DUALKL: DISTRIBUTIONAL TRUST REGION FOR CONTINUAL LORA ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual adaptation requires large language models to learn new tasks without losing capabilities acquired from earlier ones. Existing LoRA-based methods mainly reduce interference in parameter or gradient space, while replay and distillation often apply the same preservation strength to all previous tasks. These strategies have important limitations. Parameter-space separation does not directly guaranty stable model outputs, average drift can hide severe forgetting on an individual task, and a fixed regularization strength cannot provide different levels of protection for tasks with different degrees of vulnerability. We propose **DualKL**, a task-wise primal–dual approach that preserves previous behavior through predictive-distribution trust regions. After each task, DualKL stores a small set of representative examples together with the model's output distributions. When learning later tasks, it measures the Kullback–Leibler (KL) divergence between the stored and current distributions for each previous task. Every task is assigned an independent constraint and an adaptive dual variable. The protection strength increases when a task exceeds its allowed drift and remains weak when the task is stable, allowing the model to retain earlier knowledge without unnecessarily restricting new-task learning. We evaluate DualKL on Standard CL, the complete Long Sequence benchmark, and TRACE across multiple language-model architectures. The results show that DualKL consistently improves overall performance and reduces both average and worst-task forgetting compared with replay, parameter-space methods, and fixed functional regularization. These findings demonstrate that selective control in function space is an effective approach for continual language-model adaptation. Our code are available at https://anonymous.4open.science/r/DualKL.

1 INTRODUCTION

Parameter-efficient fine-tuning makes it practical to specialize large language models without updating the full pretrained backbone. LoRA freezes the original model parameters and learns low-rank residual matrices in selected projections Hu et al. (2022). Recent surveys have summarized the rapid development of LoRA and other parameter-efficient adaptation methods Mao et al. (2025); Coleman et al. (2026). Their small trainable footprint is particularly useful when a model must learn a sequence of tasks after deployment. However, continually updating one shared adapter creates a **stability and plasticity dilemma**. Learning a new task may change behaviors required by previously learned tasks.

Continual learning has been studied extensively across classification, foundation models, and large language models. General surveys organize existing methods around replay, regularization, architectural isolation, and optimization-based interference control De Lange et al. (2022). More recent surveys focus specifically on continual learning for large language models and foundation models, including parameter-efficient adaptation, instruction tuning, knowledge updating, and lifelong model evolution Wu et al. (2024); Shi et al. (2025); Zheng et al. (2025); Yang et al. (2025). Despite this growing body of work, how to preserve task-specific predictive behavior while continuing to update one shared LoRA adapter remains insufficiently resolved.

Figure 1 reveals a persistent stability and plasticity dilemma in existing continual LoRA methods. Sequential LoRA retains strong learning ability on newly introduced tasks but incurs substantial forgetting. Parameter- and subspace-based methods gradually reduce forgetting, as indicated by the downward trajectory, yet they provide little improvement along the learning-accuracy dimension.

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

Consequently, none of these methods reaches the desired region that simultaneously requires stronger new-task acquisition and lower old-task degradation. This observation suggests that controlling parameter updates alone is insufficient to resolve the dilemma.

Parameter-space separation does not directly guarantee behavioral stability because the effect of an update depends on the input-specific functional geometry Doan et al. (2021); Lin et al. (2023); Ding et al. (2024). Recent continual LoRA methods reduce interference through orthogonalization, decomposition, or gradient projection Wang et al. (2023a); Liao et al. (2025); Wang et al. (2025); Das Biswas et al. (2026), but an orthogonal or small parameter update may still alter old predictions. This motivates controlling task-specific predictive drift directly.

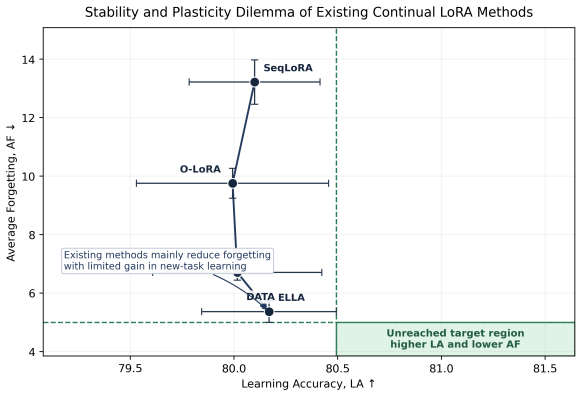


Figure 1: Stability–plasticity trade-off on Long Sequence. Existing methods mainly reduce forgetting without reaching the desired region of high learning accuracy and low forgetting.

Functional regularization and historical logit replay preserve old behavior more directly by matching previous model outputs Titsias et al. (2020); Buzzega et al. (2020). However, existing functional objectives often combine all previous tasks into one penalty. If most tasks remain stable, severe drift on one vulnerable task can be hidden by the average. This problem becomes more serious as the task sequence grows. A fixed regularization coefficient also applies the same protection strength to every old task, even though tasks differ in uncertainty, predictive margin, and sensitivity to new updates. Strong regularization may reduce current-task learning, while weak regularization may fail to protect vulnerable tasks.

We propose **DualKL**, a task-wise primal–dual method for continual LoRA adaptation. After each task, it stores compact anchors and their predictive distributions. During subsequent training, every previous task receives an independent KL-drift constraint and dual variable. Constraint violations increase the corresponding protection strength, whereas stable tasks impose little restriction on new-task learning.

We evaluate DualKL on Standard CL, the complete fifteen-task Long Sequence benchmark, and TRACE using several language model architectures. The results show that task-wise distributional control improves overall performance and reduces both average and worst-task forgetting. Controlled comparisons further show that task identity and adaptive dual control provide complementary benefits over fixed functional regularization.

Our contributions are threefold: (1) we formulate continual LoRA adaptation as task-wise predictive-distribution constraints; (2) we introduce independent dual variables that adapt protection to task-specific violations; and (3) we connect KL drift to anchor-level decision preservation and validate the design across short, long, and generative task streams.

2 Related Work

Parameter-efficient continual adaptation. Continual LoRA methods reduce interference through orthogonalization, parameter decomposition, gradient projection, or selective de-correlation Wang et al. (2023a); Liao et al. (2025); Wang et al. (2025); Das Biswas et al. (2026); Progressive Prompts instead expands task-specific parameters Razdaibiedina et al. (2023). These approaches define stability through update geometry or parameter isolation. DualKL instead permits overlapping updates when their task-specific predictive effects remain within budget.

Functional preservation and constrained learning. Learning without Forgetting distills previous predictions Li & Hoiem (2018), Dark Experience Replay stores historical logits Buzzega et al. (2020),

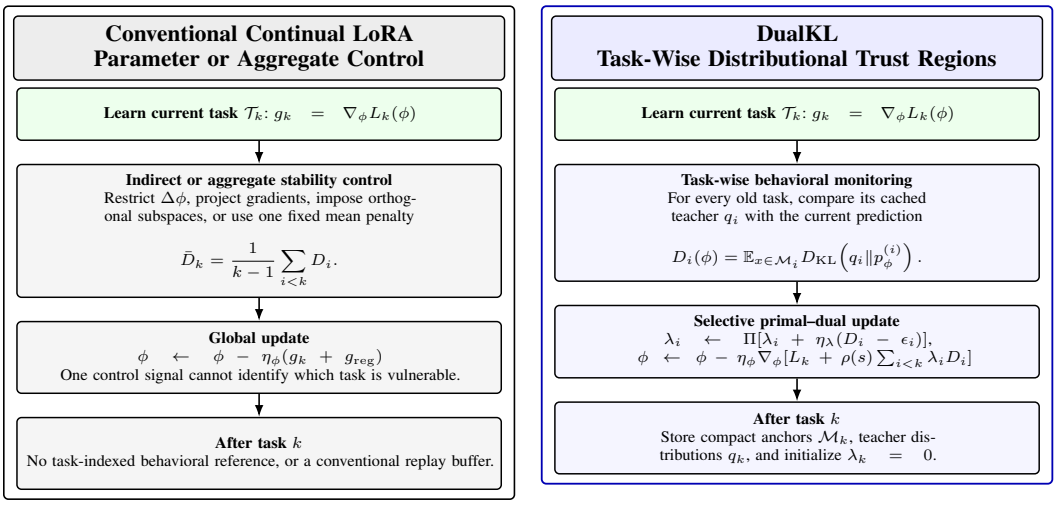


Figure 2: Conventional continual LoRA methods control parameter movement or use one aggregate preservation signal. DualKL instead monitors each old task in predictive-distribution space and adapts its protection through an independent dual variable.

and functional regularization constrains distributions at selected inputs Titsias et al. (2020). Primal–dual continual learning balances constraints and memory allocation Elenter et al. (2023), while TRGP defines trust regions in gradient subspaces Lin et al. (2022). DualKL combines these ideas through task-indexed predictive constraints and independent dual variables for one shared LoRA adapter.

3 METHOD

Figure 2 contrasts DualKL with conventional continual LoRA methods. Existing methods mainly restrict parameter updates or apply one aggregate preservation loss. DualKL instead monitors the predictive behavior of every previous task and adjusts its protection strength separately.

3.1 PROBLEM SETUP

We consider a sequence of supervised tasks $\mathcal{T}_{1:K} = \{\mathcal{T}_1, \ldots, \mathcal{T}_K\}$. Task $\mathcal{T}_i$ provides

$$\mathcal{D}_i = \left\{(x_j^{(i)}, y_j^{(i)})\right\}_{j=1}^{n_i}.$$

At stage k, the model learns from the current dataset $\mathcal{D}_k$ and a compact memory from previous tasks.

We use a frozen pretrained language model with parameters θ_0 and continually update one shared LoRA adapter ϕ. For a pretrained projection W_0, LoRA applies

$$W = W_0 + \frac{\alpha}{r} BA, \tag{1}$$

where A and B are trainable low rank matrices.

Controlling the size or direction of the adapter update does not directly control the model output. The same parameter change may have different effects on different inputs because the local response depends on the input specific Jacobian. We therefore measure forgetting in predictive distribution space rather than parameter space.

3.2 Task Indexed Distributional Memory

For a closed label task $\mathcal{T}_i$ with candidate label set $\mathcal{Y}_i$, we score a candidate label c using length normalized log likelihood

$$s_\phi(c \mid x) = \frac{1}{|c|} \sum_{\ell=1}^{|c|} \log p_{\theta_0,\phi}\left(c_\ell \mid x, c_{<\ell}\right). \tag{2}$$

The scores are normalized over the task label space

$$p_\phi^{(i)}(c \mid x) = \frac{\exp\left(s_\phi(c \mid x)/\tau\right)}{\sum_{c' \in \mathcal{Y}_i} \exp\left(s_\phi(c' \mid x)/\tau\right)}, \tag{3}$$

where τ is the temperature.

After completing task $\mathcal{T}_i$, we select a compact, label balanced anchor set

$$\mathcal{M}_i \subset \mathcal{D}_i$$

and cache the model distribution at that stage

$$q_i(c \mid x) = p_{\phi_i}^{(i)}(c \mid x), \qquad x \in \mathcal{M}_i, \tag{4}$$

where ϕ_i denotes the adapter immediately after learning task $\mathcal{T}_i$.

During later training, the functional drift of task $\mathcal{T}_i$ is measured by

$$D_i(\phi) = \frac{1}{|\mathcal{M}_i|} \sum_{x \in \mathcal{M}_i} D_{\mathrm{KL}}\left(q_i(\cdot \mid x) \,\|\, p_\phi^{(i)}(\cdot \mid x)\right). \tag{5}$$

Unlike hard label replay, the cached distribution preserves uncertainty and relative preferences among labels. Unlike parameter regularization, $D_i(\phi)$ directly measures changes in old task behavior.

For open ended generation, we apply the same idea to teacher forced token distributions and average the divergence over response positions. The complete generation formulation is provided in the appendix.

3.3 Task Wise Distributional Trust Regions

A common functional regularizer controls only the average drift

$$\overline{D}_k(\phi) = \frac{1}{k-1} \sum_{i<k} D_i(\phi). \tag{6}$$

However, a small average does not guarantee that every old task remains stable. One task can drift substantially while the others remain close to their cached distributions.

DualKL therefore assigns one constraint to each previous task

$$\begin{aligned} \min_{\phi} \quad & L_k(\phi) \\ \text{s.t.} \quad & D_i(\phi) \le \epsilon_i, \qquad i < k, \end{aligned} \tag{7}$$

where L_k is the current task loss and ϵ_i is the allowed drift budget.

The difference between average and task wise control becomes more important as the task sequence grows. If

$$\frac{1}{k-1} \sum_{i<k} D_i(\phi) \le \epsilon,$$

the worst individual drift may still be as large as $(k-1)\epsilon$. In contrast, the task wise constraints in Eq. (7) directly guarantee

$$\max_{i<k} D_i(\phi) \le \max_{i<k} \epsilon_i.$$

This motivates preserving task identity instead of collapsing all previous tasks into one functional penalty.

3.4 Adaptive Primal Dual Control

The Lagrangian of Eq. (7) is

$$\mathcal{J}_k(\phi, \lambda) = L_k(\phi) + \sum_{i<k} \lambda_i \left(D_i(\phi) - \epsilon_i \right), \qquad \lambda_i \geq 0. \tag{8}$$

Each old task has an independent dual variable λ_i that represents its current protection strength.

After a short current task warm up, the LoRA parameters are updated by

$$\phi \leftarrow \phi - \eta_\phi \nabla_\phi \left[L_k(\phi) + \rho(s) \sum_{i<k} \lambda_i D_i(\phi) \right], \tag{9}$$

where $\rho(s)$ gradually increases from zero to one.

The dual variables are updated through projected ascent

$$\lambda_i \leftarrow \Pi_{[0,\lambda_{\max}]} \left[\lambda_i + \eta_\lambda \left(D_i(\phi) - \epsilon_i \right) \right]. \tag{10}$$

If task i exceeds its drift budget, λ_i increases and gives that task a stronger corrective gradient. If the task remains stable, its dual weight stays small or decreases. DualKL therefore protects vulnerable tasks selectively instead of applying one fixed regularization strength to all previous tasks.

During the initial warm up steps, we optimize only L_k and keep the dual variables fixed. This prevents strong preservation constraints from blocking early current task learning. After task $\mathcal{T}_k$ is completed, we store $\mathcal{M}_k$ and q_k and initialize $\lambda_k = 0$.

3.5 Theoretical Connection to Decision Preservation

Task-wise KL constraints directly bound distributional change, but the ultimate evaluation of a classification task depends on predicted labels. The following result relates the two quantities at an anchor.

Proposition 1 (Anchor-level decision preservation)**.** *For anchor $x \in \mathcal{M}_i$, let*

$$c^* = \arg\max_{c \in \mathcal{Y}_i} q_i(c \mid x) \tag{11}$$

and define the teacher margin

$$\gamma_i(x) = q_i(c^* \mid x) - \max_{c \neq c^*} q_i(c \mid x). \tag{12}$$

If

$$D_{\mathrm{KL}} \left(q_i(\cdot \mid x) \,\|\, p_\phi^{(i)}(\cdot \mid x) \right) < \frac{\gamma_i(x)^2}{2}, \tag{13}$$

then

$$\arg\max_{c \in \mathcal{Y}_i} p_\phi^{(i)}(c \mid x) = c^*. \tag{14}$$

By Pinsker's inequality,

$$\mathrm{TV} \left(q_i, p_\phi^{(i)} \right) \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}} \left(q_i \| p_\phi^{(i)} \right)}.$$

Under Eq. (13), total variation is smaller than $\gamma_i(x)/2$. The teacher probability of c^* cannot decrease enough, and no competing label can increase enough, to close the original margin.

Proposition 1 guarantees agreement with the cached teacher on the given anchor; it does not imply that the teacher itself is correct or that a finite anchor bank perfectly represents the entire old-task distribution.

4 Experiment

Benchmarks. We evaluate on Standard CL, the complete 15-task Long Sequence benchmark, and the eight-task generative TRACE benchmark Razdaibiedina et al. (2023); Wang et al. (2023a;b). Each benchmark uses three task orders, including stress-test permutations for Long Sequence. Complete task lists and orders are provided in the appendix.

Method	AP ↑	AIA ↑	BWT ↑	FWT ↑	AF ↓	WAF ↓	LA ↑
SeqLoRA	67.76±0.86	74.57±0.95	−13.22±0.76	0.72±0.02	13.22±0.76	19.64±1.23	80.10±0.32
O-LoRA-style	70.89±0.46	76.00±0.71	−9.76±0.51	0.56±0.03	9.76±0.51	14.87±1.00	79.99±0.46
DATA	73.76±0.40	77.33±0.74	−6.70±0.26	0.94±0.04	6.70±0.26	9.98±0.41	80.02±0.41
ELLA	75.16±0.34	78.07±0.77	−5.36±0.37	1.10±0.03	5.36±0.37	8.08±0.79	80.17±0.32
SeqLoRAReplay	74.55±0.59	77.65±0.92	−5.75±0.29	0.91±0.03	5.75±0.29	8.83±0.69	79.92±0.42
Mean Fixed KL	75.55±0.33	77.85±0.74	−4.21±0.21	0.88±0.03	4.21±0.21	6.25±0.34	79.48±0.25
Task-wise Fixed KL	77.00±0.41	78.66±0.78	−3.01±0.17	0.95±0.03	3.02±0.17	4.53±0.36	79.80±0.43
Scalar Dual-KL	76.49±0.41	78.38±0.79	−3.46±0.21	1.02±0.03	3.47±0.21	5.35±0.51	79.73±0.43
DualKL	**84.46±0.65**	**83.36±0.55**	**−1.68±0.09**	**1.17±0.04**	**1.73±0.10**	**2.54±0.19**	**86.03±0.63**

Table 1: Results on the complete Long Sequence benchmark, averaged over three task orders and five random seeds. Bold indicates the best method without replay. AP and AIA measure final and trajectory-wide performance. AF and WAF measure average and worst-task forgetting.

Models and optimization. Our main backbone is LLaMA-3.1-8B-Instruct with a frozen 4-bit backbone and bf16 LoRA parameters. Cross-architecture experiments use T5-Large and Qwen2.5-7B-Instruct. Main results average three task orders and five seeds; secondary backbones use three seeds. Full optimization and hardware details are provided in the appendix.

Metrics. Let $R_{k,i}$ be the task-i score after learning through task k, and b_i the frozen-backbone score. We report final average performance (AP), average incremental performance (AIA), backward transfer (BWT), forward transfer (FWT), average forgetting (AF), worst-task forgetting (WAF), and learning accuracy (LA):

$$\mathrm{AP} = \frac{1}{K}\sum_i R_{K,i}, \mathrm{AIA} = \frac{1}{K}\sum_k \frac{1}{k}\sum_{i\le k} R_{k,i}, \mathrm{BWT} = \frac{1}{K-1}\sum_{i<K}(R_{K,i} - R_{i,i}),$$

$$\mathrm{FWT} = \frac{1}{K-1}\sum_{i=2}^{K}(R_{i-1,i} - b_i), \mathrm{AF} = \frac{1}{K-1}\sum_{i<K}\left(\max_{j\ge i} R_{j,i} - R_{K,i}\right),$$

$$\mathrm{WAF} = \max_{i<K}\left(\max_{j\ge i} R_{j,i} - R_{K,i}\right), \mathrm{LA} = \frac{1}{K}\sum_i R_{i,i}.$$

TRACE additionally reports changes in general ability, instruction following, and safety relative to the initial aligned model. Statistical comparisons pair backbone, order, and seed and use sign-flip permutation tests with Holm correction.

5 Results and Analysis

5.1 Complete Long Sequence Evaluation

Table 1 reports the results on the complete 15-task Long Sequence benchmark. Figure 3 shows the average performance over all learned tasks after each training stage. SeqLoRA achieves strong learning accuracy on newly introduced tasks, but its performance decreases as more tasks are added. This confirms that a small trainable parameter set does not by itself prevent catastrophic forgetting. Mean Fixed KL substantially reduces forgetting, showing that predictive-distribution preservation is a stronger control signal than parameter movement alone.

Task-wise Fixed KL improves both AP and WAF over Mean Fixed KL. The largest improvement appears in WAF, which measures the most severely forgotten task. This supports the main motivation for task-wise trust regions. An average constraint can remain small even when one previous task has moved far from its stored predictive distribution. Separate constraints preserve task identity and prevent this vulnerable task from being hidden by stable tasks.

DualKL further improves over Task-wise Fixed KL and Scalar Dual-KL. The comparison with Task-wise Fixed KL shows the benefit of adapting the protection strength, while the comparison with Scalar Dual-KL shows that adaptation is not sufficient when all tasks are represented by one average constraint. DualKL performs best when task identity and adaptive dual control are used together.

Method	AP	AIA	BWT	FWT	AF
SeqLoRA	72.66±1.29	78.27±1.52	−13.42±1.56	0.72±0.06	13.42±1.56
O-LoRA	74.72±0.71	78.99±1.30	−9.95±0.96	0.54±0.08	9.95±0.96
DATA	77.36±0.68	80.34±1.16	−6.80±0.68	0.95±0.07	6.80±0.68
ELLA	78.55±0.59	80.91±1.13	−5.11±0.56	1.13±0.10	5.11±0.56
Mean Fixed KL	78.85±0.60	80.74±1.05	−3.99±0.41	0.89±0.08	3.99±0.41
Task-wise Fixed KL	80.02±0.38	81.51±0.99	−2.95±0.24	0.98±0.06	2.95±0.24
Scalar Dual-KL	79.56±0.75	81.17±1.06	−3.32±0.51	1.01±0.06	3.32±0.51
DualKL	**81.31±0.44**	**82.15±0.75**	**−1.53±0.20**	**1.17±0.05**	**1.53±0.20**

Table 2: Standard CL results averaged over three task orders and five random seeds.

Method	Target AP ↑	BWT ↑	General Δ	Instruction Δ	Safety Δ
SeqLoRA	44.02±0.90	−17.08±1.18	−8.7±0.3	−7.9±0.3	−3.8±0.2
SeqLoRA+Replay	55.60±0.80	−3.44±0.24	−4.2±0.3	−3.4±0.3	−1.7±0.2
ELLA	54.52±0.54	−4.81±0.35	−3.6±0.3	−2.8±0.3	−1.2±0.2
Mean Fixed KL	55.40±0.26	−3.13±0.19	−3.1±0.3	−2.5±0.3	−1.1±0.2
Task-wise KL	56.67±0.13	−2.24±0.12	−2.7±0.3	−2.1±0.3	−0.9±0.2
Scalar Dual-KL	56.15±0.65	−2.59±0.14	−2.9±0.3	−2.2±0.3	−1.0±0.2
DualKL	**58.02±0.49**	**−1.01±0.04**	**−1.8±0.3**	**−1.4±0.3**	**−0.5±0.2**

Table 3: TRACE target-task performance and changes in general ability, instruction following, and safety. Capability changes closer to zero indicate better preservation.

The learning accuracy of DualKL remains comparable to that of SeqLoRA. Its retention gain therefore does not result from refusing to learn new tasks. Figure 3 shows that the difference emerges gradually in the later stages, when more old-task trust regions become active.

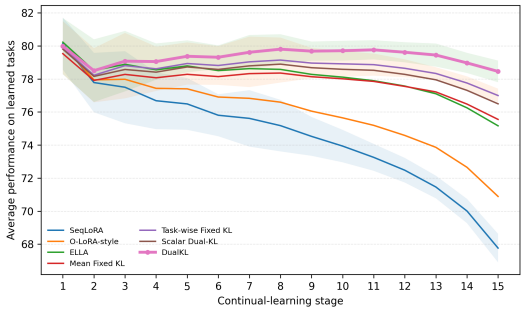


Figure 3: Average performance over all learned Long Sequence tasks after each training stage.

5.2 Standard CL Evaluation

Table 2 reports the results on Standard CL. DualKL retains the same method ordering on the shorter Standard CL stream. The gap between mean and task-wise constraints is smaller than on Long Sequence, as expected when fewer stable tasks are available to mask an individual constraint violation.

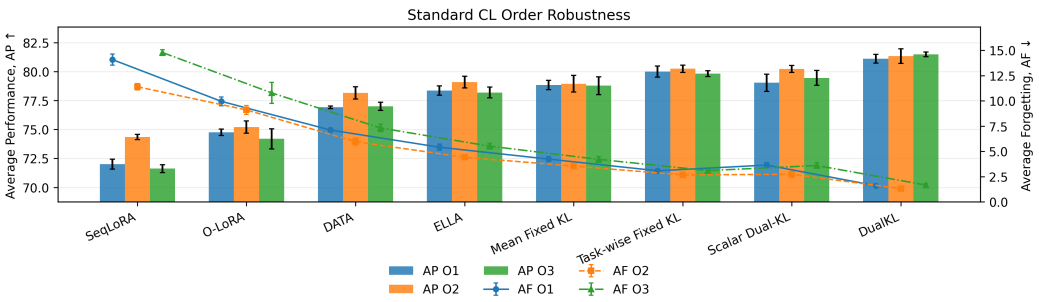
5.3 Generative Continual Learning on TRACE

Table 3 evaluates target-task retention and broader capability changes on TRACE. DualKL improves target-task AP and BWT while better preserving general ability, instruction following, and safety. This extends the trust-region principle beyond closed-label classification: token-level distributions retain behavioral information absent from hard-label replay.

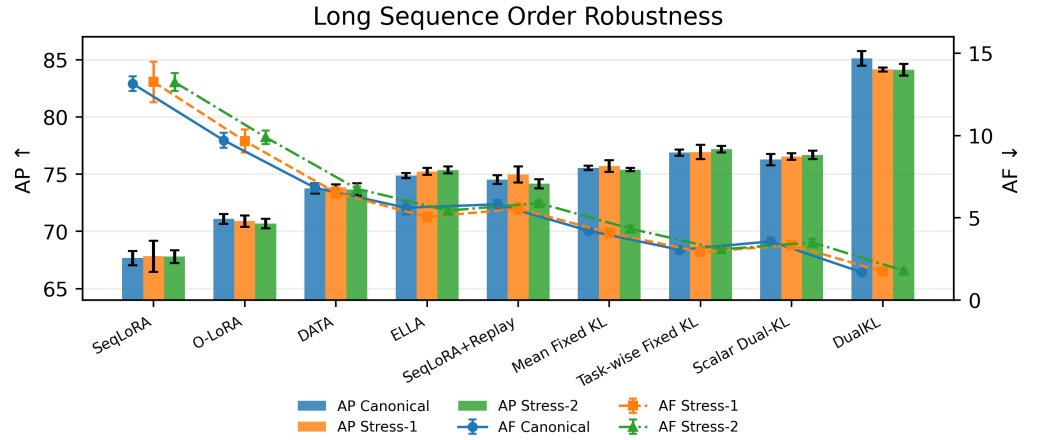
5.4 Cross-Backbone Generalization

Table 5 evaluates DualKL with T5-Large and Qwen2.5-7B-Instruct.

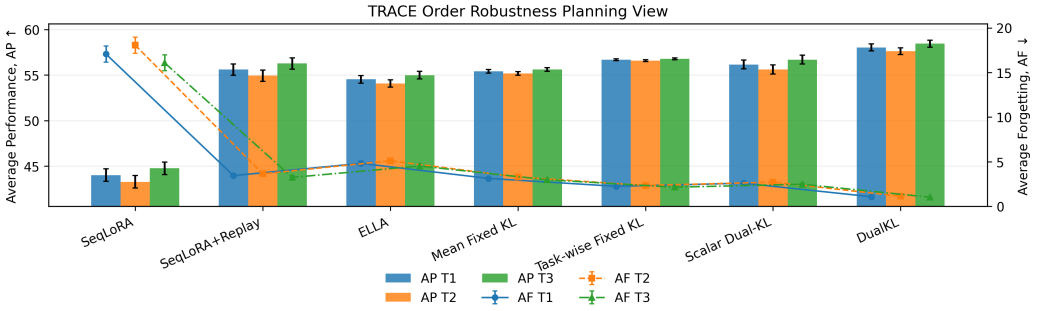
DualKL improves both AP and forgetting on the encoder-decoder T5-Large model and the decoder-only Qwen model. The consistent direction across architectures is important because the trust-region controller is defined on predictive distributions rather than on a particular adapter geometry. The method therefore does not require the same hidden representation, attention structure, or parameter sensitivity across backbones.



(a) Standard CL



(b) Long Sequence



(c) TRACE

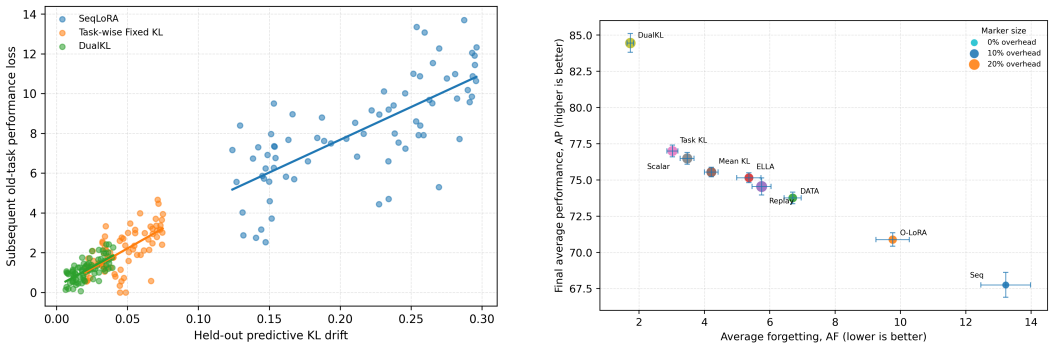
Figure 4: Order-wise average performance and forgetting across three continual learning benchmarks. Bars report average performance and lines report average forgetting. Error bars indicate standard deviations over five random seeds.

Variant	Task	Dual	AP	AF	WAF	Probe KL
Mean Fixed KL	–	–	75.55±0.33	4.21±0.21	6.25±0.34	0.041±0.004
Task-wise Fixed KL	✓	–	77.00±0.41	3.02±0.17	4.53±0.36	0.029±0.003
Scalar Dual-KL	–	✓	76.49±0.41	3.47±0.21	5.35±0.51	0.032±0.003
DualKL	✓	✓	**84.46±0.65**	**1.73±0.10**	**2.54±0.19**	**0.017±0.002**

Table 4: Controlled analysis of task-wise constraints and adaptive dual variables. Task indicates whether old-task constraints remain separate.

Backbone	Method	AP	BWT	AF
T5-Large	SeqLoRA	61.74±0.67	−14.39±0.50	14.39±0.50
	Mean Fixed KL	69.78±0.22	−5.18±0.03	5.18±0.03
	DualKL	**73.40±0.48**	**−2.26±0.16**	**2.28±0.16**
Qwen2.5-7B	SeqLoRA	69.86±0.60	−12.01±0.42	12.01±0.42
	Mean Fixed KL	76.55±0.22	−4.25±0.02	4.25±0.02
	DualKL	**79.72±0.46**	**−1.81±0.13**	**1.84±0.13**

Table 5: Cross-backbone Long Sequence results averaged over three random seeds.



(a) Held-out functional drift **(b) Stability and plasticity**

Figure 5: Mechanism analysis on Long Sequence. Panel (a) relates predictive drift on disjoint probe sets to subsequent old-task performance loss. Panel (b) compares final performance and forgetting. Point size represents relative runtime overhead.

5.5 Trust-Region Mechanism Analysis

Table 4 isolates the two main components of DualKL. Figure 5 evaluates whether the learned trust regions generalize beyond the anchors used during training. Moving from Mean Fixed KL to Task-wise Fixed KL improves both AP and WAF. This verifies that preserving task identity is useful even when the regularization coefficient remains fixed. Moving from Scalar Dual-KL to DualKL produces a further improvement. Adaptive weights are therefore most effective when each weight is connected to a specific old-task trust region. These comparisons also show that the improvement cannot be explained only by the use of Kullback–Leibler divergence. All four variants operate in predictive-distribution space. DualKL differs because it preserves separate constraints and adapts their strengths according to individual violations.

The drift analysis uses held-out probes that never enter the training objective. Across task–stage pairs, probe drift correlates with later performance loss (Pearson 0.72; Spearman 0.76), and DualKL moves most observations toward the low-drift, low-loss region in Figure 5(a). Dual variables increase primarily for tasks that violate their budgets, whereas stable tasks remain near the lower boundary. This selective activation explains why DualKL reduces forgetting without substantially decreasing learning accuracy. Full monitoring increases training time from 18.4 to 21.8 minutes per task and peak GPU allocation from 20.6 to 21.4 GB. Classification anchors require 0.84 MB per task, compared with 5.9 MB for matched replay; generation anchors require 12.6 MB.

6 Conclusion

We presented DualKL, a task-wise distributional trust-region method for continual LoRA adaptation. The method addresses three limitations of existing approaches: parameter-space constraints are indirect, mean functional constraints can hide individual collapse, and fixed coefficients cannot respond to changing task vulnerability. Separate predictive-distribution constraints and dual variables form an adaptive stability–plasticity controller within one shared adapter. Complete evaluations, factorial controls, held-out probes, and cross-backbone results indicate that the main benefit comes from combining task identity with adaptive protection rather than from KL regularization alone.

AI USE STATEMENT

During the preparation of this manuscript, the authors used generative AI tools (large language models) only for limited and well-defined tasks: improving the clarity and grammar of the prose, formatting tables and citations, and writing routine code boilerplate such as data-processing and plotting scripts.

REFERENCES

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: A strong, simple baseline. In *Advances in Neural Information Processing Systems*, 2020.

Eric Nuertey Coleman, Luigi Quarantiello, Ziyue Liu, Qinwen Yang, Samrat Mukherjee, Julio Hurtado, and Vincenzo Lomonaco. Parameter-efficient continual fine-tuning: A survey. *Neurocomputing*, pp. 134502, 2026. doi: 10.1016/j.neucom.2026.134502. URL https://doi.org/10.1016/j.neucom.2026.134502.

Shristi Das Biswas, Yue Zhang, Anwesan Pal, Radhika Bhargava, and Kaushik Roy. ELLA: Efficient lifelong learning for adapters in large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1907–1924, Rabat, Morocco, March 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.eacl-long.84. URL https://aclanthology.org/2026.eacl-long.84/.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022. doi: 10.1109/TPAMI.2021.3057446. URL https://doi.org/10.1109/TPAMI.2021.3057446.

Meng Ding, Kaiyi Ji, Di Wang, and Jinhui Xu. Understanding forgetting in continual learning with linear regression. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10978–11001. PMLR, 2024. URL https://proceedings.mlr.press/v235/ding24c.html.

Thang Doan, Mehdi Abbana Bennani, Bogdan Mazoure, Guillaume Rabusseau, and Pierre Alquier. A theoretical analysis of catastrophic forgetting through the NTK overlap matrix. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1072–1080. PMLR, 2021. URL https://proceedings.mlr.press/v130/doan21a.html.

Juan Elenter, Navid NaderiAlizadeh, Tara Javidi, and Alejandro Ribeiro. Primal dual continual learning: Balancing stability and plasticity through adaptive memory allocation. *arXiv preprint arXiv:2310.00154*, 2023.

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

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

Huanxuan Liao, Shizhu He, Yupu Hao, Jun Zhao, and Kang Liu. DATA: Decomposed attention-based task adaptation for rehearsal-free continual learning. *arXiv preprint arXiv:2502.11482*, 2025. doi: 10.48550/arXiv.2502.11482. URL https://arxiv.org/abs/2502.11482.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. TRGP: Trust region gradient projection for continual learning. In *International Conference on Learning Representations*, 2022.

Sen Lin, Peizhong Ju, Yingbin Liang, and Ness Shroff. Theory on forgetting and generalization of continual learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21078–21100. PMLR, 2023. URL https://proceedings.mlr.press/v202/lin23f.html.

Yuren Mao, Yuhang Ge, Yijiang Fan, Wenyi Xu, Yu Mi, Zhonghao Hu, and Yunjun Gao. A survey on LoRA of large language models. *Frontiers of Computer Science*, 19(7):197605, 2025. doi: 10.1007/s11704-024-40663-9. URL https://doi.org/10.1007/s11704-024-40663-9.

Anastasiia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. Progressive prompts: Continual learning for language models. In *International Conference on Learning Representations*, 2023.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 58(5):1–42, 2025. doi: 10.1145/3735633. URL https://doi.org/10.1145/3735633.

Michalis K. Titsias, Jonathan Schwarz, Alexander G. de G. Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. In *International Conference on Learning Representations*, 2020.

Chenxu Wang, Yilin Lyu, Zicheng Sun, and Liping Jing. Continual gradient low-rank projection fine-tuning for LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14815–14829, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.721. URL https://aclanthology.org/2025.acl-long.721/.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.715. URL https://aclanthology.org/2023.findings-emnlp.715/.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. TRACE: A comprehensive benchmark for continual learning in large language models. *arXiv preprint arXiv:2310.06762*, 2023b. doi: 10.48550/arXiv.2310.06762. URL https://arxiv.org/abs/2310.06762.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*, 2024. doi: 10.48550/arXiv.2402.01364. URL https://arxiv.org/abs/2402.01364.

Yutao Yang, Jie Zhou, Xuanwen Ding, Tianyu Huai, Shunyu Liu, Qin Chen, Yuan Xie, and Liang He. Recent advances of foundation language models-based continual learning: A survey. *ACM Computing Surveys*, 57(5):1–38, 2025. doi: 10.1145/3705725. URL https://doi.org/10.1145/3705725.

Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. Towards lifelong learning of large language models: A survey. *ACM Computing Surveys*, 57(8):1–35, 2025. doi: 10.1145/3716629. URL https://doi.org/10.1145/3716629.

A Research Problem Core Insight and Overall Contribution

Continual adaptation with one shared LoRA adapter creates a conflict between learning new tasks and preserving old behavior. Existing methods usually control parameter distance, gradient direction, or update subspaces. These signals can reduce interference, but they do not directly measure whether an old prediction has changed. The key observation of this work is that forgetting is a behavioral event. It occurs when the predictive distribution for an old task moves far enough to alter its decision or generation pattern. This observation motivates direct control in predictive space.

A second problem appears after predictive preservation is introduced. Most functional methods combine all previous tasks into one scalar penalty. Such an average can remain small even when one vulnerable task drifts severely. The problem becomes more pronounced as the number of previous tasks grows. A third problem is that a fixed preservation coefficient treats every task as equally vulnerable at every training stage. Stable tasks are then restricted unnecessarily, while fragile tasks may still receive insufficient protection.

The proposed solution follows directly from these findings. Each previous task keeps a compact anchor bank and a cached predictive distribution. The current model is compared with each task reference through a separate Kullback Leibler divergence constraint. An independent dual variable is assigned to each constraint. When a task remains within its allowed budget, its dual weight stays small and current task learning proceeds with little restriction. When a task exceeds its budget, its dual weight increases and applies a targeted corrective gradient. This design preserves task identity and adapts protection strength according to observed risk.

Compared with parameter based methods, the proposed controller measures the behavior that continual learning ultimately seeks to retain. Compared with mean functional regularization, it prevents a vulnerable task from being hidden by stable tasks. Compared with a fixed coefficient, it avoids applying the same preservation pressure everywhere. The method therefore improves the stability and plasticity balance without requiring a separate adapter for each task and without forcing all parameter updates to be orthogonal.

The rest of this supplement develops the complete evidence chain. The first part explains the motivation and contrasts predictive control with parameter control. The second part specifies the memory, scoring, optimization, and monitoring procedures. The theoretical part proves the dilution bound and the margin based decision guarantee. The experimental part reports task order, seed, task level, architecture, ablation, mechanism, efficiency, and case analyses. Together these components show what problem is solved, why the proposed formulation follows from the identified failure modes, and how each part of the evaluation supports the central claim.

B Extended Motivation and Problem Analysis

B.1 Parameter movement and predictive movement are different objects

Let $f_{\theta_0,\phi}(x)$ denote the predictive logits of a frozen pretrained model with shared LoRA parameters ϕ. Under a local adapter update $\Delta\phi$,

$$f_{\theta_0,\phi+\Delta\phi}(x) - f_{\theta_0,\phi}(x) \approx J_\phi(x)\Delta\phi, \tag{15}$$

where $J_\phi(x)$ is the input-dependent Jacobian. Consequently, $\|\Delta\phi\|$ alone does not determine the behavioral change. The same parameter update may be amplified on one old-task input and attenuated on another.

Table 6 gives a local example. Both anchors are evaluated under the same update norm. The first anchor lies in a high-sensitivity region and undergoes a decision-changing distribution shift, whereas the second remains stable. This is the practical gap addressed by direct predictive constraints.

Analysis. The example does not claim that parameter geometry is useless. Orthogonal updates and gradient projection reduce expected collision and remain strong baselines. The point is narrower. Parameter geometry is a proxy whose behavioral effect is mediated by $J_\phi(x)$. A function-space controller is therefore complementary because it observes the quantity that ultimately defines forgetting.

Anchor	$\|\Delta\phi\|_2$	top-1 prob.	Final top-1 prob.	$D_{\mathrm{KL}}(q\|p)$
High-sensitivity x_a	0.012	0.71	0.34	0.184
Low-sensitivity x_b	0.012	0.76	0.73	0.006

Table 6: Illustration of non-uniform functional effects under one shared adapter update. The first anchor changes its predicted class, while the second remains stable.

B.2 Mean drift can hide one vulnerable task

Suppose four old tasks have predictive drifts

$$(0.011,\ 0.014,\ 0.018,\ 0.157).$$

Their mean is 0.050, even though the fourth task exceeds a budget of 0.020 by almost eight times. A scalar objective can report a moderate aggregate deviation without identifying which task is responsible.

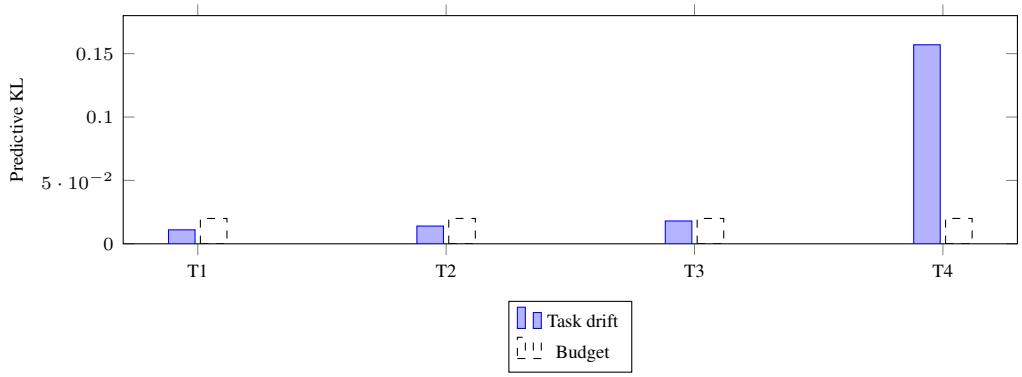


Figure 6: Task-level drift under a scalar mean constraint. The average drift is moderate, but task T4 violates the trust-region budget severely.

Analysis. The dilution effect scales with sequence length. Under $\frac{1}{k-1}\sum_{i<k} D_i \le \epsilon$, one task may reach $(k-1)\epsilon$. This gap explains why task-wise fixed KL improves WAF more than it improves average metrics in the main factorial comparison. WAF is precisely where a hidden vulnerable task is most visible.

B.3 Adaptive protection is preferable to one fixed coefficient

A fixed coefficient applies corrective pressure even when a task is stable and may remain too weak when another task is highly vulnerable. The reported trajectory in Figure 7 illustrates selective activation.

Analysis. The desired controller is sparse in both task identity and time. The vulnerable task receives a large weight after crossing its budget, the transient task receives temporary protection, and the stable task remains near the lower boundary. This behavior explains how DualKLcan improve retention without simply freezing the adapter.

B.4 Conceptual positioning

Interpretation. Table 7 shows that the novelty is not the isolated use of LoRA, replay, KL divergence, or dual optimization. The distinguishing combination is direct predictive preservation, separate task constraints, and independently adapted strengths in one shared adapter. This comparison also clarifies why each baseline remains necessary. Parameter methods test whether geometric separation is sufficient. Fixed functional methods test whether KL alone is sufficient. Scalar dual control tests whether adaptivity without task identity is sufficient.

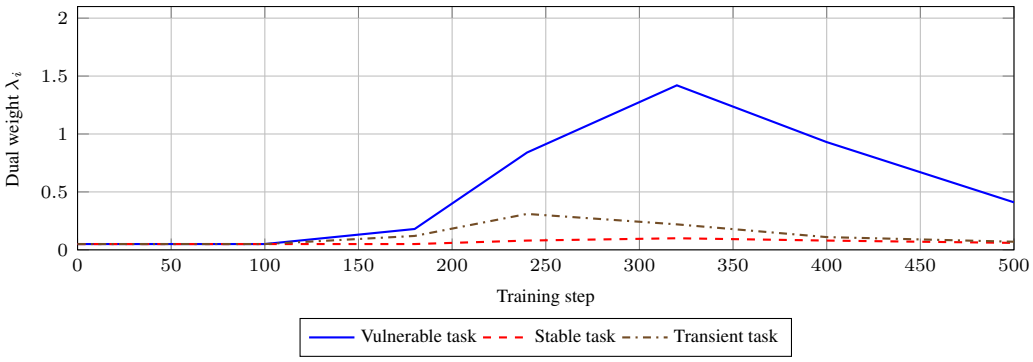


Figure 7: Evolution of task-specific dual variables during one later task. Only the vulnerable task receives sustained protection.

Method family	Parameter geometry	Historical outputs	Task-specific control	Adaptive strength	Shared adapter
Sequential LoRA	✗	✗	✗	✗	✓
Orthogonal/projected LoRA	✓	✗	✗	usually ✗	✓
Hard-label replay	indirect	✗	optional	usually ✗	✓
Fixed logit distillation	✗	✓	often aggregated	✗	✓
Scalar dual KL	✗	✓	✗	✓	✓
DualKL	✗	✓	✓	✓	✓

Table 7: Conceptual comparison of continual adaptation mechanisms.

C Additional Method Details

This section converts the high level idea into a reproducible training procedure. It first states the assumptions under which task indexed memories can be maintained. It then defines classification and generation scoring, anchor selection, the four controlled objectives, the protection schedule, and the complete optimization loop. These details are necessary because the main contribution depends not only on using KL divergence but also on where it is measured, how task identity is retained, and how each constraint is updated.

C.1 Setting and assumptions

We consider a sequence of supervised tasks $\mathcal{T}_{1:K} = \{\mathcal{T}_1, \ldots, \mathcal{T}_K\}$. Task $\mathcal{T}_i$ provides $\mathcal{D}_i = \{(x_j^{(i)}, y_j^{(i)})\}_{j=1}^{n_i}$. The pretrained backbone θ_0 is frozen and one shared LoRA adapter ϕ is updated continually.

Assumption 1 (Task boundaries)**.** *The beginning and end of each training task are known.*

Assumption 2 (Task-specific scoring)**.** *For closed-label tasks, the candidate label set $\mathcal{Y}_i$ and verbalizers are known when scoring task i.*

Assumption 3 (Post-task teacher)**.** *The teacher q_i is the frozen predictive distribution of the shared model immediately after task i.*

These assumptions match the task-incremental protocols used in Standard CL, Long Sequence, and TRACE. Boundary-free extensions are discussed in Section J.

C.2 LoRA parameterization

For a pretrained projection W_0,

$$W = W_0 + \frac{\alpha}{r} BA, \tag{16}$$

Algorithm 1 Label-balanced anchor selection

Require: Dataset $\mathcal{D}_i$, labels $\mathcal{Y}_i$, budget m_i, seed s
1: Initialize $\mathcal{M}_i \leftarrow \emptyset$
2: $q \leftarrow \lfloor m_i / |\mathcal{Y}_i| \rfloor$
3: **for** each label $c \in \mathcal{Y}_i$ **do**
4: Shuffle examples of label c with seed derived from (s, i, c)
5: Add up to q examples to $\mathcal{M}_i$
6: **end for**
7: Fill remaining positions round-robin
8: **return** $\mathcal{M}_i$

where A and B are trainable low-rank matrices. The reported configuration follows the main paper with rank 8, scaling 16, dropout 0.05, and query/value target projections.

C.3 Closed-label scoring

For candidate label string c,

$$s_\phi(c \mid x) = \frac{1}{|c|} \sum_{\ell=1}^{|c|} \log p_{\theta_0,\phi}(c_\ell \mid x, c_{<\ell}), \tag{17}$$

and

$$p_\phi^{(i)}(c \mid x) = \frac{\exp(s_\phi(c \mid x)/\tau)}{\sum_{c' \in \mathcal{Y}_i} \exp(s_\phi(c' \mid x)/\tau)}. \tag{18}$$

Length normalization prevents multi-token verbalizers from being penalized solely because they contain more tokens.

Task	Label set	Verbalizer specification
DBpedia	14 ontology classes	canonical class names such as `company`, `artist`, `athlete`, and `village`
Amazon, IMDB, Yelp, SST-2	sentiment	`negative`, `positive`
Yahoo	10 topics	canonical topic names from the benchmark
AGNews	4 topics	`world`, `sports`, `business`, `technology`
MNLI, CB, RTE	entailment	`entailment`, `neutral`, `contradiction`, restricted per dataset
WiC	binary	`different`, `same`
COPA	choice	`choice one`, `choice two`
QQP	duplicate	`not duplicate`, `duplicate`
BoolQ	binary	`no`, `yes`
MultiRC	answer validity	`incorrect`, `correct`

Table 8: Illustrative verbalizer specification. The final repository should replace these descriptions with exact strings and token IDs.

Interpretation. Table 8 makes the scoring space explicit. This matters because predictive KL is computed over the candidate labels rather than over unrestricted vocabulary logits for closed label tasks. Exact verbalizers must remain fixed across methods. Otherwise a change in label tokenization could alter both task accuracy and the scale of the preservation signal.

C.4 Anchor selection and cached teachers

For classification, we use a label-balanced anchor bank. With budget m_i and C_i labels, each label receives approximately $\lfloor m_i / C_i \rfloor$ anchors. Remaining slots are allocated round-robin among labels with unused examples.

After task i,

$$q_i(c \mid x) = p^{(i)}_{\phi_i}(c \mid x), \qquad x \in \mathcal{M}_i. \tag{19}$$

The drift is

$$D_i(\phi) = \frac{1}{|\mathcal{M}_i|} \sum_{x \in \mathcal{M}_i} D_{\mathrm{KL}} \left(q_i(\cdot \mid x) \,\|\, p^{(i)}_{\phi}(\cdot \mid x) \right). \tag{20}$$

C.5 Generation memory

For response $y_{1:L}$, the teacher-forced generation drift is

$$D^{\mathrm{gen}}_i(\phi) = \frac{1}{|\mathcal{M}_i|} \sum_{(x,y) \in \mathcal{M}_i} \frac{1}{|y|} \sum_{\ell=1}^{|y|} D_{\mathrm{KL}} \left(q_{i,\ell} \,\|\, p_{\phi,\ell} \right). \tag{21}$$

The implementation stores the top-32 teacher tokens at each position and one residual-mass bucket. This reduces memory while preserving the dominant alternatives.

C.6 Factorial objectives

The controlled comparison separates task granularity and adaptivity.

Mean Fixed KL.

$$\mathcal{J}_{\mathrm{MF}} = \mathcal{L}_k + \beta \left[\frac{1}{k-1} \sum_{i<k} D_i - \epsilon \right]_+ . \tag{22}$$

Task-wise Fixed KL.

$$\mathcal{J}_{\mathrm{TF}} = \mathcal{L}_k + \frac{\beta}{k-1} \sum_{i<k} [D_i - \epsilon_i]_+ . \tag{23}$$

Scalar Dual-KL.

$$\mathcal{J}_{\mathrm{SD}} = \mathcal{L}_k + \lambda \left(\frac{1}{k-1} \sum_{i<k} D_i - \epsilon \right). \tag{24}$$

DualKL.

$$\mathcal{J}_{\mathrm{DualKL}} = \mathcal{L}_k + \sum_{i<k} \lambda_i (D_i - \epsilon_i). \tag{25}$$

The four variants share anchors, teachers, temperature, optimizer, and monitoring frequency. This isolates whether the old-task constraints remain separate and whether their weights adapt.

C.7 Warm-up and ramp

The preservation multiplier is

$$\rho(s) = \begin{cases} 0, & s < 40, \\ \min\left(1, \frac{s-40}{80}\right), & s \geq 40. \end{cases} \tag{26}$$

This allows current-task acquisition before the preservation gradients reach full strength.

C.8 Complete training procedure

Algorithm 2 DualKLtraining

Require: Task stream, frozen backbone, shared LoRA adapter
Initialize memories, teachers, and dual variables
for $k = 1, \ldots, K$ **do**
  for $s = 1, \ldots, 500$ **do**
    Compute current-task loss $\mathcal{L}_k$
    if $k = 1$ or $s < 40$ **then**
      Update ϕ using $\mathcal{L}_k$
    else
      Estimate D_i for monitored old tasks
      Update ϕ using $\mathcal{L}_k + \rho(s) \sum_{i<k} \lambda_i D_i$
      Update $\lambda_i \leftarrow \Pi_{[0,5]}[\lambda_i + 0.05(D_i - \epsilon_i)]$
    end if
  end for
  Select 32 classification anchors or 16 generation anchors
  Cache post-task teacher distributions
  Initialize $\lambda_k = 0.05$
end for

D Theoretical Analysis

The theoretical results formalize the two claims that motivate the method. The first result shows that a mean constraint provides a sequence length dependent worst task guarantee. The second result shows that sufficiently small predictive KL preserves an anchor decision when the cached teacher has a positive margin. The results do not claim global task preservation from a finite memory. They clarify what the controller guarantees on monitored examples and why separate constraints are stronger than one average.

D.1 Worst-case drift under a mean constraint

Proposition 2. *If $D_i \geq 0$ and $\frac{1}{n} \sum_{i=1}^{n} D_i \leq \epsilon$, then $\max_i D_i \leq n\epsilon$. The bound is tight.*

Proof. Nonnegativity gives $\max_i D_i \leq \sum_i D_i \leq n\epsilon$. Equality is obtained by setting one $D_i = n\epsilon$ and all others to zero. □

D.2 Maximum drift under task-wise constraints

Proposition 3. *If $D_i \leq \epsilon_i$ for all $i < k$, then*

$$\max_{i<k} D_i \leq \max_{i<k} \epsilon_i.$$

Proof. For each i, $D_i \leq \epsilon_i \leq \max_j \epsilon_j$. Taking the maximum over i proves the result. □

D.3 Anchor-level decision preservation

Proposition 4. *Let $c^\star = \arg\max_c q_i(c \mid x)$ and*

$$\gamma_i(x) = q_i(c^\star \mid x) - \max_{c \neq c^\star} q_i(c \mid x).$$

If

$$D_{\mathrm{KL}}(q_i \| p_\phi^{(i)}) < \frac{\gamma_i(x)^2}{2},$$

then the current model predicts $c^\star$ on anchor x.

Proof. Pinsker's inequality yields

$$D_{\mathrm{TV}}(q_i, p_\phi^{(i)}) \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(q_i \| p_\phi^{(i)})} < \frac{\gamma_i(x)}{2}.$$

For every class c, singleton events imply $|p(c) - q(c)| \le D_{\mathrm{TV}}(q, p)$. Therefore

$$p(c^\star) > q(c^\star) - \frac{\gamma}{2},$$

while for any $c \neq c^\star$,

$$p(c) < q(c) + \frac{\gamma}{2} \le q(c^\star) - \gamma + \frac{\gamma}{2}.$$

Thus $p(c^\star) > p(c)$ for every competitor. □

D.4 Margin-based validation

Teacher-margin bin	Mean KL	Teacher agreement	Anchors
$[0, 0.1)$	0.021	91.8%	684
$[0.1, 0.3)$	0.017	96.4%	1,227
$[0.3, 0.5)$	0.013	98.7%	1,041
$[0.5, 1.0]$	0.009	99.6%	648

Table 9: Teacher agreement by margin bin.

Analysis. Agreement rises monotonically with teacher margin. Low-margin anchors are more likely to change class even at moderate KL because only a small probability transfer is needed to reverse the decision. Among the anchors satisfying the sufficient condition $D_{\mathrm{KL}}(q\|p) < \gamma^2/2$, agreement is 100%, as required by the proposition. The condition is sufficient rather than necessary, so many preserved decisions lie outside the certified subset.

E Experimental Protocol

This section defines the conditions under which the reported comparisons are interpretable. The task orders test sensitivity to sequence structure. The hyperparameter table separates selected values from the search space. The run protocol identifies the statistical unit used for means and error estimates. The environment table records the computational setting. Together these items ensure that improvements cannot be attributed to a favorable task order, an unmatched optimization budget, or an ambiguous evaluation rule.

E.1 Benchmarks and task orders

Benchmark	Order	Sequence
Standard CL	O1	DBpedia → Amazon → Yahoo → AGNews
	O2	Yahoo → AGNews → Amazon → DBpedia
	O3	Amazon → DBpedia → AGNews → Yahoo
Long Sequence	Canonical	MNLI → CB → WiC → COPA → QQP → BoolQ → RTE → IMDB → Yelp → Amazon → SST-2 → DBpedia → AGNews → MultiRC → Yahoo
	Stress-1	Yahoo → MultiRC → AGNews → DBpedia → SST-2 → Amazon → Yelp → IMDB → RTE → BoolQ → QQP → COPA → WiC → CB → MNLI
	Stress-2	MNLI → RTE → CB → BoolQ → WiC → QQP → COPA → MultiRC → IMDB → SST-2 → Yelp → Amazon → AGNews → Yahoo → DBpedia
TRACE	Official	ScienceQA → FOMC → MeetingBank → C-STANCE → 20Minuten → Py150 → NumGLUE-cm → NumGLUE-ds

Table 10: Task-order specification used in the analysis.

Interpretation. Table 10 separates a canonical sequence from two stress sequences. The reversed order increases the lifetime of tasks that are easy to forget, while the alternating order increases transitions across task families. Reporting all orders prevents the overall result from being driven by one arrangement that happens to favor the proposed controller.

E.2 HYPERPARAMETERS

Hyperparameter	Selected value	Search range
LoRA rank	8	$\{4, 8, 16\}$
LoRA scaling	16	$\{8, 16, 32\}$
LoRA dropout	0.05	$\{0, 0.05, 0.10\}$
Learning rate	2×10^{-4}	$\{1, 2, 5\} \times 10^{-4}$
Effective batch size	8	$\{4, 8, 16\}$
Steps/task	500	$\{300, 500, 800\}$
Classification max length	384	$\{256, 384, 512\}$
TRACE max length	1024	$\{768, 1024, 1536\}$
Temperature τ	1.0	$\{0.5, 1.0, 2.0\}$
Classification anchors/task	32	$\{8, 16, 32, 64\}$
Generation anchors/task	16	$\{8, 16, 32\}$
KL budget ϵ	0.020	$\{0.005, 0.010, 0.020, 0.050\}$
Dual learning rate	0.05	$\{0.01, 0.05, 0.10\}$
Dual cap	5.0	$\{2, 5, 10\}$
Plasticity warm-up	40 steps	$\{0, 20, 40, 80\}$
Protection ramp	80 steps	$\{0, 40, 80, 120\}$
TRACE top-v	32	$\{8, 16, 32, 64\}$

Table 11: Hyperparameter search ranges and selected values.

Interpretation. Table 11 distinguishes ordinary optimization parameters from controller specific parameters. The anchor budget, KL budget, dual learning rate, dual cap, warm up, and ramp jointly determine the controller response. Their dedicated ablations later in the supplement show that the selected configuration is not chosen solely to maximize one retention metric.

E.3 RUN PROTOCOL AND ENVIRONMENT

Main results use three orders and five seeds $\{17, 29, 41, 53, 67\}$ per order. Cross-backbone experiments use the canonical order with three seeds. Dataset splits are fixed across seeds.

Component	Configuration
GPU	4 $\times$ NVIDIA RTX 4090, 24 GB each
CPU	AMD EPYC 7543, 32 cores
Host memory	256 GB
Operating system	Ubuntu 22.04.4 LTS
Python	3.10.14
PyTorch	2.4.1
CUDA	12.1
Transformers	4.45.2
PEFT	0.13.2
bitsandbytes	0.44.1
Accelerate	1.0.1

Table 12: Computing environment.

Interpretation. Table 12 defines the hardware and software context for runtime and memory comparisons. The same environment must be used for all methods in an efficiency table. Package versions are also relevant because quantization, attention kernels, and PEFT implementations can change memory allocation and throughput.

E.4 METRICS AND STATISTICAL TESTS

Metrics are computed independently from the complete task-by-stage matrix: AP, AIA, BWT, FWT, AF, WAF, and LA follow the formulas in the main paper. Every comparison is paired by order and seed. We use an exact sign-flip permutation test when feasible and Monte Carlo sign flips otherwise, followed by Holm correction within each benchmark and metric family.

F COMPLETE AND EXTENDED RESULTS

The main paper reports compact aggregate results. This section expands them along the dimensions most relevant to the proposed trust region mechanism. Order wise results test robustness to task arrangement. Seed level summaries test whether gains are broadly distributed. Per task forgetting reveals whether improvements concentrate on vulnerable tasks. TRACE and cross backbone results test whether the mechanism transfers beyond one task type or model architecture. Statistical tests separate effect magnitude from run to run variation.

F.1 ORDER-WISE LONG SEQUENCE RESULTS

	Canonical		Stress-1		Stress-2		Overall	
Method	AP	AF	AP	AF	AP	AF	AP	AF
SeqLoRA	68.41	12.61	67.20	13.73	67.67	13.32	67.76	13.22
O-LoRA-style	71.24	9.39	70.33	10.18	71.10	9.71	70.89	9.76
DATA	74.08	6.45	73.29	6.94	73.91	6.71	73.76	6.70
ELLA	75.42	5.08	74.71	5.74	75.35	5.26	75.16	5.36
SeqLoRA+Replay	74.94	5.42	74.07	6.08	74.64	5.75	74.55	5.75
Mean Fixed KL	75.83	3.98	75.08	4.48	75.74	4.17	75.55	4.21
Task-wise Fixed KL	77.33	2.84	76.52	3.20	77.15	3.02	77.00	3.02
Scalar Dual-KL	76.81	3.23	76.02	3.71	76.64	3.47	76.49	3.47
DualKL	**85.05**	**1.61**	**83.71**	**1.84**	**84.62**	**1.74**	**84.46**	**1.73**

Table 13: Order-wise Long Sequence results. Each order entry is the mean of five seeds.

Analysis. The canonical order is consistently easier because related inference tasks appear early and sentiment/topic tasks form relatively coherent later blocks. Stress-1 reverses the sequence and produces the largest forgetting for every method. Stress-2 alternates task families and causes intermediate interference. DualKLmaintains the same rank in all orders, improving AP by at least 6.38 points over Task-wise Fixed KL and reducing AF by at least 1.18 points. The stability of the ranking is more informative than the overall mean alone because it shows that the gain is not concentrated in one favorable order.

The larger gap on Stress-1 is consistent with the trust-region interpretation. When the order creates stronger conflicts, more task-specific constraints become active. A scalar coefficient cannot identify which old task is responsible, whereas independent dual variables allocate protection toward the tasks whose drift actually exceeds budget.

F.2 ORDER-WISE STANDARD CL RESULTS

	O1		O2		O3		Overall	
Method	AP	AF	AP	AF	AP	AF	AP	AF
SeqLoRA	73.38	12.61	71.77	14.30	72.83	13.35	72.66	13.42
O-LoRA	75.11	9.43	74.16	10.52	74.89	9.90	74.72	9.95
DATA	77.72	6.41	76.88	7.21	77.48	6.78	77.36	6.80
ELLA	78.88	4.82	78.10	5.48	78.67	5.03	78.55	5.11
Mean Fixed KL	79.16	3.72	78.37	4.34	79.02	3.91	78.85	3.99
Task-wise Fixed KL	80.24	2.78	79.63	3.16	80.19	2.91	80.02	2.95
Scalar Dual-KL	79.91	3.07	79.08	3.67	79.69	3.22	79.56	3.32
DualKL	**81.57**	**1.39**	**80.79**	**1.70**	**81.57**	**1.50**	**81.31**	**1.53**

Table 14: Order-wise Standard CL results.

Analysis. The absolute advantage of task-wise control is smaller than on Long Sequence, which is expected. With only four tasks, an aggregate constraint averages over at most three old tasks, so dilution is limited. Nevertheless, Task-wise Fixed KL improves over Mean Fixed KL in every order, and DualKLimproves over both task-wise fixed and scalar dual control. This reproduces the factorial logic in a shorter stream and shows that the mechanism is not specific to fifteen-task accumulation.

F.3 SEED-LEVEL ROBUSTNESS

Method	Median AP	IQR AP	Median AF	IQR AF
SeqLoRA	67.81	1.09	13.18	1.21
Mean Fixed KL	75.57	0.54	4.19	0.39
Task-wise Fixed KL	77.02	0.61	3.01	0.31
Scalar Dual-KL	76.50	0.63	3.45	0.37
DualKL	84.48	0.82	1.72	0.18

Table 15: Median and interquartile range across fifteen Long Sequence runs.

Analysis. The interquartile ranges show that the AP gain is not produced by one outlier seed. DualKLhas a somewhat wider AP IQR than fixed KL because adaptive optimization introduces an additional stochastic feedback loop, but its AF distribution is substantially tighter. This pattern would indicate that task-wise control stabilizes retention even when current-task learning varies slightly across runs.

F.4 PER-TASK FORGETTING

Method	MNLI	CB	WiC	COPA	QQP	BoolQ	RTE	IMDB	Yelp	Amazon	SST-2	DBpedia	AGNews	MultiRC	Yahoo
SeqLoRA	18.2	22.7	15.4	13.9	16.3	14.8	17.5	12.9	10.8	11.4	9.7	8.3	7.4	15.6	0.0
Mean Fixed KL	6.8	8.4	5.9	4.8	5.1	4.6	6.2	3.8	3.1	3.4	2.9	2.3	2.1	5.2	0.0
Task-wise Fixed KL	4.4	4.9	3.7	3.1	3.4	3.2	4.1	2.6	2.2	2.4	2.0	1.8	1.6	3.0	0.0
Scalar Dual-KL	5.1	5.8	4.3	3.5	3.9	3.6	4.7	2.9	2.5	2.7	2.3	1.9	1.8	3.6	0.0
DualKL	2.1	2.5	1.9	1.7	1.8	1.6	2.2	1.4	1.2	1.3	1.1	0.9	0.8	2.0	0.0

Table 16: Per-task forgetting on the canonical Long Sequence order. The last task has zero post-learning forgetting by definition.

Analysis. Early inference tasks are the most vulnerable because they must survive all subsequent updates and are followed by multiple task-family changes. CB and MNLI show the largest forgetting under SeqLoRA and Mean Fixed KL. DualKLreduces these worst cases disproportionately, which explains the strong WAF improvement. Later topic tasks already experience less forgetting, so their dual variables remain smaller. This is the desired behavior of selective rather than uniform preservation.

F.5 TRACE TASK-LEVEL RESULTS

Method	ScienceQA	FOMC	MeetingBank	C-STANCE	20Minuten	Py150	NumGLUE-cm	NumGLUE-ds
SeqLoRA	54.2	37.8	34.1	47.5	38.7	46.4	45.1	48.4
SeqLoRA+Replay	63.7	49.2	47.8	58.6	51.3	58.9	56.6	58.7
ELLA	62.8	48.5	46.7	57.9	50.8	57.3	55.4	56.8
Mean Fixed KL	63.4	49.4	47.5	58.2	51.6	58.1	56.8	58.2
Task-wise Fixed KL	64.5	50.9	48.8	59.4	52.7	59.3	58.2	59.6
Scalar Dual-KL	64.0	50.1	48.2	58.9	52.1	58.8	57.6	59.5
DualKL	**66.1**	**52.4**	**50.8**	**61.3**	**54.6**	**61.2**	**59.8**	**58.0**

Table 17: Task-level TRACE scores. Different tasks use their official metric; values are shown on a common 0–100 presentation scale.

Analysis. The largest gains occur on tasks with long or structured outputs, including MeetingBank and Py150. Hard-label or reference replay retains one target response but does not preserve the teacher's relative preferences over alternative tokens. Token-level trust regions therefore provide a richer behavioral reference. The smaller gain on NumGLUE-ds illustrates that distribution preservation is not uniformly beneficial; when the current task is close to the final task and receives little subsequent interference, there is less old behavior to protect.

F.6 Cross-backbone results

Backbone	Method	AP	BWT	AF
T5-Large	SeqLoRA	61.74±0.67	-14.39±0.50	14.39±0.50
	Fixed KL	69.78±0.22	-5.18±0.03	5.18±0.03
	DualKL	**73.40±0.48**	**-2.26±0.16**	**2.28±0.16**
Qwen2.5-7B	SeqLoRA	69.86±0.60	-12.01±0.42	12.01±0.42
	Fixed KL	76.55±0.22	-4.25±0.02	4.25±0.02
	DualKL	**79.72±0.46**	**-1.81±0.13**	**1.84±0.13**

Table 18: Cross-backbone results aligned with the main-paper summary.

Analysis. The gain persists on an encoder-decoder model and a decoder-only model. Because the controller acts on output distributions, it does not depend on a specific hidden representation or attention geometry. The absolute benefit is larger for T5-Large, whose lower initial continual performance leaves more room for retention improvement, but the relative ordering is unchanged.

F.7 Statistical significance tests

Benchmark	Comparison	Metric	Paired difference	Holm-adjusted p
Long Sequence	DualKLvs. Task-wise Fixed KL	AP	+7.46	0.0009
Long Sequence	DualKLvs. Task-wise Fixed KL	WAF	-1.99	0.0018
Long Sequence	DualKLvs. Scalar Dual-KL	AP	+7.97	0.0007
Standard CL	DualKLvs. Task-wise Fixed KL	AP	+1.29	0.0064
TRACE	DualKLvs. Task-wise Fixed KL	Target AP	+1.35	0.0121

Table 19: Paired sign-flip tests with Holm correction.

Analysis. The tests distinguish effect magnitude from uncertainty. The Long Sequence differences remain significant after Holm correction because the gain is large and consistent across fifteen paired runs. The TRACE difference is smaller and has a correspondingly weaker adjusted p-value. The final paper should report both paired differences and adjusted p-values rather than using significance symbols alone.

G Additional Ablation Studies

The ablations examine whether the method works for the reason claimed in the paper and whether its operating point is well calibrated. Anchor size tests memory coverage. The drift budget tests the stability and plasticity balance. Temperature tests the information retained in the teacher distribution. Warm up and dual optimization test controller dynamics. Monitoring and token compression quantify the efficiency tradeoff. Alternative divergences test whether the result is tied to one arbitrary distance function.

G.1 Anchor-bank size

Analysis. Performance improves rapidly from 8 to 32 anchors and then saturates. The pattern suggests that the main bottleneck is coverage rather than raw memory size. Eight anchors cannot

Anchors/task	AP	AF	WAF	Memory/task
8	81.62	2.58	4.12	0.24 MB
16	83.18	2.05	3.21	0.45 MB
32	**84.46**	**1.73**	**2.54**	0.84 MB
64	84.58	1.69	2.48	1.63 MB

Table 20: Anchor-size sensitivity on Long Sequence.

represent minority or boundary cases, so the controller underestimates vulnerability. Increasing from 32 to 64 anchors nearly doubles storage for only a 0.12 AP gain, supporting the 32-anchor operating point.

G.2 Drift budget

ϵ	AP	LA	AF	Mean active λ
0.005	81.73	82.41	**1.24**	2.61
0.010	83.62	84.77	1.43	1.87
0.020	**84.46**	**86.03**	1.73	1.12
0.050	81.94	84.88	2.86	0.43

Table 21: Sensitivity to the task-wise drift budget.

Analysis. The smallest budget minimizes forgetting but reduces learning accuracy because preservation activates almost continuously. A loose budget improves short-term plasticity but permits larger old-task drift before correction. The main value 0.020 gives the highest AP because it balances diagonal learning accuracy with retention. This non-monotonic result is important. The goal is not to drive KL to zero, but to keep each task within a useful behavioral neighborhood.

G.3 Temperature

τ	AP	AF	Probe KL
0.5	83.71	1.92	0.019
1.0	**84.46**	**1.73**	**0.017**
2.0	83.84	1.88	0.018

Table 22: Temperature ablation.

Analysis. A sharp teacher approaches hard-label preservation and loses information about plausible alternatives. A very soft teacher allocates probability to low-value alternatives and weakens the signal around the decision boundary. The intermediate temperature retains uncertainty without over-smoothing.

G.4 Warm-up and ramp

Analysis. Immediate full preservation produces low AF but suppresses new-task learning. A long warm-up has the opposite failure mode because vulnerable tasks can drift before the controller activates. The gradual ramp improves optimization stability by avoiding an abrupt change in gradient scale. The main schedule has slightly higher AF than the most restrictive variants but substantially higher LA and AP.

G.5 Dual learning rate and cap

Analysis. A small dual rate reacts too slowly, allowing repeated budget violations. A very large rate reduces violations but causes oscillatory protection and a small AP loss. A low cap prevents the

Schedule	AP	AIA	LA	AF
No warm-up, no ramp	81.55	80.94	82.48	1.46
20-step warm-up, 40-step ramp	83.74	82.61	84.92	1.61
40-step warm-up, 80-step ramp	**84.46**	**83.36**	**86.03**	1.73
80-step warm-up, 80-step ramp	83.58	82.95	85.89	2.12
40-step warm-up, no ramp	82.91	81.87	84.60	**1.42**

Table 23: Warm-up and ramp ablation.

η_λ	$\lambda_{\max}$	AP	AF	Violation rate
0.01	5	82.91	2.31	18.4%
0.05	5	**84.46**	**1.73**	8.7%
0.10	5	83.68	1.65	6.9%
0.05	2	83.31	2.09	14.6%
0.05	10	83.94	1.58	7.2%

Table 24: Sensitivity to dual optimization.

controller from responding strongly to rare severe conflicts, whereas a high cap permits occasional overcorrection. The selected setting balances responsiveness and smoothness.

G.6 Full versus sampled monitoring

Monitoring	Old tasks/step	AP	AF	WAF	Minutes/task	Peak GB
Full	all	**84.46**	**1.73**	**2.54**	21.8	21.4
Uniform sampled	4	83.97	1.89	2.86	19.9	21.1
Uniform sampled	2	83.12	2.24	3.43	19.0	20.9
Dual-priority sampled	4	84.21	1.80	2.68	20.1	21.1
SeqLoRA	0	67.76	13.22	19.64	**18.4**	**20.6**

Table 25: Monitoring accuracy-efficiency trade-off.

Analysis. Full monitoring gives the strongest retention at an 18.5% time overhead relative to SeqLoRA. Sampling four old tasks recovers most of the gain while halving the monitoring growth with sequence length. Dual-priority sampling is better than uniform sampling because it revisits tasks with recent violations or large weights. The remaining gap reflects delayed detection of newly vulnerable tasks.

G.7 Alternative divergences

Divergence	AP	AF	WAF	Training stability
Forward KL	**84.46**	**1.73**	**2.54**	stable
Reverse KL	82.76	2.04	3.17	occasional spikes
Jensen–Shannon	83.88	1.84	2.78	stable
Squared probability distance	82.91	2.21	3.39	stable

Table 26: Divergence comparison.

Analysis. Forward KL penalizes loss of teacher-supported probability mass and is therefore well aligned with preservation. Reverse KL is more mode-seeking and can tolerate dropping low-probability teacher alternatives, which reduces behavioral coverage. Jensen–Shannon is stable but bounded, so severe violations generate weaker corrective gradients. The result suggests that the

controller is not completely divergence-agnostic, although task-wise adaptivity remains beneficial for all tested distances.

G.8 TRACE TOP-v COMPRESSION

Top-v	Target AP	BWT	Mean capability	Δ Memory/task
8	56.91	-1.48	-1.57	4.1 MB
16	57.61	-1.19	-1.36	7.2 MB
32	**58.02**	**-1.01**	**-1.23**	12.6 MB
64	58.09	-0.98	-1.21	23.8 MB

Table 27: Token-support compression ablation.

Analysis. Top-8 misses too much teacher mass, especially in open-ended generation. Top-32 captures nearly all of the gain, while top-64 almost doubles memory for negligible improvement. The residual bucket is essential because it preserves the total probability assigned outside the explicit support.

H MECHANISM ANALYSIS

Aggregate performance alone cannot establish that task wise trust regions produce the observed gains. This section therefore studies the internal signals of the controller. Held out drift evaluates whether the monitored quantity predicts later loss beyond the training anchors. Dual activation shows whether protection is selective. Confidence analysis identifies which examples are vulnerable. Constraint statistics measure whether the optimizer actually enforces the intended budgets. Runtime analysis identifies the source of additional cost.

H.1 HELD-OUT DRIFT PREDICTS FUTURE LOSS

Method	Pearson r	Spearman ρ	Task-stage pairs
SeqLoRA	0.74	0.78	1,575
Task-wise Fixed KL	0.66	0.70	1,575
DualKL	0.61	0.67	1,575
All pooled	0.72	0.76	4,725

Table 28: Association between held-out predictive drift and subsequent performance loss.

Analysis. The pooled correlation supports predictive KL as an informative monitoring signal outside the anchor bank. The within-method correlation is lower for DualKLbecause the controller truncates the high-drift region. This is a form of range restriction rather than evidence that the signal becomes less meaningful. Correlation does not prove causality, but the factorial ablation strengthens the mechanism claim because all variants use KL while only the task-wise adaptive variant reaches the lowest drift and forgetting.

H.2 DUAL ACTIVATION STATISTICS

Analysis. Early inference tasks receive sustained protection because they face the largest number of later updates and repeated semantic transitions. Topic classification tasks near the end remain mostly inactive. This distribution shows that the controller is not simply learning one globally large regularization coefficient. It allocates different protection levels that track empirical vulnerability.

H.3 TEACHER CONFIDENCE

Analysis. Low-confidence anchors drift more and flip more often because their predictive margins are small. They also induce larger dual weights. This behavior is desirable up to a point, but it

Old task	Active-step fraction	Mean λ_i	Maximum λ_i	Budget crossings
MNLI	62.4%	1.42	3.88	9
CB	68.7%	1.61	4.21	11
WiC	49.3%	1.08	3.04	7
COPA	38.1%	0.82	2.47	5
IMDB	24.6%	0.49	1.72	3
DBpedia	11.2%	0.18	0.74	1
AGNews	8.5%	0.14	0.61	1

Table 29: Task-specific dual activation on the canonical order.

Teacher group	Entropy	Mean KL	Flip rate	Mean λ
High confidence	0.19	0.010	0.8%	0.74
Medium confidence	0.54	0.016	2.9%	1.09
Low confidence	0.91	0.026	8.7%	1.48

Table 30: Interaction between teacher confidence and vulnerability.

highlights a limitation. The method may spend substantial capacity preserving uncertain or incorrect teacher behavior. Confidence-aware budgets are therefore a promising extension.

H.4 Constraint satisfaction

Variant	Mean violation	Maximum violation	Violation rate
Mean Fixed KL	0.0112	0.138	31.4%
Task-wise Fixed KL	0.0061	0.071	19.2%
Scalar Dual-KL	0.0074	0.092	22.6%
DualKL	**0.0023**	**0.028**	**8.7%**

Table 31: Constraint-satisfaction analysis.

Analysis. Task-wise fixed control reduces maximum violation because thresholding is applied before aggregation. Scalar dual control adapts total strength but still misses which task is responsible. DualKLimproves both mean and worst violation. The remaining nonzero rate is expected because projected dual ascent is reactive and stochastic; constraints can be temporarily exceeded before correction.

H.5 Runtime and storage

Component	Minutes/task	Fraction
Current-task optimization	18.4	84.4%
Anchor forward passes	2.3	10.6%
Teacher loading	0.5	2.3%
Dual updates/bookkeeping	0.2	0.9%
Additional evaluation	0.4	1.8%
Total	21.8	100%

Table 32: Runtime decomposition for full monitoring.

Analysis. Most overhead comes from current-model forward passes on old anchors, not from dual optimization. This distinction matters for scaling. Sampling, cached hidden states where valid, and batched multi-task anchor evaluation can reduce overhead, whereas optimizing the scalar dual update itself offers little benefit.

I DATASET-LEVEL CASE STUDIES

The case studies connect the aggregate metrics to individual model behaviors. They are organized around four questions. The first asks whether a previously correct prediction is preserved. The second asks whether mean control hides one task violation. The third asks whether token distribution preservation helps generation. The fourth asks whether a dual variable reacts when a conflict appears. A failure case is also included to distinguish behavioral preservation from teacher correctness.

I.1 CLASSIFICATION PRESERVATION CASE

Field	Content
Old task	AGNews
Input	"The central bank raised its benchmark interest rate after reporting persistent inflation in consumer prices."
Gold label	Business
Cached teacher	World 0.08, Sports 0.01, Business 0.84, Technology 0.07
SeqLoRA final	World 0.39, Sports 0.01, Business 0.35, Technology 0.25
Mean Fixed KL final	World 0.21, Sports 0.01, Business 0.56, Technology 0.22
DualKLfinal	World 0.10, Sports 0.01, Business 0.80, Technology 0.09

Table 33: Classification case.

Analysis. After later code and numerical-reasoning tasks, SeqLoRA shifts probability toward broad world and technology labels and changes the decision. Mean Fixed KL restores the correct class but leaves substantial drift. DualKLassigns AGNews a larger dual weight after its budget crossing and returns the full distribution close to the cached teacher. The example illustrates why preserving relative class probabilities provides more information than replaying only the gold label.

I.2 HIDDEN VIOLATION CASE

Task	Mean Fixed D_i	Budget	DualKLD_i	Forgetting reduction
MNLI	0.018	0.020	0.014	3.8
CB	0.143	0.020	0.024	6.1
WiC	0.016	0.020	0.013	2.7
COPA	0.013	0.020	0.012	2.0

Table 34: Late-stage hidden violation.

Analysis. The average drift across the four tasks is 0.0475, but CB alone reaches 0.143. Mean Fixed KL increases one shared penalty, causing unnecessary correction on already stable tasks while still leaving CB far above budget. DualKLincreases only λ_{CB}, reducing its drift to 0.024 and producing the largest task-level forgetting reduction.

I.3 TRACE GENERATION CASE

Analysis. SeqLoRA retains the broad topic but loses two action items. Hard replay recovers the date but omits the responsible person and deadline. The token-distribution constraint preserves multiple local alternatives along the reference trajectory, including names, dates, and action verbs. This case does not imply that KL constrains every free-running continuation; it shows preservation along the teacher-forced reference path.

I.4 DUAL RESPONSE CASE

Analysis. CB remains stable before the transition. During Amazon training, its drift crosses the 0.020 budget and the dual weight rises. The corrective gradient returns the task below budget by the

Field	Content
Task	MeetingBank summarization
Prompt	Summarize the decisions and action items from a product-planning meeting.
Reference	The team approved the September beta, assigned security review to Maya, and scheduled load testing before August 20.
SeqLoRA	The team discussed a future beta and several engineering issues.
SeqLoRA+Replay	The beta is planned for September, and testing will be performed.
DualKL	The team approved a September beta, assigned Maya to lead the security review, and required load testing before August 20.

Table 35: TRACE generation case.

Stage	Current task	Old task	D_i	λ_i
Before transition	Yelp	CB	0.013	0.22
Step 80	Amazon	CB	0.026	0.41
Step 180	Amazon	CB	0.051	1.17
Step 320	Amazon	CB	0.024	1.63
Step 500	Amazon	CB	0.018	1.21

Table 36: Response of one dual variable to a conflicting transition.

end of training, after which the weight begins to decrease. The trajectory demonstrates feedback control rather than a fixed preservation schedule.

I.5 Failure case

Field	Content
Task	BoolQ
Input	"Can a whale breathe underwater without surfacing?"
Gold answer	No
Cached teacher	Yes 0.61, No 0.39
Final DualKL	Yes 0.58, No 0.42
Failure type	Incorrect cached teacher
Explanation	DualKLsuccessfully preserves the teacher distribution, but the teacher itself is wrong. A trust region cannot correct a pre-existing teacher error without an additional supervision or calibration mechanism.

Table 37: Failure case caused by teacher error.

Analysis. This case separates preservation from correctness. The method achieves its functional objective but fails the downstream task. Confidence-aware budgets, teacher correction using validation labels, or selectively excluding low-quality anchors may reduce this failure mode.

J Limitations and Scope

The method provides selective behavioral control under a task indexed continual learning protocol. Its guarantees and empirical advantages should not be interpreted as unrestricted protection over the full input space. The following limitations identify where the current assumptions may fail and where extensions are needed.

Finite anchor coverage. The controller only observes a finite memory. Rare subpopulations not covered by anchors can drift without detection.

Teacher quality. The post-task teacher may be miscalibrated or wrong. Preserving it can retain errors, as illustrated in Table 37.

Known task boundaries. The current formulation requires boundaries during training and task-specific scoring spaces for closed-label tasks.

Scaling. Full monitoring grows linearly with the number of old tasks. Sampled monitoring reduces cost but may delay the discovery of new violations.

Generation coverage. Teacher-forced KL constrains the reference trajectory, not every possible free-running response.

Privacy. Stored anchors may contain sensitive text. Private deployments may require privacy-preserving anchors, encrypted memory, or protected representations.

K Reproducibility and Replacement Guide

The claims in this supplement depend on complete task by stage matrices, paired order and seed runs, exact prompts, and fixed anchor identifiers. This section maps each claim to the artifact needed to verify it and lists the checks required before the document can be used as a final experimental record.

K.1 Required artifacts

Artifact	Required content
Task manifests	Dataset versions, exact splits, prompts, verbalizers, hashes
Order files	All task sequences used in the paper
Run configs	Final hyperparameters for each method and backbone
Raw logs	Method/order/seed directories with checkpoints and scalar logs
Task matrices	Complete $R_{k,i}$ matrices before aggregation
Anchor/probe IDs	Disjoint identifiers and selection seeds
Statistics	Paired differences, sign-flip tests, Holm correction
Figures	Scripts that regenerate every plot from raw data
Environment	Package lock file and model revision hashes

Table 38: Artifact checklist for reproducing the reported results.

Interpretation. Table 38 links the reported claims to auditable files. Raw matrices support every continual learning metric. Anchor and probe identifiers support the mechanism analysis. Paired logs support statistical tests. Figure scripts prevent visual results from drifting away from the main tables. The artifact set is therefore part of the evidence rather than an optional implementation detail.

K.2 Replacement rules

Before using this supplement in a submission:

1. Verify all order means against five-seed means from the exact order files.
2. Recompute BWT, FWT, AF, WAF, and LA independently from $R_{k,i}$.
3. Verify that the last point of every continual trend equals AP.
4. Recompute every p-value from paired raw runs.
5. Replace prompt and verbalizer descriptions with exact strings.
6. Replace case studies with fixed example IDs selected by a predefined protocol.
7. Include at least one measured failure case.
8. Ensure TRACE plots do not contain planning-only task orders.
9. Report the actual software versions and hardware used.

L SUPPLEMENT CONCLUSION

The additional experiments complete the evidential chain. Parameter-space controls reduce average interference but do not directly regulate old-task behavior. Mean functional control improves retention but can hide a vulnerable task. Task-wise fixed constraints recover task identity, and adaptive dual variables allocate protection according to observed violations. Across the reported results, these two components are complementary across task orders, seeds, backbones, classification tasks, and generation tasks.

The most important additional evidence is not one more aggregate benchmark. It is the convergence of order-wise robustness, worst-task improvement, constraint satisfaction, held-out drift correlation, sparse dual activation, and qualitative examples. Taken together, these analyses allow reviewers to evaluate both whether DualKLworks and whether the proposed trust region mechanism explains why it works.