond specific counterfactual instances, we utilize the CLOMO tuning dataset (Huang et al., 2024) for this optimization phase. We train the selected adapters using a standard causal language modeling objective, minimizing the negative log-likelihood of the next token:

$$\mathcal{L} = -\sum_t \log P(x_t \mid x_{<t}; \Theta_{fixed}, \Delta\Theta_{\mathcal{S}^*}),$$

where $\Delta\Theta_{\mathcal{S}^*}$ represents the trainable LoRA parameters injected only at the depths defined by $\mathcal{S}^*$. Implementation details vary slightly by architecture to maximize efficiency. For dense models, we target only the attention modules within $\mathcal{S}^*$ using a rank $r = 16$ and scaling factor $\alpha = 32$. For the Mixture-of-Experts architecture, we scale the intervention to accommodate its sparse activation patterns, targeting both attention modules and expert MLPs within the selected layers with $r = 64$ and $\alpha = 128$, while explicitly keeping the router networks frozen to maintain routing stability. Full hyperparameter configurations are detailed in Appendix B.

4 Experiments

In this section, we evaluate the efficacy of our automated layer selection framework across three critical dimensions: counterfactual fidelity, general reasoning robustness, and computational efficiency.

4.1 Experimental Setup

We conduct a comprehensive evaluation across four state-of-the-art open-source LLMs spanning dense and sparse architectures: Llama-3.1 (8B), Qwen-2.5 (7B), Qwen-3 Dense (8B), and Qwen-3 MoE (30B-A3B).

Table 1: ConJNLI accuracy (higher is better). Best per backbone is **bold**.

Backbone	Base	All-layer	AdaLoRA	LISA	Ours
Llama-3.1	0.3307	0.3034	0.3323	0.2681	**0.4029**
Qwen-2.5	0.5056	0.4928	0.5056	0.5072	**0.5136**
Qwen-3 Dense	0.5393	0.5634	0.5377	**0.6661**	0.5281
Qwen-3 MoE	0.6581	0.6212	0.6581	**0.6613**	0.6517

Table 2: CounterFact harmonic score S (higher is better). Best per backbone is **bold**.

Backbone	Base	All-layer	AdaLoRA	LISA	Ours
Llama-3.1	0.2966	0.2837	0.2972	**0.3043**	0.2917
Qwen-2.5	0.3969	0.3972	0.3974	0.3971	**0.3978**
Qwen-3 Dense	0.3811	0.3812	0.3808	0.3811	**0.3815**
Qwen-3 MoE	0.3616	0.3595	**0.3622**	0.3594	0.3605

To isolate the effect of surgical layer selection, we compare against four baselines: (i) the unmodified Base model (zero-shot), (ii) standard All-Layer LoRA (Hu et al., 2021), (iii) AdaLoRA (Zhang et al., 2023), and (iv) LISA (Rui et al., 2024).

All layer search operations utilize the balanced MLP probing dataset (approx. 1,500 samples per model) described in Section 3. We evaluate the resulting models along three axes: (1) Counterfactual Fidelity on our Counterfactual Benchmark, (2) automatic counterfactual benchmarks (ConJNLI and CounterFact (Meng et al., 2022)), and (3) General Robustness on ARC (Clark et al., 2018), RACE (Lai et al., 2017), DROP (Dua et al., 2019), BBH (Suzgun et al., 2022), and GSM8K (Yu et al., 2023). We report efficiency metrics in Section 4.3.

4.2 Main Results

We evaluate the framework on a held out test set of 500 counterfactual reasoning problems generated specifically for final validation and distinct from the pool used for probe training. On this internal benchmark, we compare targeted Low Rank Adaptation against the unmodified base model and a standard global adaptation baseline. For automatic and open benchmarks, we additionally report results for AdaLoRA and LISA.

To complement the primary evaluation, we assess performance on two automatic benchmarks that do not require language model judges. Table 1 reports classification accuracy on the ConJNLI dataset. The targeted approach achieves the highest accuracy on Llama-3.1, while results across the Qwen-3 variants show varying optimal methods. Following the standard CounterFact evaluation protocol (Meng et al., 2022), Table 2 reports the harmonic score across 2191 samples, demonstrating consistent baseline maintenance or marginal improvements across the tested architectures.

Table 3 presents the average scores on the internal counterfactual benchmark across the tested architectures. For Llama-3.1, targeted adaptation increases the base model score from 4.57 to 5.88, exceeding both LISA and AdaLoRA. While global adaptation achieves a higher absolute score of 6.46 on this specific architecture, this unconstrained approach causes performance inconsistencies across other model families. On Qwen-2.5, targeted adaptation improves performance to 4.69, surpassing AdaLoRA. In the Qwen-3 Dense architecture, the targeted approach achieves 6.59, approximating the global adaptation score of 6.63 and exceeding LISA. For the sparse Qwen-3 MoE model, targeted adaptation reaches 7.49, remaining competitive with global adaptation at 7.63. These results indicate that localizing interventions to specific layers provides an optimization signal comparable to full network updates while maintaining architectural sparsity.

Table 3: Comparison of average scores (0-10 scale) across tested LLMs with different fine-tuning methods. Best results for each model are **bold**.

Model	Base	Ours	All-layer	LISA	AdaLoRA
Llama-3.1	4.57	5.88	**6.46**	4.21	4.53
Qwen-2.5	4.13	4.69	5.36	**6.56**	4.24
Qwen-3 Dense	6.60	6.59	**6.63**	6.53	6.61
Qwen-3 MoE	7.53	7.49	**7.63**	7.60	7.47

Beyond counterfactual fidelity, we evaluate whether targeted adaptation affects general reasoning utility across five open benchmarks, including ARC, BBH, DROP, GSM8K, and RACE. Table 4 incorporates AdaLoRA and LISA as strong parameter efficient baselines. The results reveal an architecture dependent tradeoff between stability and plasticity. Targeted adaptation improves semantic reasoning metrics on ARC, BBH, and DROP for Qwen-2.5, but yields lower accuracy on the mathematics focused GSM8K benchmark for the Qwen-3 Dense model. In the sparse Qwen-3 MoE architecture, the targeted method improves BBH and DROP scores while maintaining competitive performance on the remaining tasks, despite updating a restricted fraction of the total parameters.

Table 4: Generalization on open benchmarks. **Bold** indicates the best method per backbone and metric.

Method	ARC	BBH	DROP	GSM8K	RACE
Llama-3.1					
Base	0.1854	0.2758	0.0087	0.0584	**0.5028**
All-layer	0.0761	0.2668	**0.1285**	**0.1607**	0.4978
AdaLoRA	0.1930	0.2700	0.0018	0.0569	0.5018
LISA	**0.3357**	0.1978	0.0010	0.0023	0.4636
Ours	0.2584	**0.2800**	0.0463	0.0364	0.4998
Qwen-2.5					
Base	0.53	0.45	0.58	0.62	**0.61**
All-layer	0.58	0.44	0.58	**0.82**	0.58
AdaLoRA	0.52	0.4482	0.6318	0.6406	0.6052
LISA	0.48	0.47	0.62	0.66	0.5883
Ours	**0.92**	**0.54**	**0.64**	0.71	0.56
Qwen-3 Dense					
Base	0.73	0.38	0.45	**0.59**	0.53
All-layer	0.79	**0.48**	0.41	0.58	**0.75**
AdaLoRA	0.73	0.33	**0.5072**	0.53	0.53
LISA	0.7258	0.31	0.50	0.46	0.57
Ours	**0.83**	0.43	0.46	0.27	0.54
Qwen-3 MoE					
Base	0.86	0.72	0.52	0.79	0.65
All-layer	**0.91**	0.70	0.59	**0.89**	**0.88**
AdaLoRA	0.84	0.72	0.59	0.78	0.67
LISA	0.7858	0.7394	0.61	0.7748	0.7168
Ours	0.90	**0.74**	**0.67**	0.86	0.85

4.3 Efficiency Analysis

We evaluate the cost of fine-tuning as a function of trainable parameter count, total training time, and peak GPU memory reservation. To quantify these benefits, we compare our Targeted LoRA strategy against standard LoRA applied to all layers, alongside advanced PEFT baselines (LISA and AdaLoRA).

For smaller dense architectures (7B/8B), the memory bottleneck is dominated by the frozen base weights rather than gradient states, resulting in negligible VRAM differences across methods. However, the storage and computational efficiency gains remain substantial. Compared to standard All-Layer LoRA, Targeted LoRA reduces the number of trainable parameters by over 82%, drastically lowering the storage overhead for checkpoints and reducing bandwidth requirements for deployment. Furthermore, when strictly aligned with the same parameter budget against dynamic PEFT methods, our approach consistently outperforms LISA in training speed on dense models, achieving a 1.6× to 1.8× speedup. This efficiency gap primarily stems from LISA's layer-switching mechanism, which incurs significant overhead in dynamically toggling `requires_grad` flags and managing optimizer states for sampled subsets.

The largest efficiency gains emerge when adapting the sparse Qwen 3 architecture. Global adaptation requires updating every expert across all layers, increasing the trainable parameter count to 3.37B and demanding 64.7 GB of memory. This requirement typically necessitates data center hardware. In contrast, the targeted strategy activates approximately 421 million parameters, achieving an 87.5% reduction in parameter volume. This localized intervention reduces peak memory usage to 38.5 GB and decreases total training time by a factor of 2.4, scaling from 5387 seconds down to 2209 seconds. The resulting memory reduction enables the adaptation of models exceeding 30B parameters on consumer workstation hardware that would otherwise experience memory exhaustion errors. Additionally, the targeted approach scales more efficiently than AdaLoRA on the sparse architecture. While the dynamic rank allocation of AdaLoRA incurs substantial computational overhead, demanding 9023 seconds and 64.0 GB of memory, the targeted approach avoids this penalty and maintains the lowest memory footprint among the evaluated methods.

Table 5: Comparison of Training Efficiency under Strict Parameter Budget Alignment.

Model	Method	Params (M)	Time (s)	Mem (GB)
Qwen-3 MoE	All-Layer LoRA	3,375.0	5,387	64.7
	LISA	421.0	2,129	42.4
	AdaLoRA	3,376.6	9,023	64.0
	Ours	**421.0**	**2,209**	**38.5**
Qwen-3 Dense	All-Layer LoRA	15.3	267	11.5
	LISA	2.5	419	14.6
	AdaLoRA	43.7	451	11.4
	Ours	**2.5**	**240**	**11.5**
Qwen-2.5	All-Layer LoRA	10.1	213	10.8
	LISA	1.8	338	13.6
	AdaLoRA	40.4	375	10.5
	Ours	**1.8**	**191**	**10.8**
Llama-3.1	All-Layer LoRA	13.6	238	10.7
	LISA	2.1	386	12.5
	AdaLoRA	41.9	406	10.4
	Ours	**2.1**	**214**	**10.6**

4.4 Ablation and Sensitivity Analysis

We conduct ablation studies focusing on two dimensions: the sensitivity of the model to the specific number of selected layers (K) and the efficacy of the UCB search strategy compared to stochastic selection. We analyze performance deviations across subset sizes of $K-1$, K, and $K+1$ for all models. Table 6 summarizes the average performance change relative to the optimal size K.

Reducing the subset size to $K-1$ results in measurable performance drops, particularly in Qwen-2.5, indicating that the restricted subset is insufficient to capture the complete reasoning circuit. Conversely, increasing the size to $K+1$ yields no marginal gains and instead causes performance degradation. This finding supports the interference hypothesis: expanding the intervention scope beyond the critical set introduces parametric noise, disrupting the pre-existing knowledge structures of the model.

Detailed per-trial statistics further validate the UCB search algorithm. We compare the top candidates identified by UCB against a baseline of randomly selected layer combinations of the same size K. The final column in Table 6 demonstrates that UCB-selected subsets consistently outperform random subsets. This advantage peaks in Qwen-2.5, where the UCB strategy surpasses the random baseline by 20.1%. These results indicate that the efficacy of the method relies on the precise anatomical localization of the intervention rather than merely the parameter efficiency.

Table 6: Sensitivity analysis. We report the optimal subset size (K) and the performance deviation (Δ) when reducing ($K-1$) or expanding ($K+1$) the selection. The final column shows the relative improvement of the UCB method over random selection.

Model	Optimal K	$\Delta(K-1)$	$\Delta(K+1)$	UCB Gain
Qwen-2.5	5	-20.2%	-19.6%	$+20.1\%$
Qwen-3 Dense	6	-3.2%	-4.9%	$+6.3\%$
Qwen-3 MoE	6	-3.2%	-0.6%	$+1.4\%$

4.5 Validation of the Proxy Signal

Because the UCB algorithm relies entirely on the probe validation score as its reward signal to navigate the combinatorial layer space, the reliability of this proxy is critical to the framework. If the probe overfits the probing set or fails to capture task-relevant features, the search would converge on arbitrary layer subsets, rendering the subsequent adaptation ineffective. To validate this proxy signal, we evaluate the gated MLP probe on the final UCB-selected subsets $\mathcal{S}^*$ using stratified 5-fold cross-validation.

As shown in Table 7, the probe achieves stable performance across all backbones, with F_1 scores ranging from 0.69 to 0.75 and ROC-AUC scores between 0.73 and 0.84. These metrics exceed the random chance baseline, confirming that the selected subsets $\mathcal{S}^*$ reliably concentrate task-relevant features. The low variance across folds, ranging from ± 0.01 to ± 0.04, indicates that the probe generalizes effectively rather than memorizing the probing set, providing a stable reward landscape for the search. The slightly lower scores for the Qwen-3 MoE model reflect the inherent complexity of sparse architectures, yet the proxy remains sufficiently discriminative to guide the selection process. These results confirm that the automated framework grounds the targeted LoRA updates in empirically verified layer subsets rather than heuristic selection.

Table 7: Gated MLP probe performance (5-fold cross-validation) on the probing set using the selected subset $\mathcal{S}^*$.

Backbone	$F_1 \uparrow$	ROC-AUC $\uparrow$
Llama-3.1	0.73 ± 0.03	0.77 ± 0.04
Qwen-2.5	0.75 ± 0.01	0.84 ± 0.02
Qwen-3 Dense	0.74 ± 0.03	0.77 ± 0.03
Qwen-3 MoE	0.69 ± 0.04	0.73 ± 0.04

5 Conclusion

We introduce budgeted counterfactual adaptation, a parameter-efficient framework that resolves context–memory conflicts in LLMs by strategically placing targeted updates. By formulating layer selection as a combinatorial optimization problem and coupling an MLP probe with a UCB search strategy, we automatically identify high-leverage layers for localized LoRA. Across dense and sparse architectures, our method matches or exceeds global adaptation in counterfactual fidelity while preserving general reasoning capabilities. Crucially, it reduces trainable parameters by over 82% and significantly lowers computational overhead, enabling efficient adaptation of large MoE models on consumer hardware. This work shifts the paradigm of efficient adaptation from scaling update magnitudes to algorithmically optimizing their structural placement, laying a quantitative foundation for targeted knowledge editing and reasoning alignment.

Limitations

Despite the promising results, our framework has several limitations warranting future investigation:

- The pipeline identifies *where* interventions work but not why; formal analysis of the func-

tional roles of selected layers remains an open direction.

- Targeted LoRA may improve semantic counterfactual reasoning at the expense of strict mathematical deduction (e.g., GSM8K drops on certain dense backbones), suggesting partial overlap between hypothetical-adoption and logical-execution circuits.

- Current interventions operate at the layer level; extending UCB search to the module or expert level (especially for MoE models) could enable finer-grained control.

- While our gated MLP probe performs robustly overall, its accuracy degrades on highly sparse architectures (e.g., Qwen-3 MoE, $F_1 \approx 0.69$), potentially leading to sub-optimal layer selection.

Acknowledgments

References

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256.

Ruth MJ Byrne. 2007. *The rational imagination: How people create alternatives to reality*. MIT press.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Melissa Dell, Jacob Carlson, Tom Bryan, Emily Silcock, Abhishek Arora, Zejiang Shen, Luca D'Amico-Wong, Quan Le, Pablo Querubin, and Leander Heldring. 2023. American stories: A large-scale structured text dataset of historical u.s. newspapers. *Preprint*, arXiv:2308.12477.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *CoRR*, abs/1903.00161.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Anthropic*. Https://transformer-circuits.pub/2021/framework/index.html.

Jörg Frohberg and Frank Binder. 2022. Crass: A novel data set for counterfactual reasoning assessment of commonsense reasoning. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 578–586.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Yinya Huang, Ruixin Hong, Hongming Zhang, Wei Shao, Zhicheng Yang, Dong Yu, Changshui Zhang, Xiaodan Liang, and Linqi Song. 2024. CLOMO: Counterfactual logical modification with large language models. *The 62nd Annual Meeting of the Association for Computational Linguistics*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, and Marco Tulio Kuchibhotla, Chris andRibeiro. 2021. Entity-based knowledge conflicts in question answering. *arXiv preprint arXiv:2109.05052*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Judea Pearl. 2009. *Causality*. Cambridge university press.

Pan Rui, Liu Xiang, Diao Shizhe, Pi Renjie, Zhang Jipeng, Han Chi, and Tong Zhang. 2024. Lisa: Layerwise importance sampling for memory-efficient large language model fine-tuning. *arXiv preprint arXiv:2403.17919*.

Aleksandrs Slivkins. 2024. Introduction to multi-armed bandits. *Preprint*, arXiv:1904.07272.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, and 150 others. 2025. Kimi k2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

Zhaofeng Wu, Yijia Lin, Sailik Rakheja, Luis Li, Haoming Jiang, and Zhixuan Xu. 2023. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *arXiv preprint arXiv:2307.02477*.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Benjamin Van Durme. 2023. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. *arXiv preprint arXiv:2305.13300*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqi Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Dataset Preparation and Grading Details

The following example presents a problem (passage and question) and the six corresponding sample answers.

> Passage: A crow was thirsty and searched everywhere for water. It saw a bottle with some water inside. However, the bottle was tall and its mouth was very narrow, so the crow could not reach the water with its beak. After thinking for a while, the crow noticed many small stones nearby. It dropped the stones into the bottle one by one. As the stones accumulated, the water level rose, and eventually the crow was able to drink.
>
> Question: If there had been no small stones, what would the crow most likely do?
>
> Sample Answers: A: I don't know; the passage does not provide enough context or information. Please provide more details so that I can answer the question. B: The crow would stare blankly at the sky, fantasizing that the water would fly into its beak automatically, without taking any practical action to solve the problem. C: The crow would peck at the bottle blindly, hoping to break it to access the water, without considering the bottle's sturdiness or the feasibility of this approach. D: It might wait for other animals to come drink nearby, observe how they obtain water, and try to imitate them. E: The crow might look for a relatively large leaf and bring it to the bottle's mouth, then repeatedly squeeze or wick the water upward until it reaches a drinkable position. F: The crow might try to find a long, thin hollow grass stem or twig and use it as a straw to suck the water from the bottle, consistent with crows' tool-using tendencies and the ready availability of such materials in nature.

Answers are graded on a 10-point scale based on comprehensiveness, accuracy, and level of detail, with 0 as the lowest and 10 as the highest. The high-level grading guidelines are:

0 points: Completely off-topic or meaningless, with no attempt to answer the question. 1–2 points: Superficially relevant but lacking clarity or counter-

factual reasoning. 3–4 points: Somewhat relevant with basic counterfactual reasoning, but with gaps or inaccuracies. 5–6 points: Clear and relevant with counterfactual reasoning, though lacking depth or nuance. 7–8 points: Well-structured and insightful, with strong counterfactual reasoning and minimal errors. 9–10 points: Fully relevant, clear, and detailed, demonstrating deep and accurate counterfactual reasoning.

> **Grading Prompt**: You will receive a Passage, a Question, and a Response. As the grader, assign a score from 0 to 10 based solely on how well the Response answers the Question using information from the Passage. Do not consider factors beyond the Passage and the Question. Passage: {Passage}. Question: {Question}. Response: {Response}. For reference, consider the following answers: The perfect answer: {F}. The second-best answer: {E}. The better answer: {D}. The medium answer: {C}. The poor/meaningless answer: {B}. The worst/meaningless answer: {A}. Your reply must begin with the score (0–10) on the first line, followed by a concise justification. Think step by step before finalizing the score.

B MLP Training and LoRA Configuration

In this section, we provide the detailed experimental setup used for training the Gated MLP Probe, ensuring reproducibility of our results.

B.1 Preprocessing and Optimization

Inputs are tokenized using the respective base LLM's tokenizer, truncated to a maximum sequence length of $T = 960$ to balance context coverage with computational efficiency. The backbone LLM parameters are strictly frozen; we only extract hidden states from the selected candidate layers to serve as inputs for the probe.

The probe is optimized using the **AdamW** optimizer (Loshchilov and Hutter, 2019) with cosine annealing learning rate scheduling ($\eta_{min} = 0.1 \times$ base_lr). To prevent gradient explosion, we apply global gradient clipping with a maximum norm of 1.0. The loss function is a weighted Binary Cross-Entropy (BCE) with Logits Loss, designed to handle potential class imbalances.

Table 8: **Hyperparameter Settings for Probe Training.**

Hyperparameter	Value
Max Sequence Length	960
Optimizer	AdamW
Loss Function	BCEWithLogitsLoss
Positive Class Weight	0.5
Gradient Clipping Norm	1.0
Learning Rate Schedule	Cosine Annealing
Minimum LR Ratio	0.1
Label Smoothing	Optional (Adaptive)
Dropout Strategy	Epoch-wise Scheduling

B.2 Model Selection and Evaluation

To ensure robust performance estimation, we employ **Stratified 5-Fold Cross-Validation**. The data is shuffled, and embeddings are re-extracted per fold. The final reported metrics are the mean $\pm$ standard deviation across these folds.

Decision Threshold Search. Since binary classification on imbalanced data is sensitive to the decision boundary, we do not use a fixed threshold of 0.5. Instead, during the validation phase, we perform a grid search for the optimal decision threshold t:

$$t^* = \underset{t \in [0.10, 0.89]}{\operatorname{argmax}} \ \text{Balanced-F1}(t) \qquad (1)$$

The search is conducted with a step size of 0.01.

Training Protocol. We implement Early Stopping based on the validation F1 score to prevent overfitting. The model weights from the best-performing epoch are restored for final testing. All experiments are managed via the Hydra configuration framework, with fixed random seeds to guarantee reproducibility on mid-to-high memory GPU environments.

C Detail Experimental Results

C.1 Interpreting Selected Layers

One of the key advantages of our method is its ability to identify the most interpretable and effective layers for downstream tasks. Table 9 lists the exact layer indices selected by SLAYERS at the optimal budget K for each model.

We observe distinct topological preferences across architectures:

- **Qwen-2.5:** The selection is highly distributed, picking layers from early, middle, and late

stages, suggesting a need for diverse feature abstraction levels.

- **Qwen-3 Dense:** The algorithm favors continuity, selecting a cluster of late-stage layers. This aligns with findings that dense models perform complex reasoning in deeper layers.
- **Qwen-3 MoE:** The selection is sparse but focused, efficiently identifying "expert-rich" layers that maximize the Mixture-of-Experts routing mechanism.

Table 9: **Specific Layer Indices Selected by Ours.** Indices are 0-indexed.

Model	Budget (K)	Selected Indices
Llama-3.1	5	[5, 24, 26, 29, 31]
Qwen-2.5	5	[14, 15, 17, 26, 27]
Qwen-3 Dense	6	[12, 27, 30, 31, 33, 35]
Qwen-3 MoE	6	[0, 3, 4, 20, 23, 31]

C.2 Statistical Significance of Random Baselines

To ensure that our performance gains are statistically significant and not artifacts of variance, we conducted extensive random search trials at budgets $K-1$, K, and $K+1$. Table 10 presents the aggregated statistics (Mean μ and Standard Deviation σ) for these random trials compared to the specific configuration identified by SLAYERS. For Qwen-2.5, our method exceeds the random distribution by a large margin (over 2σ), demonstrating high confidence. For Qwen-3 MoE, while the variance is lower, SLAYERS consistently performs above the mean, validating its robustness.

For LISA, the number of active layers (k) and target modules were configured to strictly align its effective trainable parameters with our approach. AdaLoRA params reflect its dynamic rank allocation overhead.

C.3 Implementation Details

All experiments were conducted on two NVIDIA A100 (80GB) GPUs. We used the `evalscope` framework developed by open-source evalscope framework for zero-shot evaluation on ARC, BBH, DROP, GSM8K, and RACE. For the UCB search algorithm, we set the exploration parameter $c = 2.0$ and the maximum number of trials to 450.

Table 10: **Statistical Comparison: Slayers vs. Random Distribution.** We report the average score across 5 benchmarks. Random baselines are derived from $N = 10$ independent trials.

Model	Experiment	Ours Score	Random Me
Qwen-2.5 ($K = 5$)	Budget $K - 1$ (Sub)	0.538	0.542
	Budget K (Opt)	**0.674**	**0.561**
	Budget $K + 1$ (Ext)	0.542	0.542
Qwen-3 MoE ($K = 6$)	Budget $K - 1$ (Sub)	0.767	0.780
	Budget K (Opt)	**0.794**	**0.783**
	Budget $K + 1$ (Ext)	0.787	0.771