GaDRA: Learning When Not to Apply LoRA in Replay-Free Continual Pre-Training

Anonymous ACL submission

Abstract

Continual pre-training (CPT) must absorb post-cutoff text without replay, which risks forgetting. Always-on LoRA is efficient but indiscriminate, applying residual updates at every token. We propose GaDRA, a gated dual-conditioned residual adapter that learns when to silence LoRA residuals. GaDRA applies a hard binary gate per position, conditioned on the frozen module output and the candidate residual, and trains gates jointly with LoRA factors via a straight-through Gumbel estimator. On replay-free CPT of Llama-3.1-8B-Instruct over two 2024 news corpora, GaDRA matches always-on LoRA in target knowledge acquisition while improving retention on math, code, and time-sensitive factual QA; similar gains hold on a Qwen3-8B cross-architecture probe. Mechanistic analyses suggest GaDRA learns layer- and module-wise activation budgets that preserve acquisition, whereas a single global budget degrades it.

1 Introduction

Updating an instruction-tuned Large Language Model (LLM) with knowledge that postdates its training cutoff, and doing so without replay, defines a narrow target: the model must acquire the new corpus while disturbing as little prior behavior as possible. Post-cutoff content can be incorporated either at inference time via **retrieval-augmented generation (RAG)** (Lewis et al., 2020) or by updating model parameters through **continual pre-training (CPT)** on a target corpus. RAG avoids parameter updates, but it cannot reliably integrate newly retrieved information with pre-trained knowledge in multi-step reasoning. CPT can internalize the new corpus, but each parameter update can also perturb prior capabilities that the target corpus never exercises, creating a risk of **catastrophic forgetting**.

Replay mitigates forgetting, but it is often infeasible when the original pre-training or instruction-tuning data are unavailable, and full-parameter CPT is computationally prohibitive. This makes parameter-efficient fine-tuning (PEFT), especially LoRA, a practical alternative. However, despite the "learn-less, forget-less" property attributed to LoRA (Biderman et al., 2024), LoRA-based CPT can still exhibit substantial forgetting (Dou et al., 2024; Li and Lee, 2024). We argue that replay-free LoRA-CPT needs an adapter-side **activation budget**: a learned decision about which layer–module blocks may use the residual often, rarely, or not at all, beyond simply shrinking the residual. We instantiate this budget as **GaDRA** (**Ga**ted **D**ual-conditioned **R**esidual **A**dapter), a per-position hard binary gate over the LoRA residual that is jointly conditioned on the frozen base output and the candidate residual (Fig. 1). Beyond how to parameterize the LoRA update, the central question is when the residual should not be applied at all.

Existing LoRA-based preservation methods (Hu et al., 2022) strengthen either initialization (**MiLoRA** (Wang et al., 2025b); **LoRA-Null** (Tang et al., 2026)) or training-time regularization (**CLoRA** (Lu et al., 2025); **O-LoRA** (Wang et al., 2023)); in both cases, they constrain *what* low-rank residual is learned. Gating variants modulate update strength from one side of the residual, per module (Mao et al., 2022) or per token (Ou et al., 2025). However, these methods are motivated and evaluated primarily under supervised fine-tuning (SFT), leaving open whether their controls suffice in the replay-free CPT regime. Two properties distinguish replay-free CPT from SFT. First, raw-document next-token loss is **dense**: it updates boilerplate, already-modeled continuations, and genuinely new facts without the response-only mask that localizes SFT gradients. Second, the target is **knowledge-bound**: post-cutoff events must be internalized as facts, so failures cannot generally be recovered by applying a learned skill to a new instance (Gekhman et al., 2024; Allen-Zhu and Li,

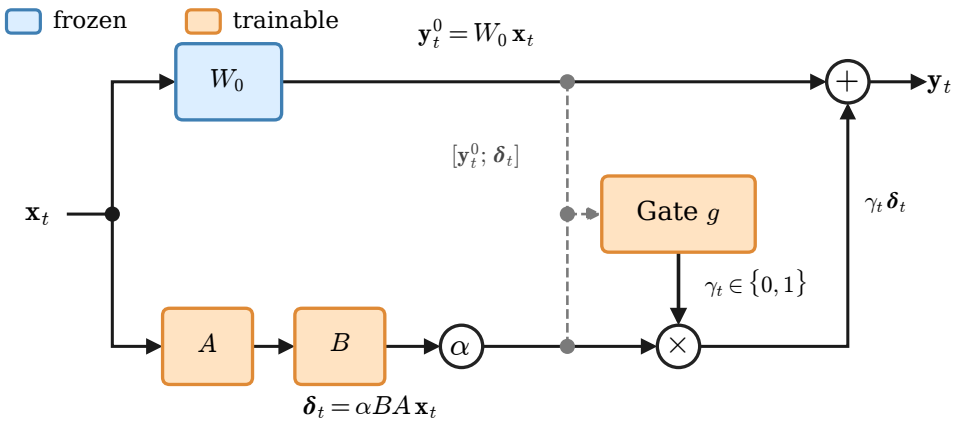

Figure 1: **GaDRA (conceptual), one adapted MLP projection at sequence position t.** A per-position hard gate $\gamma_t \in \{0, 1\}$, *dual-conditioned* on the frozen base output $\mathbf{y}_t^0$ and the LoRA residual $\boldsymbol{\delta}_t$, decides whether to apply the residual; when it closes (γ_t=0) the module output is bit-identical to the frozen base. Frozen and trainable components are colored. §4 formalizes the gate and its Gumbel straight-through training.

2024). We therefore hypothesize that preservation under replay-free CPT requires an adapter-side *activation budget*. This control primitive constrains both what residual is learned and how often the residual pathway is active, through per-position gates.

Our contributions are as follows:

1. **CPT-specific preservation framing.** We isolate two pressures that separate replay-free CPT from the SFT setting targeted by prior LoRA preservation work: dense next-token updates without a response-token mask, and knowledge-bound target acquisition. We use them to motivate an adapter-side activation budget: a learned per-position decision over residual activity whose dominant structure is a per-(layer × module) allocation (§6).

2. **GaDRA.** A hard, dual-conditioned per-position gate over the LoRA residual, trained jointly with the LoRA factors via a Gumbel straight-through estimator (§4).

3. **Empirical and mechanistic evidence.** GaDRA reaches target accuracy close to always-on LoRA and improves retention to near-best levels at substantially lower target cost. Gate overrides in our intervention cell argue against token-level routing: preserving each block's activation rate barely moves performance, while flattening the layer–module budget sharply hurts target acquisition.

2 Related Work

2.1 Forgetting in Continual Pre-Training of Instruction-Tuned LLMs

Prior work shows that continual pre-training on a target corpus can erode an aligned LLM's instruction-following and reasoning ability, even under small adaptation budgets (Li and Lee, 2024; Luo et al., 2023). Evidence from adapters is consistent with this trend. Pletenev et al. (2025) find that LoRA-based knowledge injection can hurt external QA even when training interleaves old and new facts, often regressing toward overrepresented and entity-biased answers. Mechanistic analyses attribute this fragility to how SFT leverages pre-trained knowledge. Gekhman et al. (2024) argue that SFT largely surfaces knowledge acquired during pre-training, whereas internalizing new facts requires repeated exposure (Allen-Zhu and Li, 2024). On the data side, forgetting can be mitigated by replay and learning-rate re-warming (Ibrahim et al., 2024), or by transforming target corpora into synthetic instruction data with dense repetition (Ovadia et al., 2025). Both are orthogonal to our replay-free, budget-limited, raw-document CPT setting.

2.2 PEFT Methods for Knowledge Preservation

Initialization-time and training-time PEFT preservation methods restrict the subspace in which LoRA updates can reside. **MiLoRA** (Wang et al., 2025b) initializes from minor singular vectors; **LoRA-Null** (Tang et al., 2026) constructs a calibration-set null-space initialization;

CLoRA (Lu et al., 2025) regularizes null-space update directions; and **O-LoRA** (Wang et al., 2023) enforces inter-task subspace orthogonality. These methods constrain the learned residual and were primarily evaluated under task-matched SFT.

GaDRA's adapter-side control is complementary: it makes per-(layer $\times$ module), per-position decisions trained jointly with the residual (§4), orthogonal to and composable with the subspace constraints above.

2.3 Gating and Conditional Adaptation in PEFT

A separate line of work introduces learned gates inside or around LoRA-style adapters. Some gates condition only on the adapter output. **UniPELT** (Mao et al., 2022), for example, gates the adapter contribution per module with a sigmoid scalar. Input-side or rank-internal variants gate different variables. **GateRA** (Ou et al., 2025) modulates LoRA per token from the input under SFT, while **SoRA** (Ding et al., 2023) controls sparsity within the rank dimension. A related budgeted-PEFT line allocates adapter capacity across ranks or layers (**AdaLoRA** (Zhang et al., 2023); **AdapterDrop** (Rücklé et al., 2021)). These set how much adapter capacity to spend or observe one side of the residual; GaDRA keeps capacity fixed and decides whether to apply it per position from both $\mathbf{y}_t^0$ and $\boldsymbol{\delta}_t$.

Multi-adapter methods instead gate *multiple* LoRA experts via learned routers. LoRAMoE (Dou et al., 2024), MoELoRA (Liu et al., 2023), AdaMoLE (Liu and Luo, 2024), and Mixture of LoRA Experts (Wu et al., 2024) route token representations to combinations of experts. These methods primarily choose *which* adapter fires, not whether to apply a single residual at a position.

Table 1 summarizes the design axes covered above.

Positioning. Tab. 1 shows GaDRA is the only entry that combines an output-side gate, dual conditioning on $(\mathbf{y}_t^0, \boldsymbol{\delta}_t)$, hard binary application, and per-position granularity under replay-free CPT. The closest one-sided baseline, UniPELT's per-module adapter-only gate lifted to a per-position hard gate, is our matched ablation **GaDRA-Mono** (§6). With a single expert, multi-adapter routers reduce to the same one-sided modulation.

3 Preliminaries

3.1 Replay-Free Continual Pre-Training

We adapt an instruction-tuned base LLM with frozen parameters θ_0 to a target raw-document corpus $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$. Training applies the dense next-token log-likelihood at every position of every document in $\mathcal{D}$, with no replay from the model's original pre-training or instruction-tuning distribution. Only PEFT adapter parameters are updated; θ_0 stays frozen. Retention is evaluated on a held-out instruction-conditioned QA suite disjoint from $\mathcal{D}$. Corpus construction details (paraphrase augmentation, languages) are in §5.1 and §A.7.

3.2 LoRA Decomposition

For a single adapted module with input dimension d_{in} and output dimension d_{out}, the LoRA adapter modifies the layer output at sequence position t as

$$\mathbf{y}_t = \mathbf{y}_t^0 + \boldsymbol{\delta}_t, \qquad \boldsymbol{\delta}_t = \alpha \cdot BA\,\mathbf{x}_t = \Delta W\,\mathbf{x}_t, \tag{1}$$

where $\mathbf{y}_t^0 = W_0\mathbf{x}_t \in \mathbb{R}^{d_{\text{out}}}$ is the frozen base contribution at the current hidden state $\mathbf{x}_t \in \mathbb{R}^{d_{\text{in}}}$, $\Delta W = \alpha BA \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is the low-rank update matrix in the standard LoRA notation (Hu et al., 2022), $B \in \mathbb{R}^{d_{\text{out}} \times r}$ and $A \in \mathbb{R}^{r \times d_{\text{in}}}$ are the trainable low-rank factors with $r \ll \min(d_{\text{in}}, d_{\text{out}})$, and α is the effective scalar output scale (it absorbs any rank normalization such as the conventional α_{LoRA}/r). Throughout, $(\mathbf{y}_t^0, \boldsymbol{\delta}_t)$ refers to this output-side pair at a single sequence position; per-module subscripts are suppressed.

4 Method

Static scaling forces a target–retention tradeoff. The LoRA residual (Eq. 1) exposes only the global output scale α. We sweep α on Llama-3.1-8B-Instruct continually pre-trained on BBC-News, holding rank, target modules, and optimization schedule fixed (LoRA at output scale α, a separate preliminary sweep; frozen base 6.00 BBC QA, 85.22 GSM8K). The sweep gives $\alpha = 0.5$: 65.75 / 69.22; $\alpha = 1.0$: 72.25 / 58.68; $\alpha = 4.0$: 75.50 / 5.00 (BBC QA / GSM8K). No single static α in this sweep gives a satisfactory target–retention tradeoff: $\alpha = 0.5$ achieves the best GSM8K retention but loses 6.50 pp of target accuracy relative to $\alpha = 1.0$, whereas $\alpha = 4.0$ achieves the best BBC QA but loses 64 pp of GSM8K relative to $\alpha = 0.5$. A static scale applies the same multiplier to every LoRA residual, without module- or

Method	Constraint form	Conditioning	Regime designed	Inference-time control
LoRA (Hu et al., 2022)	none	n/a	any fine-tuning	none
LoRA-Null (Tang et al., 2026)	null-space init from calibration	static (calibration-set)	task-matched SFT	none
MiLoRA (Wang et al., 2025b)	minor-singular init	static (frozen weights)	task-matched SFT	none
CLoRA (Lu et al., 2025)	null-space orth. regularization	static (calibration-set)	continual SFT	none
O-LoRA (Wang et al., 2023)	inter-task subspace orthogonality	static (per-task)	continual SFT	none
UniPELT (Mao et al., 2022)	adapter-output gate	adapter-only	SFT	learned (one-sided)
SoRA (Ding et al., 2023)	internal rank sparsity gate	rank-internal	SFT	learned (rank only)
GateRA (Ou et al., 2025)	input-side per-token gate	input-only	SFT	learned (one-sided)
GaDRA (ours)	**output-side hard gate**	**dual:** $(\mathbf{y}_t^0, \boldsymbol{\delta}_t)$	**replay-free CPT**	**learned (dual)**

Table 1: Representative single-residual LoRA-family methods. *Constraint form* is the residual restriction; *conditioning* is the gate input; *regime* is the original evaluation setting. Multi-adapter mixture-of-LoRA methods are excluded; see §2.3.

input-conditioned variation. These sweep runs are separate from the finalized main-table runs and are used only to expose the within-sweep trend.

We introduce a per-position decision of whether to apply the residual. This can be viewed as an activation budget jointly conditioned on the frozen base output and the candidate residual. We instantiate this idea as **GaDRA** (Gated Dual-conditioned Residual Adapter): an output-side hard binary gate that decides, at every adapted module and every sequence position, whether to apply the LoRA residual.

Gated LoRA. At sequence position t, GaDRA replaces the standard LoRA output $\mathbf{y}_t = \mathbf{y}_t^0 + \boldsymbol{\delta}_t$ (Eq. 1) with a gated form:

$$\mathbf{y}_t = \mathbf{y}_t^0 + \gamma_t \, \boldsymbol{\delta}_t, \qquad \gamma_t \in \{0, 1\}. \tag{2}$$

Here, γ_t is a scalar per-position gate obtained by hard-thresholding a router logit z_t. We compare two router families that differ only in the inputs to z_t:

$$z_t = \begin{cases} g(\boldsymbol{\delta}_t), & \text{adapter-only}, \\ g\big([\mathbf{y}_t^0;\, \boldsymbol{\delta}_t]\big), & \text{dual-conditioned}, \end{cases} \tag{3}$$
$$\gamma_t = \mathbf{1}[\sigma(z_t) > 0.5].$$

The gate acts at the output of the residual pathway and is defined per position, distinct from the per-module scale α in Eq. 1. For each adapted module, a single router $g(\cdot)$ is shared across all sequence positions within that module: positions differ only through the router inputs $(\mathbf{y}_t^0, \boldsymbol{\delta}_t)$, not through its parameters. With L decoder blocks and M adapted projections per block, the model carries $L \cdot M$ routers, one per LoRA adapter module. The gate is hard in the forward pass at both training and inference time. Below we justify the two remaining design choices in Eq. 3: dual conditioning and hard binary commitment.

Why dual and why hard. The dual-conditioned form $g([\mathbf{y}_t^0; \boldsymbol{\delta}_t])$ is needed because the gated module output is fully determined by the pair $(\mathbf{y}_t^0, \boldsymbol{\delta}_t)$: $\mathbf{y}_t^0$ indicates the preserved behavior at position t, while $\boldsymbol{\delta}_t$ indicates the proposed change. Conditioning on only one side cannot distinguish a needed correction from a harmful perturbation; the single-conditioned ablation $g(\boldsymbol{\delta}_t)$ (**GaDRA-Mono**, §5) averages over heterogeneous base states and learns a coarser policy. Using hard $\gamma_t \in \{0, 1\}$ makes the module output *bit-identical* to the frozen base wherever the gate closes ($\gamma_t = 0$); a soft $\gamma_t \in [0, 1]$ would leak a fraction of the residual through at every position. We test this dual $\times$ hardness interaction with the difference-in-differences (DID) factorial in Tab. 6.

Training. Hard binary gates have zero gradient and would block end-to-end optimization. We therefore train with a Gumbel-sigmoid straight-through estimator (Jang et al., 2017; Maddison et al., 2017; Bengio et al., 2013) at a fixed temperature $\tau = 1$ (no annealing). At each training step and position t, with independent Gumbel noise samples $(\eta_0, \eta_1) \sim \mathrm{Gumbel}(0, 1)$ for each $(t, \text{layer}, \text{module})$, the relaxed gate is

$$\widetilde{\gamma}_t = \sigma\left(\frac{z_t + \eta_1 - \eta_0}{\tau}\right), \qquad z_t = g\big([\mathbf{y}_t^0;\, \boldsymbol{\delta}_t]\big). \tag{4}$$

We binarize in the forward pass as $\gamma_t = \mathbf{1}[\widetilde{\gamma}_t > 0.5] \in \{0, 1\}$ and let gradients flow through the sigmoid relaxation (straight-through). At inference, we drop the Gumbel noise and recover the deterministic gate of Eq. 3. The router $g(\cdot)$ is a single affine projection, $g([\mathbf{y}_t^0; \boldsymbol{\delta}_t]) = \mathbf{w}_g^\top [\mathbf{y}_t^0; \boldsymbol{\delta}_t] + b_g$, with $\mathbf{w}_g \in \mathbb{R}^{2d_{\text{out}}}$ and $b_g \in \mathbb{R}$. We initialize b_g positive so gates start open (early training $\approx$ always-on LoRA) and learn to close as the LoRA factors

Corpus	#Art.	#Rew.	Train / Val
BBC-News	5,587	15	83,805 / 5,587
CC-zh-TW-News	5,340	14	74,760 / 5,340

Table 2: CPT corpora. #Art. = source articles, #Rew. = paraphrase rewrites per article; Train / Val are document counts.

Axis	Task	N	Protocol	Metric
Target	BBC QA	400	closed-book	judge acc†
	CCQA	250	closed-book	judge acc†
Retention	GSM8K	1,319	8-shot CoT	EM (maj@1)
	MBPP	500	3-shot	pass@1
	TiEBe	1,332	closed-book	judge acc†

Table 3: Evaluation tasks and protocols. † scored by a format-free GPT-4.1-mini semantic-sufficiency judge (§5.1).

A, B co-train. The LoRA scale α remains at its training value: γ_t provides the per-position on/off decision, while α controls the residual magnitude when $\gamma_t = 1$.

5 Experiments

5.1 Datasets and Evaluation

CPT corpora. We inject knowledge published after the base model's pretraining cutoff. Our primary backbone, Llama-3.1-8B-Instruct, has a December 2023 cutoff, so we use news from 2024. We build two replay-free CPT corpora that differ in naturalness and in how well they match the backbone's pretraining language (Tab. 2): (1) English 2024 BBC-News articles from the LatestEval monthly news collection (Li et al., 2024), and (2) Traditional Chinese 2024 news from the same period sourced from Common Crawl (Common Crawl Foundation, 2024). The second corpus probes cross-lingual generality while controlling for language-specific bias.

For each source article, Llama-3.1-405B-Instruct produces multiple rewrites from six instruction templates (15 for BBC-News, 14 for CC-zh-TW-News; §A.7); the original plus all but the last rewrite are training documents, and the final rewrite is held out for validation. This repeated exposure encourages next-token CPT to internalize content rather than surface form (Allen-Zhu and Li, 2024; Ovadia et al., 2024).

Evaluation tasks. We evaluate along two axes (Tab. 3) and **compare scores only within an axis**, not across axes. *Target tasks* measure acquisition of the injected 2024 knowledge. BBC QA and CCQA are closed-book QA sets generated from each corpus by GPT-4o (§A.6); each is a fixed random subset (N=400 and N=250, Tab. 3) drawn once before any training or model comparison. Because raw-document CPT and closed-book QA are different tasks, and the frozen bases score low on both target sets ("Frozen base" rows in Tab. 4 and Tab. 5), the large gains after CPT reflect closed-book recall of the injected content. The Llama base is near floor, so its gains cannot come from pre-existing knowledge; the Qwen3 base is somewhat higher and may include partial prior exposure, so we treat Qwen3 as a cross-architecture probe (§A.1).

Retention tasks measure preservation of prior capabilities. GSM8K (math) and MBPP (code) follow Meta's official Llama-3.1 evaluation configuration, including the few-shot counts and prompt templates from the `Llama-3.1-8B-Instruct-evals` release (Dubey et al., 2024; Cobbe et al., 2021; Austin et al., 2021), so the frozen Llama base reproduces Meta's reported scores. TiEBe (Almeida et al., 2025) is a time-sensitive factual-QA benchmark; we use its pre-2023 subset (events the base has already seen) to probe retention of time-anchored facts. BBC QA, CCQA, and TiEBe are scored by a single format-free GPT-4.1-mini semantic-sufficiency judge (Zheng et al., 2023). Per-language judge prompts are in §A.5.

5.2 Setup

Backbones and adapters. We use two frozen instruction-tuned backbones: Llama-3.1-8B-Instruct (32 decoder layers) as the primary model, and Qwen3-8B (Yang et al., 2025) (36 layers) as a cross-architecture probe. We insert rank-512, $\alpha = 1$ adapters into every MLP projection (up-, gate-, down-) in every decoder block. GaDRA additionally uses the dual-conditioned router family in Eq. 3. Full hyperparameters are in §A.1.

Baselines. On Llama we compare GaDRA with methods matched in rank, target modules, optimizer and schedule, data order, evaluation, seed, and step-0 function as the frozen base. They differ only in their published intervention: initialization, the frozen/trainable split, a regularizer, or a gate (design axes in Tab. 1):

- **LoRA**: the standard always-on adapter at $\alpha = 1$, our parity anchor.

Method	BBC-News				CC-zh-TW-News				Avg. Δ vs LoRA		
	BBC QA	GSM8K	MBPP	TiEBe	CCQA	GSM8K	MBPP	TiEBe	Δ_{tgt}	$\Delta_{\bar{r}}$	Overall
Frozen base	6.00	85.22	60.60	52.03	5.60	85.22	60.60	52.03	—	—	—
+ LoRA	72.75	58.30	45.80	24.47	**71.20**	38.13	31.80	15.09	—	—	—
+ MiLoRA	**74.50**	41.24	36.60	19.66	<u>70.39</u>	1.74	13.80	5.70	**+0.47**	−15.81	−7.67
+ LoRA-Null	70.25	24.18	32.20	17.87	67.60	2.58	1.20	4.73	−3.05	−21.81	−12.43
+ CLoRA	<u>73.50</u>	62.70	<u>48.00</u>	28.00	63.60	52.69	37.60	<u>25.45</u>	−3.43	+6.81	+1.69
+ GaDRA-Mono	69.25	**65.20**	**51.20**	**30.78**	64.40	<u>64.29</u>	**46.20**	25.30	−5.15	**+11.56**	<u>+3.21</u>
+ **GaDRA (ours)**	73.00	<u>63.76</u>	**51.20**	<u>29.73</u>	68.00	**64.44**	<u>43.60</u>	**27.32**	<u>−1.48</u>	<u>+11.08</u>	**+4.80**

Table 4: **Main results** on Llama-3.1-8B-Instruct (r=512): raw scores per corpus. The right block reports mean improvement over LoRA across corpora (Δ_{tgt} target; $\Delta_{\bar{r}}$ retention over GSM8K/MBPP/TiEBe; Overall = $(\Delta_{\text{tgt}} + \Delta_{\bar{r}})/2$; see §5.3). **Bold** / <u>underline</u> = best / second-best among adapted methods; *Base* and LoRA are anchors.

- **MiLoRA** (Wang et al., 2025b): adapters initialized from the minor singular components of each frozen weight.
- **LoRA-Null** (Tang et al., 2026): adapters initialized in the null space of pre-trained-knowledge activations, estimated from 256 NQ-Open (Kwiatkowski et al., 2019) calibration sequences disjoint from our corpora and evaluation, with both adapter factors trained.
- **CLoRA** (Lu et al., 2025): an orthogonal-subspace regularizer over $k = 512$ vectors.
- **GaDRA-Mono**: GaDRA's dual-conditioned gate replaced by the adapter-only gate $g(\boldsymbol{\delta}_t)$ of §4.

For MiLoRA and LoRA-Null, parity means a matched step-0 *function*: they residualize the frozen weight ($W_0' = W_0 - B_{\text{init}}A_{\text{init}} \neq W_0$), though the merged model reproduces W_0 at step 0 (§A.4). We use the same high rank for every method because the CPT-memorization premise (§3.1) needs sufficient adapter capacity: at $r = 128$ even plain LoRA loses substantial target accuracy (Tab. 10). The retention losses of MiLoRA and LoRA-Null here reflect them under this unified high-rank policy, not under-tuning. CLoRA's penalty diverges at its default λ in our high-rank MLP setting, so we use the largest stable value $\lambda = 10^{-3}$ (§A.4). We evaluate CLoRA after merging and all other methods as trained adapters; for Qwen3-8B we run only LoRA and GaDRA.

5.3 Main Results

Table 4 reports the six-method comparison on Llama, and Table 5 reports the Qwen3-8B probe (LoRA vs. GaDRA). To summarize acquisition and retention, we compute each method's improvement over the always-on LoRA baseline, averaged across the two corpora: Δ_{tgt} for target performance, $\Delta_{\bar{r}}$ for retention (the mean over GSM8K, MBPP, and TiEBe), and an *Overall* score $(\Delta_{\text{tgt}} + \Delta_{\bar{r}})/2$ that weights the two goals equally.

Acquisition vs. retention. All adapted methods learn the injected knowledge, scoring far above the near-floor Base. Their target scores remain within a few points of LoRA (averaged Δ_{tgt} from $+0.47$ to -5.15), but retention varies widely. GaDRA's target gap is within roughly one standard error of LoRA (§A.3), so we read its target as comparable and treat retention as the robust signal. The initialization-based methods (MiLoRA, LoRA-Null) largely preserve target performance but **forget catastrophically** at this rank, with averaged retention deltas of -15.81 and -21.81. By contrast, the gated and orthogonal methods (CLoRA, GaDRA-Mono, GaDRA) improve retention over LoRA.

GaDRA gives the best overall trade-off. Among the retention-improving methods, the key distinction is target cost. CLoRA and GaDRA-Mono gain retention with sizable target drops (averaged Δ_{tgt} of -3.43 and -5.15), whereas GaDRA stays close to parity (-1.48) while retaining nearly as much as GaDRA-Mono and more than CLoRA ($\Delta_{\bar{r}} = +11.08$). GaDRA-Mono retains marginally more ($+11.56$) but gives up 3.7 pp more target, which its adapter-only gate cannot recover. Overall ranks **GaDRA first** ($+4.80$), ahead of GaDRA-Mono ($+3.21$) and CLoRA ($+1.69$). §6 attributes GaDRA's target recovery to its dual conditioning.

Backbone transfer. The Qwen3-8B probe shows a consistent direction under an architecture swap: GaDRA trades a small target cost ($\Delta_{\text{tgt}} = -3.35$) for a large retention gain ($\Delta_{\bar{r}} = +12.82$), with Overall $= +4.73$, close to Llama. For Qwen3, the per-method optima coincide across methods on

Method	BBC-News				CC-zh-TW-News				Avg. Δ vs LoRA		
	BBC QA	GSM8K	MBPP	TiEBe	CCQA	GSM8K	MBPP	TiEBe	Δ_{tgt}	$\Delta_{\bar{r}}$	Overall
Frozen base	18.50	89.46	68.20	41.74	10.40	89.46	68.20	41.74	—	—	—
+ LoRA	**70.25**	35.48	32.60	11.79	**67.20**	68.31	50.60	15.69	—	—	—
+ GaDRA (ours)	66.75	**72.17**	**52.40**	**18.76**	64.00	**75.89**	**51.20**	**20.95**	**−3.35**	**+12.82**	**+4.73**

Table 5: **Supplementary** Qwen3-8B probe (LoRA vs. GaDRA). **Bold** = better adapted row; Avg. Δ columns follow Table 4. Optimizer settings are in §A.1.

Conditioning	Soft	Hard	Δ
Adapter-only	67.25	69.25	+2.00
Dual	65.50	**73.00**	+7.50

Table 6: Dual-conditioning × gate-hardness factorial on BBC QA target accuracy ($N = 400$, GPT-judged %). Columns compare soft ($\gamma \in [0, 1]$) with hard ($\gamma \in \{0, 1\}$), Δ = hard − soft. Rows compare adapter-only $g(\boldsymbol{\delta}_t)$ with dual $g([\mathbf{y}_t^0; \boldsymbol{\delta}_t])$ gating.

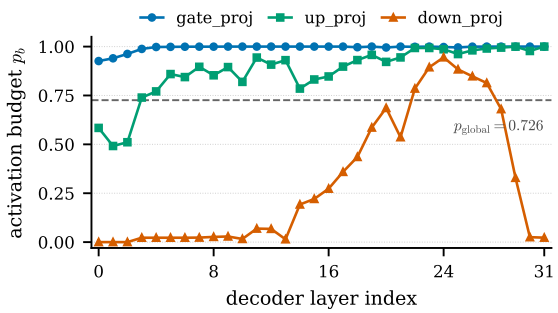

Figure 2: Learned GaDRA activation-budget profile p_b for Llama-3.1-8B-Instruct×BBC-News, estimated on a 64-example BBC QA probe. Lines are MLP module types over the 32 decoder layers; the dashed line is $p_{\text{global}} = 0.726$, used by the uniform-budget override in Row 3 of Tab. 7. This single-probe profile is descriptive.

BBC-News and differ on CC-zh-TW-News (§A.1). Qwen3 is a cross-architecture check, not the primary evidence.

6 Understanding GaDRA

The main results (§5.3) show that GaDRA acquires the injected knowledge nearly as well as the always-on LoRA baseline while substantially improving retention. This section asks why. We test whether it comes from a per-(layer × module) *activation budget* (which adapted modules remain active and how often), rather than increased adapter capacity or token-level, "surgical" routing. §6.1 tests whether dual conditioning and hard gating interact as predicted in §4; §6.2 intervenes on the learned gate to identify the load-bearing part of the budget and test update magnitude as an alternative. Unless noted, all analyses use the Llama-3.1-8B-Instruct×BBC-News cell.

6.1 Design Ablation: Do Dual Conditioning and Hard Gating Interact?

§4 makes two coupled commitments: the gate is dual-conditioned on $(\mathbf{y}_t^0, \boldsymbol{\delta}_t)$ and hard ($\gamma_t \in \{0, 1\}$). We expect hardening to help more when the gate can also observe the base state $\mathbf{y}_t^0$. We test this with a 2×2 factorial over conditioning and gate hardness on BBC-News CPT under the §5 protocol (Table 6); the two hard-gated cells are the GaDRA and GaDRA-Mono adapters in Table 4.

Hardening helps far more under dual conditioning (+7.50 pp) than adapter-only (+2.00 pp), with $\widehat{\text{DID}} = +5.50$ pp. This supports the §4 prediction that dual conditioning amplifies the benefit of hard gating. This factorial tests target acquisition; the gate-override study below diagnoses retention.

6.2 Gate Intervention Study

We probe why GaDRA helps with inference-only counterfactuals that leave the adapter weights untouched and replace only the gate signal γ_t at each adapted module. Let b index the 96 adapted blocks (32 decoder layers × 3 MLP projections) and t index non-padded positions. We summarize the learned gate by its per-block *activation rate* $p_b = \mathbb{E}_t[\gamma_{t,b}] \in [0, 1]$, collected into a profile $\mathbf{p} = (p_b)_{b=1}^{96}$ estimated on a 64-example BBC QA probe (mean $p_{\text{global}} = 0.7259$). As Fig. 2 shows, this budget is structured by module and depth: capacity concentrates in `gate_proj` (mean 0.99) and deeper `up_proj` (mean 0.88), while most `down_proj` blocks remain largely closed (mean 0.31). Thus p_b records where adapter capacity is spent, not which individual tokens matter. Each override in Tab. 7 targets one candidate explanation while holding the others fixed. If token-specific routing is essential, $\text{Bernoulli}(p_b)$ draws should hurt (Row 2); if only the total active rate matters, flattening p_b to p_{global} should not (Row 3); and

#	Setup	BBC QA	GSM8K	MBPP	TiEBe
1	**GaDRA-native**	73.00	63.76	51.20	29.73
2	Per-token shuffle	$71.50_{\pm 0.74}$	$65.98_{\pm 0.69}$	$49.07_{\pm 0.68}$	$27.00_{\pm 0.50}$
3	Uniform-budget	$60.33_{\pm 0.66}$	$67.85_{\pm 0.94}$	$50.33_{\pm 0.50}$	$31.01_{\pm 1.03}$
4	Gate transplant	$60.67_{\pm 0.82}$	$70.89_{\pm 0.50}$	$51.00_{\pm 0.71}$	$29.90_{\pm 0.55}$
5	**LoRA-native**	72.75	58.30	45.80	24.47

Table 7: γ-override study on fully trained adapters ($N = 400$ BBC QA; retention N as in §5.1). Rows replace the learned gate γ_t with the named counterfactual: shuffled $\mathrm{Bern}(p_b)$, flat $\mathrm{Bern}(p_{\mathrm{global}})$, transplanted GaDRA p_b, or $\gamma \equiv 1$; definitions are in the text. Stochastic rows (2–4) report mean $\pm$ std over three inference-time RNG seeds; deterministic rows (1, 5) are single runs. **Bold** row labels (1, 5) mark the two trained models, reproducing the BBC-News cells of Table 4.

if the budget is independent of the weights, transplanting GaDRA's p_b onto the always-on adapter should preserve the target (Row 4).

(a) Per-token placement is fungible given the block budget. Row 2 (*per-token shuffle*) draws $\gamma_{t,b}$ from $\mathrm{Bernoulli}(p_b)$ independently per token, preserving the 96-dimensional activation-rate profile while removing token-specific structure. Because Rows 2–4 are stochastic, each is averaged over three inference-time RNG seeds; the spread is small (std ≤ 1.1 pp on every axis), so the comparison is not driven by a single draw. Relative to Row 1, the per-token shuffle changes BBC-News target accuracy by only -1.50 pp and shifts retention by $+2.22/-2.13/-2.73$ on GSM8K / MBPP / TiEBe, small relative to the target drop under the uniform-budget override in (b). Given the same per-block budget, exact token placement is largely fungible in this cell: the trained gate does not act as a surgical, token-level router within a block.

(b) The per-block budget shape p_b is load-bearing for the target. Row 3 (*uniform-budget*) replaces p_b with $\mathrm{Bernoulli}(p_{\mathrm{global}} = 0.7259)$ uniformly across all 96 blocks. This keeps the same mean activation rate as Row 2 but removes the module- and depth-structured allocation in Fig. 2. BBC-News accuracy drops by 11.2 pp ($71.50 \rightarrow 60.33$), far beyond the across-seed std, while all retention metrics improve ($+1.87/+1.26/+4.01$). The mean activation rate is therefore insufficient; the 96-vector $\mathbf{p}$ specifies which adapted modules stay active and how often.

(c) The adapter must be co-adapted to its gate. Row 4 (*gate transplant*) applies GaDRA's p_b to the LoRA adapter, trained with $\gamma \equiv 1$. BBC-News drops to 60.67, below both GaDRA-native (73.00) and LoRA-native (72.75), while retention improves relative to LoRA-native ($58.30 \rightarrow 70.89$ GSM8K, $45.80 \rightarrow 51.00$ MBPP, $24.47 \rightarrow 29.90$ TiEBe). Conversely, overriding GaDRA's own gate to $\gamma \equiv 1$ lowers its BBC-News target from 73.00 to 69.00 (the symmetric counterfactual to Row 4). Decoupling gate from adapter hurts target in both directions, consistent with co-adaptation.

Capacity control. The effect is not explained by GaDRA learning smaller adapter factors. The per-module raw Frobenius norm $\|BA\|_F$ is nearly identical across methods (GaDRA 22.72, GaDRA-Mono 22.14, LoRA 21.48). The gated methods do not learn smaller updates; they learn updates shaped for their training-time gate regime. The target–retention difference in Table 4 is therefore consistent with the activation budget rather than a change in adapter capacity.

7 Conclusion

GaDRA shows that replay-free CPT can improve retention of instruction-tuned behavior, relative to always-on LoRA, by learning an activation budget for the LoRA residual, a hard binary gate conditioned on the frozen base output and candidate residual and trained jointly with the LoRA factors. On Llama-3.1-8B-Instruct across two target corpora, with a Qwen3-8B cross-architecture probe, GaDRA adapts to post-cutoff content comparably to always-on LoRA while substantially improving retention on math, code, and time-sensitive factual QA, with the best overall target–retention trade-off among the compared LoRA-CPT preservation variants. Our analysis indicates the benefit is not primarily explained by smaller updates or token-level routing in this cell; it depends on the learned per-(layer $\times$ module) budget profile, co-adapted with the adapter. This positions activation-budget control as a preservation primitive for replay-free CPT, orthogonal to initialization-based (MiLoRA, LoRA-Null) and regularization-based (CLoRA, O-LoRA) approaches.

GaDRA is explicitly designed for the dense next-token loss of raw-document CPT. Optimizing dynamic budgets for the distinct gradient distribution of response-masked SFT remains an open question. Finally, extending this gating mechanism to 70B+ scale models and investigating its interplay with Mixture-of-Experts (MoE) routing are critical next steps for scalable knowledge injection.

Limitations

Mechanism transfer is untested. The intervention analysis in §6 is conducted only on Llama-3.1-8B-Instruct×BBC-News. Although the Qwen3-8B probe (Tab. 5) shows the same target–retention trade-off under an architecture swap, we do not test whether the learned gate mechanism transfers across base architectures or pretraining language regimes. Here, the learned mechanism means the per-(layer × module) activation-budget profile and the co-adaptation between the gate and the adapters. Future work could add a second-cell mechanism probe that extracts p_b and applies a uniform-budget override on Qwen3×BBC-News or Llama×CC-zh-TW-News.

Continual pre-training scope. Our thesis focuses on replay-free CPT, and we do **not** claim that it extends to response-masked SFT. In SFT settings (XBRL formula generation (Wang et al., 2025a) and AusLaw citation (Han et al., 2026); full tables in Tabs. 12 and 13), GaDRA shows a similar directional effect: target gaps stay within a comparable range ($-1.44/-1.50$ EM), while GSM8K retention improves ($+7.58/+5.15$ EM). However, the SFT regime differs structurally, through a response-token loss mask and more skill-specific targets. We conjecture, but do not test, that these differences weaken the §4 motivation for an adapter-side budget primitive. We therefore treat the SFT cells as a scope check, not evidence for the CPT thesis. Optimizing dynamic activation budgets tailored specifically to the response-masked SFT gradient distribution remains an open question for future research.

Single training seed. Single-seed training is standard for LLM-scale CPT at 8B (Biderman et al., 2024; Ibrahim et al., 2024). We use the consistent cross-cell sign pattern in Tab. 4 as the primary robustness check for single-seed risk. Appendix A.3 reports evaluation-set binomial standard errors, which place the small GaDRA-vs-LoRA target gaps within sampling noise. Furthermore, the extremely low variance observed across stochastic gate overrides (Tab. 7) strongly mitigates the risk of seed-specific artifacts.

Judge protocol. Target QA (BBC QA, CCQA) uses an LLM judge, given the source document, scored for paraphrase-tolerant semantic sufficiency (§A.5): the model answers closed-book, and an answer counts as correct if it contains information sufficient to answer, even alongside extraneous content. These scores measure closed-book recall rather than strict factual consistency; the TiEBe judge additionally marks contradictions of the expected answer incorrect. We explicitly opted against strict n-gram metrics (e.g., Exact Match or ROUGE) for target QA, as they heavily penalize valid semantic recall—a well-documented issue when evaluating raw-document CPT. Nevertheless, we acknowledge that LLM judges carry inherent biases, which remains a general limitation of current long-form evaluation.

Scalability and MoE routing. Finally, our experiments focus on 8B-parameter dense models. Extending the dual-conditioned gating mechanism to larger-scale models (e.g., 70B+) and investigating its interplay with native Mixture-of-Experts (MoE) routers—specifically clarifying the boundary between token-level expert selection and residual activation budgets—are critical next steps toward highly scalable, replay-free knowledge injection.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2024. Physics of language models: Part 3.1, knowledge storage and extraction. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 1067–1077. PMLR / OpenReview.net.

Thales Sales Almeida, Giovana Kerche Bonás, João Guilherme Alves Santos, Hugo Abonizio, and Rodrigo Nogueira. 2025. TiEBe: A benchmark for assessing the current knowledge of large language models. ArXiv:2501.07482.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. ArXiv:2108.07732.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432.

Dan Biderman, Jacob P. Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. 2024. Lora learns less and forgets less. *Trans. Mach. Learn. Res.*, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. ArXiv:2110.14168.

Common Crawl Foundation. 2024. Common crawl.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. ULTRAFEEDBACK: boosting language models with scaled AI feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 9722–9744. PMLR / OpenReview.net.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. 2023. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4133–4145. Association for Computational Linguistics.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 1932–1945. Association for Computational Linguistics.

Abhimanyu Dubey and 1 others. 2024. The Llama 3 herd of models. ArXiv:2407.21783; evaluation configs from the Llama-3.1-8B-Instruct-evals release.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning llms on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 7765–7784. Association for Computational Linguistics.

Jiuzhou Han, Paul Burgess, and Ehsan Shareghi. 2026. Legal citation prediction with llms: a comparative evaluation of instruction tuning, retrieval, and jurisdiction-specific pre-training on the auslaw citation benchmark. *Artificial Intelligence and Law*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Gregory Anthony, Eugene Belilovsky, Timothée Lesort, and Irina Rish. 2024. Simple and scalable strategies to continually pre-train large language models. *Trans. Mach. Learn. Res.*, 2024.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chen-An Li and Hung-Yi Lee. 2024. Examining forgetting in continual pre-training of aligned large language models. *CoRR*, abs/2401.03129.

Yucheng Li, Frank Guerin, and Chenghua Lin. 2024. Avoiding data contamination in language model evaluation: Dynamic test construction with latest materials. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Source of the RealTimeData monthly BBC News collection. arXiv:2312.12343.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2023. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications. *CoRR*, abs/2310.18339.

Zefang Liu and Jiahua Luo. 2024. Adamole: Fine-tuning large language models with adaptive mixture of low-rank adaptation experts. *CoRR*, abs/2405.00361.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yuheng Lu, Bingshuo Qian, Caixia Yuan, Huixing Jiang, and Xiaojie Wang. 2025. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. ArXiv:2410.16801.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *CoRR*, abs/2308.08747.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Scott Yih, and Madian Khabsa. 2022. Unipelt: A unified framework for parameter-efficient language model tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6253–6264. Association for Computational Linguistics.

Jie Ou, Shuaihong Jiang, Yingjun Du, and Cees G. M. Snoek. 2025. Gatera: Token-aware modulation for parameter-efficient fine-tuning. *Preprint*, arXiv:2511.17582.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2024. Fine-tuning or retrieval? comparing knowledge injection in llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 237–250. Association for Computational Linguistics.

Oded Ovadia, Meni Brief, Rachel Lemberg, and Eitam Sheetrit. 2025. Knowledge-instruct: Effective continual pre-training from limited data using instructions. *Preprint*, arXiv:2504.05571.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Sergey Pletenev, Maria Marina, Daniil Moskovskiy, Vasily Konovalov, Pavel Braslavski, Alexander Panchenko, and Mikhail Salnikov. 2025. How much knowledge can you pack into a lora adapter without harming llm? In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, Findings of ACL, pages 4309–4322. Association for Computational Linguistics.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2020, Virtual Event / Atlanta, Georgia, USA, November 9-19, 2020*, page 20. IEEE/ACM.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7930–7946.

Pengwei Tang, Xiaolin Hu, Yong Liu, Lizhong Ding, Dongjie Zhang, Xing Wu, and Debing Zhang. 2026. Put the space of LoRA initialization to the extreme to preserve pre-trained knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*. ArXiv:2503.02659.

Dannong Wang, Jaisal Patel, Daochen Zha, Steve Y. Yang, and Xiao-Yang Liu. 2025a. Finlora: Benchmarking lora methods for fine-tuning llms on financial datasets. *Preprint*, arXiv:2505.19819.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. 2025b. MiLoRA: Harnessing minor singular components for parameter-efficient LLM finetuning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4823–4836.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Findings of ACL, pages 10658–10671. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Xun Wu, Shaohan Huang, and Furu Wei. 2024. Mixture of lora experts. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, 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)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. 2024. Opencodeinterpreter: Integrating code generation with execution and refinement. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Findings of ACL, pages 12834–12859. Association for Computational Linguistics.

A Implementation Details and Judge Prompts

A.1 Optimizer and Per-Cell Training Configuration

All four training cells (corpus $\times$ base-model) share the same optimization setup: AdamW (Loshchilov and Hutter, 2019), cosine learning-rate schedule, BF16 mixed precision, DeepSpeed ZeRO stage-2 (Rajbhandari et al., 2020), FlashAttention-2 (Dao, 2024), sequence length 1024, global batch size 128 (micro-batch 16), gradient clipping 0.1, weight decay 0.01, single training seed 42, and 3 epochs. We report inference at the fully trained adapter. Every method trains rank-512 adapters on all three MLP projections of each decoder layer, about 0.9B trainable parameters for Llama-3.1-8B (roughly 11% of the base); the gated methods add one affine router per adapted module, a negligible overhead. We select the learning rate and warmup ratio per method by best target accuracy in every cell (Tab. 8); on the Llama cells and the BBC-News$\times$Qwen3 cell the selected values coincide across methods, so the primary comparison is at matched hyperparameters, while on the supplementary CC-zh-TW-News$\times$Qwen3 cell they differ. The procedure is identical for every method, so no method receives a privileged search, and selection is on target accuracy rather than retention, so the retention gains are not an artifact of the search. We do not separately verify Qwen3-8B's pretraining cutoff; its higher zero-shot target score (Tab. 5) suggests partial prior exposure to the 2024 corpora, so we treat the Qwen3 cell as a cross-architecture probe rather than a clean post-cutoff injection.

Corpus	Base model	Methods	LR	Warmup
BBC-News	Llama-3.1-8B-Inst.	All	3×10^{-4}	0.05
CC-zh-TW-News	Llama-3.1-8B-Inst.	All	4×10^{-4}	0.20
BBC-News	Qwen3-8B	All	8×10^{-4}	0.20
CC-zh-TW-News	Qwen3-8B	LoRA	1×10^{-3}	0.10
CC-zh-TW-News	Qwen3-8B	GaDRA	6×10^{-4}	0.20

Table 8: Cell-specific learning rate and warmup ratio. "All" = one setting shared by every compared method in that corpus–base-model cell.

A.2 Reproducibility: Compute, Software, and Artifacts

Computational budget. Training runs on 2$\times$ NVIDIA H100-80GB. A single CPT run (one corpus $\times$ backbone, 3 epochs) takes about 4 hours of wall-clock time ($\approx$ 8 GPU-hours). Closed-book QA generation, the LLM judge, and corpus paraphrasing run on hosted APIs and consume no local GPU time.

Software. We build on PyTorch (Paszke et al., 2019) and HuggingFace Transformers (Wolf et al., 2020), with DeepSpeed ZeRO-2 (Rajbhandari et al., 2020) and FlashAttention-2 (Dao, 2024); the LoRA adapters and gates use our own implementation. Versions: PyTorch 2.6.0, Transformers 4.53.3.

Artifacts and licenses. Base models are Llama-3.1-8B-Instruct and Llama-3.1-405B-Instruct (Llama 3.1 Community License) and Qwen3-8B (Apache-2.0). Retention and calibration data are GSM8K, MBPP, TiEBe, and NQ-Open; the CPT corpora are built from 2024 BBC-News (via LatestEval) and Traditional Chinese Common Crawl news; the SFT scope-check uses FinLoRA (XBRL) and the AusLaw citation benchmark. Closed-book QA generation and judging use the hosted GPT-4o and GPT-4.1-mini models. We use each artifact for non-commercial research consistent with its intended use.

A.3 Evaluation-Set Uncertainty and Seeds

All main-table numbers use a single training seed (42; §A.1); the consistent cross-cell sign pattern (Tab. 4) is our robustness check (Limitations). For evaluation-set sampling alone, a judged accuracy p on N items has binomial standard error $\sqrt{p(1-p)/N}$: about 3.0 pp at the smallest set (CCQA, N=250), 2.3 pp for BBC QA (N=400), 2.2 pp for MBPP (N=500), and below 1.5 pp for GSM8K (N=1319) and TiEBe (N=1332). Across the four target cells, the GaDRA-vs-LoRA gap ranges from 0.25 to 3.50 pp; the standard error of the difference between two judged accuracies is $\sqrt{2}$ times the per-score values above (3–4 pp at the target N), so these gaps stay within roughly one standard error, consistent with the calibrated "comparable target" claim.

A.4 Baseline Implementation Details

Table 9 summarizes how each compared method intervenes on the frozen base, all at the unified rank $r = 512$ over the same MLP target modules. Every method starts from the same step-0 function as the instruction-tuned base; they differ in initialization, the frozen/trainable split, and any added regularizer or gate.

Method	Frozen wt.	Trainable	Aux. data	Added control
LoRA	W_0	A, B	—	—
MiLoRA	residual W_0'	A, B	SVD of W_0	—
LoRA-Null	residual W_0'	A, B	256 NQ-Open	—
CLoRA	W_0	A, B	k=512 vec.	orth. λ=10^{-3}
GaDRA-Mono	W_0	A, B, gate	—	gate $g(\boldsymbol{\delta})$
GaDRA	W_0	A, B, gate	—	gate $g(\mathbf{y}^0, \boldsymbol{\delta})$

Table 9: Baseline implementation at the unified rank $r = 512$. $W_0' = W_0 - B_{\text{init}}A_{\text{init}}$ is the residualized frozen weight used by the initialization-based methods.

Initialization-based methods. MiLoRA and LoRA-Null are init-then-residualize methods: the adapter is initialized to a fixed subspace of W_0 and the frozen weight is set to the complement $W_0' = W_0 - B_{\text{init}}A_{\text{init}}$, so that $W_0' + B_{\text{init}}A_{\text{init}} = W_0$ at step 0. MiLoRA uses the minor singular components of W_0; LoRA-Null projects W_0 onto the null space of the input activations of 256 NQ-Open sequences (length 1024). In our build the null-space reconstruction matches W_0 to a per-module error below 10^{-3}. We train both adapter factors A and B, matching the description in the LoRA-Null paper.

CLoRA regularization weight. CLoRA augments the language-modeling loss with an orthogonality penalty between the adapter factors and k pre-defined regularization vectors, weighted by λ. The penalty magnitude grows with r and k: at our $r = 512$, $k = 512$ the unscaled penalty is on the order of 10^3 times the language-modeling loss, so CLoRA's published default $\lambda = 1$ (set for low-rank attention adapters) diverges in our high-rank MLP CPT setting. Every $\lambda > 10^{-3}$ we tried diverged; we therefore use the largest stable value, $\lambda = 10^{-3}$, throughout.

A.5 LLM Judge Prompts

The judge is GPT-4.1-mini (temperature 0, single evaluation per item). Three prompt templates are used, keyed by the evaluation language of each task. All three templates produce a binary *Correct* / *Incorrect* verdict; the rule across all three is paraphrase-tolerant semantic sufficiency.

English prompt

System. "Based on the information in the 'Document', determine whether the 'Answer' contains content that correctly answers the 'Question'. Scoring rules: As long as the 'Answer' contains relevant information sufficient to answer the 'Question', even if it also includes incorrect or extraneous descriptions, it should be judged as 'Correct'. Only when the 'Answer' contains absolutely no content that answers the 'Question' should it be judged as 'Incorrect'. Please strictly respond only with 'Correct' or 'Incorrect', and do not output any other content."

User template. `{"Document": "{doc}", "Question": "{q}", "Answer": "{a}"}`

Labels. Correct / Incorrect.

Chinese prompt

System.
請根據以下 "文章內容" 的資訊，判斷 "答案"是否有包含能夠正確回答"問題"的內容。
評分規則：
只要"答案"中 有出現足以回答"問題"的相關資訊，即使同時包含錯誤或多餘的描述，也一律判斷為 "正確"。
只有在"答案"完全沒有包含能回答"問題"的內容時，才判斷為 "錯誤"。
請嚴格只回答 "正確" 或 "錯誤"，不要輸出其他內容。

User template. `{"文章內容": "{document}", "問題": "{question}", "答案": "{answer}"}`

Labels. 正確 (Correct) / 錯誤 (Incorrect).

TiEBe prompt

System. "I will provide a question, an expected answer, and the candidate's answer. Your task is to verify if the candidate's answer is correct. The expected answer is the ground truth, so if the candidate's answer contradicts the expected answer or refuses to answer, it is incorrect. Please strictly respond only with 'Correct' or 'Incorrect', and do not output any other content."

User template. `Question: "{q}"\nExpected answer: "{doc}"\nCandidate answer: "{a}"`

Labels. Correct / Incorrect.

A.6 CPT QA-Set Generation Prompt

The closed-book target QA sets are generated from the source articles by GPT-4o with the template below, which elicits one self-contained, discriminative factual question and a short ground-truth answer per article. BBC QA uses the English template; CCQA uses its Traditional Chinese counterpart. The 100-word question and 20-word answer caps keep items concise, and the self-containment rule removes article-referential phrasing such as "according to the article."

CPT QA-set generation prompt (GPT-4o)

System. "Act as a teacher designing a closed-book factual memory test based on the provided article. Design one discriminative question and its corresponding answer. **Rules:** (1) **Length Constraints**: The Question must be concise and important, not exceeding 100 words. The Answer must be extremely brief, strictly under 20 words. (2) **Depth & Factuality**: Focus on specific facts, figures, or statements that only someone who read the article would know. Avoid general knowledge. (3) **Contextual Clarity**: Do not use 'According to the article'. Instead, include necessary background (dates, names, or locations) within the question so it is self-contained, but keep it brief. (4) **Format**: Output only a JSON object."

Output format. `{"Question 1": "...", "Answer 1": "..."}`

A.7 CPT Corpus Paraphrasing Prompts

Both CPT corpora are built by paraphrasing each source article with **Llama-3.1-405B-Instruct** using the six instruction templates below, designed to vary surface form (lexical / syntactic / discourse-level paraphrase) while preserving factual content. The parenthetical multiplier $(\times k)$ on each template is the number of independent rewrites it produces (sampling at non-zero temperature). For BBC-News this yields $3+2+2+3+2+3 = 15$ rewrites per article; CC-zh-TW-News uses the same templates with one fewer rewrite (14 per article). The $(\times k)$ multipliers shown below are for BBC-News. In both corpora the original article plus all but the last rewrite form the training set, and the final rewrite is held out for validation. This many-way replication is the repeated-exposure structure CPT needs to internalize content rather than memorize surface form (Allen-Zhu and Li, 2024; Ovadia et al., 2024).

Paraphrasing instruction templates ($\times k$ = independent rewrites)

P1 $(\times 3)$ "Rewrite or restructure the following article while ensuring the original meaning remains the same."

P2 $(\times 2)$ "Rewrite or restructure the following article while ensuring the original meaning remains the same, but adapt it into a more approachable and friendly tone."

P3 $(\times 2)$ "Rewrite or restructure the following article while ensuring the original meaning remains the same. Since the audience consists of elementary school students, please use a more accessible and easy-to-understand tone."

P4 $(\times 3)$ "Rewrite or restructure the following article. Try to change the original order as much as possible while ensuring the meaning remains unchanged."

P5 $(\times 2)$ "Rewrite or restructure the following article while ensuring the original meaning remains the same. Aim to include the 'Subject, Action, Time, Location, and Object' in every sentence; if any information is missing, supplement it based on the surrounding context."

P6 $(\times 3)$ "Rewrite or reorganize the following text while preserving its original meaning. Attempt to include not only all explicit information but also implicit details reasonably inferred from the context in every sentence—specifically the six elements: Who, What, When, Where, Why, and How. The first pronoun in each sentence should clearly identify the subject to ensure completeness. Avoid making assumptions or adding information not supported by the text."

B Rank Ablation and the Unified-Rank Policy for Init-Based Baselines

B.1 Rank Ablation at $r = 128$ (BBC-News CPT, Llama-3.1-8B-Instruct)

Method	BBC QA	GSM8K	MBPP	TiEBe
LoRA at $r = 128$	54.75	69.52	51.20	32.06
GaDRA at $r = 128$	**63.50**	**71.19**	49.80	29.05
Δ vs. LoRA	$+8.75$	$+1.67$	-1.40	-3.01

Table 10: Rank ablation, BBC-News CPT, Llama-3.1-8B-Instruct.

C Loss-Support Asymmetry Between Response-Masked SFT and Dense CPT

We include this probe only to quantify the loss-support difference between response-masked SFT and dense, replay-free CPT; it is *not* evidence that GaDRA's gains arise from fine-grained token-level routing (§6 shows the dominant learned structure is the per-(layer $\times$ module) budget). SFT local-

izes loss to response tokens via a mask, whereas raw-document CPT applies loss uniformly across positions. Let C_ℓ denote the fraction of the layer-ℓ hidden-state gradient mass that falls on response-aligned positions during a single optimizer step. Holding the content fixed, we toggle only the loss mask on the same SFT sequences (Table 11).

Layer ℓ	C_ℓ (masked)	C_ℓ (dense)	Δ (pp)	Ratio
4	0.287 ± 0.176	0.101 ± 0.082	$+18.6$	$2.83\times$
16	0.491 ± 0.207	0.122 ± 0.088	$+36.9$	$4.03\times$
28	$\mathbf{0.642 \pm 0.203}$	0.150 ± 0.111	$\mathbf{+49.3}$	$\mathbf{4.29\times}$

Table 11: Response-only SFT loss mask vs. dense counterfactual (106 SFT sequences pooled across BBC QA, MetaMath (Yu et al., 2024), UltraFeedback (Cui et al., 2024), CodeFeedback (Zheng et al., 2024); layers $\{4, 16, 28\}$; $\pm$: pooled-domain s.d.).

The mask concentrates $2.83\times$–$4.29\times$ more gradient mass on response tokens than the dense counterfactual, and this amplification increases with depth: at $\ell = 28$, removing the mask shifts about 50 pp of gradient mass from response tokens to prompt tokens on the same sequences. Under the dense language-modeling loss of raw-document CPT, gradient mass is distributed approximately uniformly across position quartiles. This indiscriminate update pressure motivates an *activation budget* rather than always-on residual updates; it does not imply that per-token placement is the operative mechanism.

D Full 5-Method SFT Scope-Check Numbers

The Limitations section's scope-check paragraph reports the GaDRA-vs-LoRA deltas on XBRL and AusLaw SFT. These runs are a scope check for response-masked SFT rather than evidence that the CPT thesis extends to SFT: they reuse five of the six main-experiment methods (excluding GaDRA-Mono) and the same adapter configuration, and ask only whether the GaDRA-vs-LoRA direction persists under SFT. Below we report the same numbers for all five methods (LoRA, GaDRA, CLoRA, MiLoRA, LoRA-Null) at the paper's **unified rank** $r = 512$. Per the §5.2 unified-rank policy, MiLoRA and LoRA-Null are reported at $r = 512$ only, since their small-rank canonical configurations fail the CPT-memorization premise that motivates the unified-rank choice (Tab. 10). Their retention loss on the SFT cells below carries the same reading as in the main results: a property of these methods under the unified high-rank policy, not under-tuning.

D.1 XBRL formula generation

XBRL formula generation is a task from the FinLoRA benchmark (Wang et al., 2025a); the target metric is exact match on formula generation over $N = 835$ examples. We fine-tune Llama-3.1-8B-Instruct with response-masked SFT, reusing the same five-method subset described above (excluding GaDRA-Mono) and the same $r = 512$ MLP-target adapter configuration, and evaluate the fully trained adapter (Tab. 12).

D.2 AusLaw citation generation

AusLaw citation generation is a task from Han et al. (2026); the target metric is exact match on citation generation over $N = 1000$ examples. The response-masked SFT recipe, the five methods, and the $r = 512$ adapter configuration match the XBRL setup above (Tab. 13).

D.3 Summary

Relative to always-on LoRA, GaDRA changes the XBRL target score by -1.44 EM while improving GSM8K retention by $+7.58$ EM, with the other retention axes near parity. On AusLaw, the corresponding changes are -1.50 EM on the target task and $+5.15$ EM on GSM8K retention, again with the other retention axes near parity. CLoRA at the unified rank is competitive with GaDRA on these SFT settings. MiLoRA and LoRA-Null show substantial retention loss at the unified rank; per Tab. 10, this reflects these methods under the unified high-rank policy, not a reporting artifact.

Method	Target EM	GSM8K (retention)	MBPP (retention)	TiEBe (retention)
LoRA	**99.64**	76.88	**54.00**	52.48
GaDRA	98.20	**84.46**	52.40	**52.78**
CLoRA	94.49	81.05	49.20	51.73
MiLoRA *(at unified rank r=512)*	100.00	9.40	14.20	27.33
LoRA-Null *(at unified rank r=512)*	100.00	2.65	3.20	41.22

Table 12: XBRL SFT, $r = 512$, fully trained adapter. Target: EM on formula generation ($N = 835$). Retention: GSM8K 8-shot EM, MBPP pass@1 ($\times 100$), TiEBe pre-2023 0-shot GPT-judged. Bold marks the best value per column among the viable methods (LoRA, GaDRA, CLoRA); MiLoRA and LoRA-Null are shown at the unified rank for completeness and excluded from bolding, as their retention collapse at this rank is the behavior under diagnosis.

Method	Target EM	GSM8K (retention)	MBPP (retention)	TiEBe (retention)
LoRA	**51.60**	75.97	52.00	47.52
GaDRA	50.10	**81.12**	**52.80**	49.17
CLoRA	49.80	80.44	52.20	**49.47**
MiLoRA *(at unified rank r=512)*	54.10	49.13	46.80	32.21
LoRA-Null *(at unified rank r=512)*	55.50	18.73	44.60	22.15

Table 13: AusLaw SFT, $r = 512$, fully trained adapter. Target: EM on citation generation ($N = 1000$). Retention as above. Bold convention as in Tab. 12.