CATL: CONFLICT-AWARE TEST-TIME LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

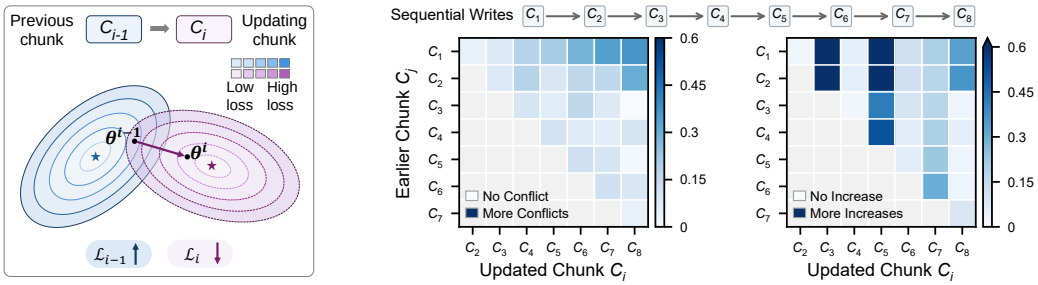
This paper studies how to rapidly acquire useful information from test-time contexts. Existing test-time learning (TTL) methods either rely on dedicated offline pretraining, or adapt lightweight parameters online without explicitly controlling the interactions among successive updates, making information encoded from earlier context portions vulnerable to interference and overwriting in long contexts. To overcome this challenge, we propose *Conflict-Aware Test-Time Learning* (CATL), a plug-and-play TTL method that casts sequential context adaptation as an online continual-learning process and introduces a conflict-aware update mechanism guided by a compact historical write memory. The memory is constructed from the optimizer update, capturing the subspace, orientation, and importance of previous writes. Guided by the memory, CATL selectively attenuates only update components that conflict with important historical writes, while preserving compatible and orthogonal directions. It thus can mitigate destructive cross-chunk interference without replaying prior context or requiring additional offline training. Experiments on long-context retrieval and reasoning tasks show that CATL outperforms all evaluated baselines at the maximum supported context length across Qwen3 base models ranging from 1.7B to 8B parameters. CATL also improves performance on instruction-tuned backbones with different attention architectures while maintaining favorable inference-time efficiency.

1 INTRODUCTION

Test-time learning (TTL) has recently emerged as a promising paradigm for adapting LLMs at inference time, where model parameters can be updated using self-supervised signals derived from the test-time input (Sun et al., 2020; Hardt & Sun, 2024). By relaxing the conventional train-then-deploy paradigm, TTL enables test-time knowledge to be encoded into the model, providing a complementary parametric mechanism for exploiting and retaining instance-specific information (Hardt & Sun, 2024; Wang et al., 2024; Ouyang et al., 2026). Existing approaches broadly pursue two directions: one redesigns the model architecture and training procedure to support test-time adaptation, using learned fast-weight or parametric-memory mechanisms to compress sequence history and reduce long-range attention costs (Sun et al., 2025; Behrouz et al., 2025; Tandon et al., 2025); the other retains pretrained model architectures and adapts only a restricted subset of its parameters on the test-time context (Hardt & Sun, 2024; Wang et al., 2024; Bansal et al., 2026; Feng et al., 2026; Ouyang et al., 2026; Zhu et al., 2026). We focus on the latter as it can be applied directly to existing pretrained models without requiring dedicated architectural redesign or offline training.

A fundamental problem in TTL is *"how can the model rapidly acquire useful knowledge from the test-time context with only a few parameter updates?"* This is non-trivial because test-time adaptation must extract useful learning signals from each context portion under tight computation and memory budgets, leaving little room for extensive optimization (Wang et al., 2026; Wakayama, 2026). Although existing methods simplify the optimization by restricting adaptation to lightweight parameter subsets or fast weights (Wang et al., 2024; Hardt & Sun, 2024; Bansal et al., 2026; Feng et al., 2026; Ouyang et al., 2026), they primarily reduce the number of trainable parameters without explicitly controlling the interaction among successive updates, making adaptation sensitive to optimization configurations (Wang et al., 2026; Wakayama, 2026). This is particularly consequential in long-context settings, where successive context portions repeatedly update the same shared parameters, and *later updates can conflict with and overwrite information encoded by earlier ones* (Yu et al., 2020; Saha & Roy, 2023; Wang et al., 2024; Chen et al., 2026), as demonstrated in Fig. 1.

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.



(a) Cross-chunk Interference (b) Directional Conflict (c) Loss Interference

Figure 1: **Cross-chunk interference under naive TTL.** Measurements are conducted with Qwen3-1.7B-Base on RULER 4K contexts. **(a)** Updating on an incoming chunk can reduce its own next-token prediction (NTP) loss $\mathcal{L}_i$ while increasing a previous chunk's loss $\mathcal{L}_{i-1}$. **(b)** Directional conflicts occur across chunk pairs and are particularly frequent for earlier chunks. **(c)** Adapting to incoming chunks often increases the NTP loss of earlier chunks, indicating substantial interference during sequential adaptation.

Another avenue is to learn the adaptation process offline, such as through meta-learning (Chen et al., 2026; Tandon et al., 2025) or dedicated pretraining (Sun et al., 2025; Behrouz et al., 2025). Yet, it shifts much of the adaptation burden to pre-deployment training, introducing additional computation and data requirements.

The sequential nature of test-time adaptation reveals a close connection between TTL and *online continual learning*. During test-time adaptation, a (potentially long) context is processed sequentially, with successive portions optimized under their own self-supervised objectives. Each portion can therefore be regarded as a new 'task', requiring the model to incorporate incoming information while retaining knowledge acquired from earlier portions. Our motivating experiments in Figure 1 support this perspective: under naive sequential TTL, update directions from different context portions frequently conflict, and adapting to an incoming portion can immediately increase the prediction loss on earlier ones.

Therefore motivated, we propose *Conflict-Aware Test-time Learning* (CATL), a plug-and-play TTL method that introduces a conflict-aware update mechanism guided by a compact historical write memory. The memory consists of three complementary components: *(i)* a subspace describing where previous context has been written, *(ii)* accumulated directions describing how those writes are oriented, and *(iii)* importance weights indicating how strongly individual directions should be protected. For each update, CATL measures its alignment with historical directions within the subspace and selectively attenuates components that conflict with earlier writes, while preserving compatible and orthogonal directions. In this way, CATL limits destructive interference while preserving the flexibility to reinforce existing knowledge and incorporate new information.

We validate the effectiveness of CATL on two challenging long-context benchmarks (BABILong and RULER) using Qwen3 base models with 1.7B, 4B, and 8B parameters. CATL achieves the highest overall accuracy and, at the longest evaluated context length of 32K, outperforms all baselines on all benchmarks. In particular, on BABILong with Qwen3-8B-Base, CATL yields a 38.1% relative accuracy gain over the base model at 32K. These benefits also extend to instruction-tuned models with hybrid local-global attention (Gemma3-1B) and full attention (Qwen3-4B), corroborating CATL's generalizability across model families and attention mechanisms.

2 Preliminaries

Test-time learning. TTL adapts a designated subset of model parameters ω, termed *fast weights*, using a self-supervised objective constructed from the test input, while keeping the remaining parameters θ frozen (Schmidhuber, 1992). It enables the model to encode test-time information directly into its parameters. Given a test-time input $\boldsymbol{c} = (c_1, \ldots, c_N)$, TTL commonly adopts the next-token

prediction loss as its self-supervised objective (Wang et al., 2024; Chen et al., 2026):

$$\mathcal{L}(\omega) = -\sum_{n=2}^{N} \log p_{\theta,\omega}\left(c_n \mid c_{<n}\right), \tag{1}$$

where c_n denotes the n-th token. Let d_k denote the update direction determined by the optimizer at adaptation step k. The fast weights are iteratively adapted to minimize $\mathcal{L}(\omega)$:

$$\omega_k = \omega_{k-1} - \eta_k d_k, \tag{2}$$

where η_k denotes the learning rate. In general, d_k may depend on the current gradient $\nabla\mathcal{L}(\omega_{k-1})$ as well as the optimizer state accumulated from preceding steps.

Subspace-based continual learning. Continual learning (CL) aims to train the model to learn from a sequence of tasks without forgetting previously learned knowledge. A common subspace-based CL strategy collects update directions associated with earlier tasks into a matrix H and extracts an orthonormal basis $S \in \mathbb{R}^{d_1 \times r}$ spanning the principal subspace (Farajtabar et al., 2020; Saha et al., 2021). This subspace captures the dominant directions associated with preceding learning, which can then be used to constrain subsequent updates and reduce interference with acquired knowledge. For example, given the SVD of the historical update matrix $H = U\Sigma V^\top$, the principal subspace can be represented by the leading r left singular vectors $S = U_{:,1:r}$. For an update direction $D \in \mathbb{R}^{d_1 \times d_2}$, its components parallel and orthogonal to the subspace are given by

$$D^{\text{proj}} = SS^\top D, \qquad D^{\text{res}} = (I - SS^\top)D. \tag{3}$$

A standard subspace-projection approach updates parameters along D^{res}, removing the components within the historical subspace and thereby avoiding modification along directions associated with previous tasks.

3 Conflict-Aware Test-Time Learning

A central problem in TTL is how to rapidly learn useful information from test-time contexts with limited updates. It is particularly challenging for long-context cases because different context portions can induce high updating variance in high-dimensional parameter spaces. As shown in Figure 1, under standard sequential TTL, update directions from different context portions frequently conflict, and adapting to a new portion can immediately increase the prediction loss on previously processed ones. This reveals a test-time *stability-plasticity dilemma*: effective TTL should remain sufficiently plastic to efficiently absorb incoming information while preserving useful information acquired from earlier context portions. This stability-plasticity tension reveals a structural connection between TTL and CL. As a long context is processed sequentially, each context chunk induces a new self-supervised objective while updating the same fast weights, closely resembling the sequential learning of multiple tasks over shared parameters. This correspondence motivates leveraging CL techniques to mitigate inter-context interference and stabilize test-time adaptation.

3.1 Sequential Context Adaptation

Let $\mathcal{C} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_M)$ denote the ordered context of a test-time instance, partitioned into M chunks that are processed sequentially. For each $\boldsymbol{c}_i$, TTL minimizes the next-token prediction loss in Eq. (1) by adapting the down-projection matrix in each of the L layers, while all remaining parameters θ are kept frozen. Denote $W_{i,t}^{(\ell)} \in \mathbb{R}^{d_1 \times d_2}$ as the fast-weight matrix of layer ℓ at adaptation step t on chunk $\boldsymbol{c}_i$. The first chunk is initialized from the pretrained fast weights, while each subsequent chunk inherits the adapted weights from the preceding chunk, $W_{i,0}^{(\ell)} = W_{i-1,T}^{(\ell)}$, where T denotes the number of adaptation steps performed per chunk. $W^{(\ell)}$ acts as a shared state that carries information forward through the chunk sequence.

We denote $D_{i,t}^{(\ell)}$ as the update direction determined by the optimizer for $W_{i,t}^{(\ell)}$, and $\widetilde{D}_{i,t}^{(\ell)}$ as the effective direction actually applied (detailed in Section 3.3). The fast weights are updated iteratively:

$$W_{i,t}^{(\ell)} = W_{i,t-1}^{(\ell)} - \eta \widetilde{D}_{i,t}^{(\ell)}, \tag{4}$$

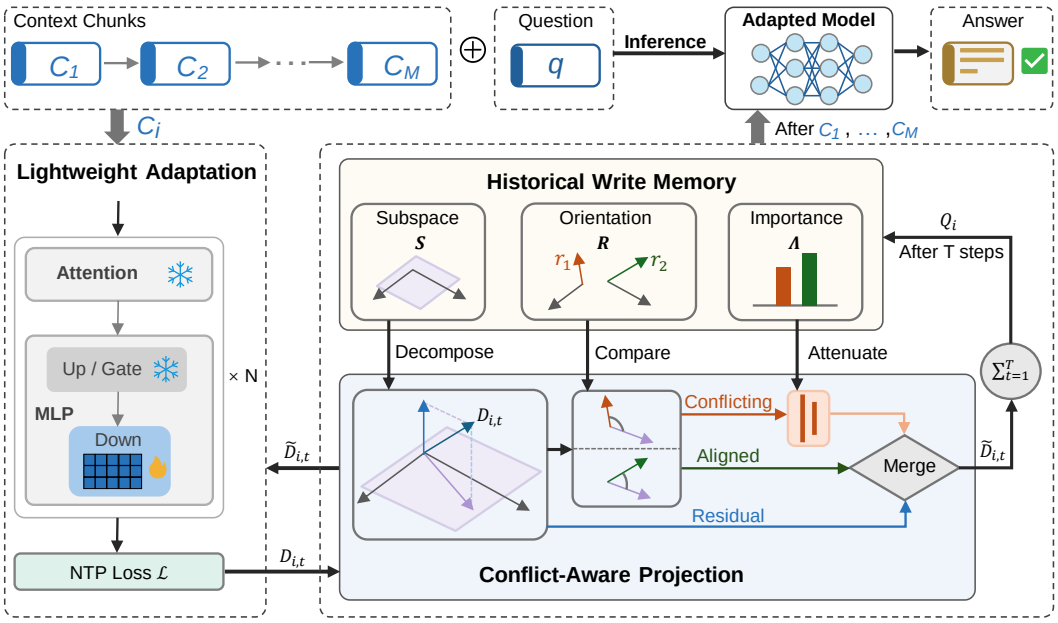


Figure 2: **Overview of CATL.** Context chunks sequentially adapt the MLP down-projection fast weights via the NTP loss, with all other parameters frozen. Guided by the historical write memory (S, R, Λ), CATL preserves the update direction's orthogonal residual and historically aligned components, and attenuates conflicting components according to their importance, yielding the effective direction $\widetilde{D}_{i,t}$. After each chunk, the applied directions form Q_i to update the memory's subspace, orientation, and importance for the next chunk.

where $W_{1,0}^{(\ell)}$ is the pretrained fast weights, and η is the learning rate. To summarize what a chunk writes into this state, we define its *chunk-level write* as the accumulated applied direction:

$$Q_i^{(\ell)} = \sum_{t=1}^{T} \widetilde{D}_{i,t}^{(\ell)} = -\frac{1}{\eta}\left(W_{i,T}^{(\ell)} - W_{i,0}^{(\ell)}\right). \tag{5}$$

The equality follows by telescoping the updates. Under a CL view, each $Q_i^{(\ell)}$ records the information contributed by task i, corresponding to chunk $\boldsymbol{c}_i$. Subsequent updates can reinforce or conflict with historical writes. Standard subspace projection protects earlier information by removing update components within a historical subspace, but it also suppresses components that are compatible with or even reinforce previous writes. This is undesirable in TTL, where limited adaptation steps must be used efficiently to absorb beneficial information from each chunk.

3.2 Historical Write Memory

To preserve useful updates within the historical subspace, we need to distinguish components that reinforce previous writes from those that conflict with them. To this end, we first construct a compact *historical write memory* from the chunk-level writes in Eq. (5), which captures the location, orientation, and importance of earlier writes and serves as a reference for assessing subsequent updates. The construction is identical across layers; we omit the layer index (ℓ) hereafter.

After processing chunk i, we denote the historical write memory by $\mathcal{M}_i = (S_i, R_i, \Lambda_i)$. Here, $S_i = [s_{i,1}, \ldots, s_{i,r_i}] \in \mathbb{R}^{d_1 \times r_i}$ is an orthonormal basis spanning the retained historical write subspace and r_i is its current rank; $R_i \in \mathbb{R}^{r_i \times d_2}$ records the accumulated orientation of historical writes along these basis directions, with its j-th row associated with $s_{i,j}$; $\Lambda_i = \mathrm{diag}(\lambda_{i,1}, \ldots, \lambda_{i,r_i}) \in \mathbb{R}^{r_i \times r_i}$ assigns an importance weight to each basis direction, indicating how strongly information along that direction should be protected. Thus, S_i, R_i, and Λ_i respectively encode *where* previous writes lie, *how* they are oriented, and *how strongly* they should be preserved, jointly providing the reference for selectively constraining subsequent writes.

First chunk. For the first nonzero chunk write Q_1, we initialize the historical write memory directly from its dominant structure. We identify the principal write subspace through the singular value decomposition $Q_1 = U_1 \Sigma_1 V_1^\top$. Let $\sigma_{1,j}$ denote the j-th singular value in descending order. We choose the smallest rank k_1 whose leading singular directions capture at least an ϵ_{th} fraction of the total write energy:

$$k_1 = \min\left\{k : \|(Q_1)_k\|_F^2 \geq \epsilon_{\text{th}} \|Q_1\|_F^2\right\}. \tag{6}$$

Here, $(Q_1)_k$ denotes the rank-k truncated SVD approximation of Q_1. The corresponding left singular vectors form the initial historical basis $S_1 = U_{1,:,1:k_1}$, giving $r_1 = k_1$. We then express the retained write in this basis as $R_1 = S_1^\top Q_1$, which records its orientation along the historical directions. Following the basis-wise importance scaling of Scaled Gradient Projection (Saha & Roy, 2023), we initialize the protection strength of each retained direction according to its singular value. Denoting the retained singular values by $\sigma_1 = (\sigma_{1,1}, \ldots, \sigma_{1,k_1})^\top$, we set

$$\lambda_{1,j} = \frac{(\alpha+1)\sigma_{1,j}}{\alpha\sigma_{1,j} + \max(\sigma_1)}, \qquad \Lambda_1 = \text{diag}(\lambda_{1,1}, \ldots, \lambda_{1,k_1}), \tag{7}$$

where $\alpha \geq 0$ controls the importance scaling. In this way, directions carrying stronger components of the initial write receive greater protection in subsequent adaptation.

Subsequent chunks. For a subsequent chunk $i \geq 2$, we update the memory components sequentially. We first expand the historical write subspace to capture novel directions introduced by the current chunk. Specifically, we decompose Q_i into the component already represented by the existing subspace and its orthogonal residual:

$$Q_i^{\text{proj}} = S_{i-1} S_{i-1}^\top Q_i, \qquad Q_i^{\text{res}} = \left(I - S_{i-1} S_{i-1}^\top\right) Q_i. \tag{8}$$

After the SVD $Q_i^{\text{res}} = \widehat{U}_i \widehat{\Sigma}_i \widehat{V}_i^\top$, we choose the smallest number k_i of additional basis directions such that, together with the energy already captured by S_{i-1}, the retained energy reaches the threshold:

$$k_i = \min\left\{k \geq 0 : \|Q_i^{\text{proj}}\|_F^2 + \|(Q_i^{\text{res}})_k\|_F^2 \geq \epsilon_{\text{th}} \|Q_i\|_F^2\right\}. \tag{9}$$

The selected left singular vectors are appended to the existing basis,

$$S_i = [S_{i-1}, \widehat{U}_{i,:,1:k_i}], \qquad r_i = r_{i-1} + k_i. \tag{10}$$

The orthogonal residual keeps the orthonormality of S_i. With S_i established, we update the directional memory by accumulating the coordinates of the current write:

$$R_i = \begin{bmatrix} R_{i-1} \\ \mathbf{0} \end{bmatrix} + S_i^\top Q_i, \tag{11}$$

thereby $R_i \in \mathbb{R}^{r_i \times d_2}$. The zero rows extend the previous directional memory to the enlarged basis, while $S_i^\top Q_i$ adds the current write coordinates along all retained historical directions.

Finally, we update the importance of each basis direction according to its contribution to the current write. For each newly added basis direction, $\sigma_{i,j}$ is given by the corresponding retained singular value of Q_i^{res}. For an existing basis $s_{i-1,j}$, which is not obtained from the residual SVD, we measure its contribution by

$$\sigma_{i,j} = \left\|s_{i-1,j}^\top Q_i\right\|_2, \qquad j \leq r_{i-1}. \tag{12}$$

Intuitively, it measures how strongly the current write acts along the existing basis direction $s_{i-1,j}$: a larger value indicates that this direction contributes more substantially to the current write and should receive stronger protection. Collecting these values in basis order as $\sigma_i = (\sigma_{i,1}, \ldots, \sigma_{i,r_i})^\top$, analogously to Eq. (7), we compute its current importance as $\lambda_{i,j}^{\text{cur}} = (\alpha+1)\sigma_{i,j}/(\alpha\sigma_{i,j} + \max(\sigma_i))$. The accumulated importance is then updated as

$$\lambda_{i,j} = \begin{cases} \min\left(1, \lambda_{i-1,j} + \lambda_{i,j}^{\text{cur}}\right), & j \leq r_{i-1}, \\ \lambda_{i,j}^{\text{cur}}, & r_{i-1} < j \leq r_i, \end{cases} \tag{13}$$

and we set $\Lambda_i = \text{diag}(\lambda_{i,1}, \ldots, \lambda_{i,r_i})$. Accumulating importance strengthens protection for write directions repeatedly used across chunks against subsequent conflicting updates. The updated memory $\mathcal{M}_i = (S_i, R_i, \Lambda_i)$ then provides the historical reference for adaptation on the next chunk.

3.3 CONFLICT-AWARE PROJECTION

With the historical write memory established, we use it to selectively *attenuate* optimizer-proposed directions that conflict with previous writes. Of note, throughout the T adaptation steps on chunk i, the memory $\mathcal{M}_{i-1}$ remains fixed and is updated only after the chunk is fully processed.

At step t, we first decompose the update direction $D_{i,t}$ (defined in Section 3.1) into its coordinates within the historical subspace and the residual component:

$$B_{i,t} = S_{i-1}^{\top} D_{i,t}, \qquad D_{i,t}^{\text{res}} = D_{i,t} - S_{i-1} B_{i,t}. \tag{14}$$

The residual component lies outside the historical subspace and is therefore retained unchanged. For each historical basis direction j, let $b_{i,t,j} = B_{i,t,j,:}$ denote the corresponding candidate coordinates and $r_{i-1,j} = R_{i-1,j,:}$ the accumulated historical coordinates. We measure their interaction by

$$a_{i,t,j} = \langle r_{i-1,j}, b_{i,t,j} \rangle. \tag{15}$$

A positive score indicates that the current direction reinforces the accumulated historical write, whereas a negative one indicates a conflict. We attenuate only the conflicting coordinates according to their importance:

$$\tilde{b}_{i,t,j} = (1 - \lambda_{i-1,j} \mathbb{I}[a_{i,t,j} < 0])\, b_{i,t,j}, \qquad \widetilde{D}_{i,t} = D_{i,t}^{\text{res}} + S_{i-1} \widetilde{B}_{i,t}, \tag{16}$$

where $\widetilde{B}_{i,t}$ stacks the attenuated rows $\tilde{b}_{i,t,j}$. Therefore, a conflicting component is removed when $\lambda_{i-1,j} = 1$ and partially retained when $0 < \lambda_{i-1,j} < 1$. Components with nonnegative alignment and the residual component are preserved, allowing subsequent chunks to reinforce existing writes and incorporate information outside the retained subspace. This selective rule avoids suppressing a component solely because it overlaps with history.

Remark (Geometric properties). The effect on the stored directional reference can be characterized explicitly: $\langle R_{i-1}, \widetilde{B}_{i,t} \rangle_F - \langle R_{i-1}, B_{i,t} \rangle_F = \sum_{j: a_{i,t,j} < 0} |\lambda_{i-1,j} a_{i,t,j}| \geq 0$. Clearly, only conflicting basis components are moved toward compatibility with historical writes, with the amount of attenuation determined by their importance, while compatible components remain unchanged. Moreover, since S_{i-1} is orthonormal and each scaling factor $1 - \lambda_{i-1,j} \mathbb{I}[a_{i,t,j} < 0]$ lies in $[0, 1]$, it can be shown that the resulting direction satisfies $\|\widetilde{D}_{i,t}\|_F \leq \|D_{i,t}\|_F$. It reveals that the conflict-aware projection is non-expansive, providing a conservative stability property for test-time adaptation. This is because the attenuation selectively suppresses conflicting components without amplifying the update direction or unnecessarily restricting compatible directions.

We apply $W_{i,t} = W_{i,t-1} - \eta \widetilde{D}_{i,t}$ at each step and, after T steps, form Q_i using Eq. (5), which is then used to update the historical memory as described in Section 3.2. The adapted weights are carried forward to the next chunk. For the first chunk, the memory is empty and hence $\widetilde{D}_{1,t} = D_{1,t}$. Thereafter, the memory is updated once per chunk and held fixed throughout all adaptation steps of the subsequent chunk, requiring neither replay of previous chunks nor additional offline training. Persistent storage is $\mathcal{O}(r_i(d_1 + d_2))$ scalars per layer when the diagonal of Λ_i is stored as a vector; the rank can grow across chunks but is bounded by d_1. After processing the final chunk, the adapted model takes the full context and query as input to produce the prediction.

We term the proposed method *Conflict-Aware Test-Time Learning* (CATL), with its pseudocode summarized in Algorithm 1. CATL maintains a compact memory of historical writes and leverages their orientation and importance to selectively attenuate conflicting update components while preserving compatible and residual directions, achieving a better balance between stability and plasticity throughout sequential test-time adaptation. Post-hoc loss visualizations in Appendix F provide further empirical support for CATL's effectiveness in mitigating cross-chunk interference.

4 EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of our method by answering the following questions: (1) Does our method perform well on long-context retrieval and reasoning tasks? (2) Does our method generalize across model families and attention architectures? (3) Which components contribute to the effectiveness of our method?

Table 1: Long-context retrieval evaluation on Qwen3 base models, measured by accuracy (%). BABILong results are averaged over QA1–QA3, while RULER results are averaged over four selected subtasks. The shaded column reports the sample-weighted average across the two benchmarks (↑: percentage-point gain over the strongest baseline for each backbone). Best results are in **bold**.

Method	BABILong				RULER				Average
	4K	8K	16K	32K	4K	8K	16K	32K	
Qwen3-1.7B-Base									
Base Model	17.67	15.33	14.33	10.00	68.60	53.92	50.63	43.62	37.11
qTTT	18.11	**18.44**	13.67	6.89	54.89	43.02	43.11	35.34	31.31
Locas-Memory	16.11	11.00	11.56	8.22	67.96	54.20	**55.08**	44.03	36.63
CATL	**19.00**	13.44	**14.33**	**11.33**	**73.56**	**56.26**	52.68	**46.14**	**38.89** ↑ 1.78
Qwen3-4B-Base									
Base Model	22.67	18.67	9.67	9.67	**82.18**	76.48	71.92	60.39	48.07
qTTT	3.11	0.89	1.67	1.78	78.51	74.14	67.01	52.05	39.61
Locas-Memory	22.00	15.89	**12.78**	10.33	75.61	71.10	67.56	53.33	44.76
CATL	**23.22**	**19.00**	12.33	**11.89**	81.32	**77.23**	**72.09**	**63.51**	**49.14** ↑ 1.07
Qwen3-8B-Base									
Base Model	23.00	17.67	14.00	9.33	**86.06**	79.99	77.97	69.35	51.62
qTTT	14.78	11.56	12.22	10.67	80.83	77.21	70.31	47.57	44.54
Locas-Memory	23.11	19.11	13.33	9.78	80.55	73.94	66.80	64.08	47.77
CATL	**27.00**	**20.11**	**15.11**	**12.89**	86.00	**81.67**	**78.14**	**71.74**	**53.40** ↑ 1.78

4.1 Experimental Setup

Datasets and baselines. We evaluate CATL on two commonly used long-context benchmarks. **(1) BABILong**: We use its QA1–QA3 tasks (Kuratov et al., 2024) to assess performance in needle-in-a-haystack settings, where models must retrieve and reason over task-relevant facts embedded in long distractor contexts. **(2) RULER**: For a complementary evaluation, we consider four of its sub-tasks (Hsieh et al., 2024), covering word-frequency statistics and multi-document question answering. We compare against two representative plug-and-play TTL methods and the unadapted base model: **(1) qTTT** (Bansal et al., 2026), which minimizes the next-token prediction loss on randomly sampled context spans to update the query projection matrices, **(2) Locas-Memory** (Lu et al., 2026), which introduces low-rank sideway FFN memory modules initialized from the pre-trained model and updates them online with a language-modeling objective, and **(3) Base Model**, directly using the original model without any parameter updates or modifications during inference.

Backbone and evaluation. We primarily use the Qwen3-1.7B, Qwen3-4B, and Qwen3-8B base models (Yang et al., 2025) as backbones and evaluate them at context lengths ranging from 4K to 32K. Unless otherwise specified, all reported results are averaged across three random seeds. The TTL configuration and evaluation protocol are detailed in Appendix C.2 and C.3, respectively.

4.2 Results on Long-Context Retrieval and Reasoning

We evaluate CATL against all baselines on BABILong and RULER, using three Qwen3 base models across context lengths from 4K to 32K, with the results reported in Table 1.

BABILong. As shown in the left section of Table 1, CATL achieves the strongest overall performance on BABILong across model scales. While qTTT and Locas-Memory improve over the base model in individual settings, qTTT exhibits substantial variation across backbones, with particularly severe performance degradation on the 4B backbone. In contrast, CATL shows more consistent gains across model scales and context lengths, achieving the best performance in most settings and at every evaluated context length on the 8B backbone. At 32K, CATL outperforms all baselines across all three backbones, with a 3.56 percentage-point improvement over the base model on the 8B backbone. Together, these results demonstrate CATL's effectiveness and consistency across model scales and context lengths, particularly in the longest-context setting.

Table 2: Evaluation of CATL on instruction-tuned backbones with hybrid local-global attention (Gemma3-1B) and full attention (Qwen3-4B). Shaded values are sample-weighted averages (↑: percentage-point gain over the base model). Best results are shown in **bold**.

Backbone	Method	BABILong				RULER				Average
		4K	8K	16K	32K	4K	8K	16K	32K	
Gemma3-1B	Base Model	9.33	4.33	4.00	1.00	38.93	27.28	30.29	21.75	18.89
	CATL	**10.89**	**5.00**	**4.89**	**2.89**	**46.13**	**33.19**	**33.53**	**23.45**	**22.01** ↑3.12
Qwen3-4B	Base Model	35.67	36.33	**38.33**	29.33	**81.73**	75.70	74.63	68.00	57.83
	CATL	**37.78**	**38.00**	38.00	**31.56**	81.27	**75.85**	**76.18**	**69.03**	**58.76** ↑0.93

Table 3: Component ablation of CATL on Qwen3-4B-Base using the 32K settings of BABILong and RULER. Column-wise best results are highlighted in **bold**.

Variant	Components			Accuracy (%)			Δ
	S	Λ	R	RULER	BABILong	Overall	
Naive TTL	–	–	–	62.82	11.22	40.71	–
Subspace-Constrained TTL	✓	–	–	60.75	9.33	38.71	-1.99
Importance-Aware TTL	✓	✓	–	61.24	10.44	39.47	-1.24
CATL (Full)	✓	✓	✓	**63.51**	**11.89**	**41.39**	**+0.68**

RULER. The RULER results in the right section of Table 1 broadly mirror those on BABILong. qTTT underperforms the base model in every evaluated setting, while Locas-Memory provides gains only in a few settings on the 1.7B backbone and degrades performance on the larger backbones. Although CATL yields mixed results at shorter contexts, it outperforms all baselines at 32K across all three backbones, with a 3.12 absolute gain over the base model on the 4B. These results are consistent with the concern that unconstrained successive test-time updates can interfere with previously acquired information, particularly as the context grows. By selectively attenuating conflicting update components while preserving compatible and orthogonal directions, CATL provides a more stable mechanism for accumulating information across sequential context adaptation.

The above results demonstrate CATL's effectiveness at 32K. To further examine how its benefits evolve with context length, we compute each method's relative gain, expressed as the change in accuracy relative to the corresponding base model. Figure 3(a) presents the results on BABILong with the 8B backbone. CATL achieves the largest relative gain at every evaluated length, reaching 38.1% at 32K, whereas qTTT and Locas-Memory yield smaller or negative gains. Additional results across model scales and benchmarks are provided in Appendix E.

4.3 Model Generalizability

We further extend the evaluation from base models to instruction-tuned backbones with different attention mechanisms, comparing a hybrid local–global attention backbone (Gemma3-1B (Team et al., 2025)) with a full-attention backbone (Qwen3-4B (Yang et al., 2025)). The results in Table 2 show that CATL remains effective across both backbones. On Gemma3-1B, CATL outperforms the base model in every evaluated setting, indicating that CATL can improve long-context performance even under a hybrid local-global attention design. CATL also improves the full-attention backbone at 32K. It corroborates the generalizability of CATL across model families and attention mechanisms.

4.4 Ablation Studies

Key component analysis. Table 3 shows that S, Λ, and R play complementary roles in CATL. Retaining only the historical subspace S slightly degrades performance relative to Naive TTL, because subspace membership alone does not distinguish compatible updates from conflicting ones. Adding the importance weights Λ partially recovers performance by modulating how strongly different historical directions are protected, but remains insufficient without the directional memory R to assess update compatibility. Combining the subspace, importance, and directional information enables full CATL to achieve the highest overall accuracy by suppressing harmful conflicts while preserving compatible and orthogonal updates.

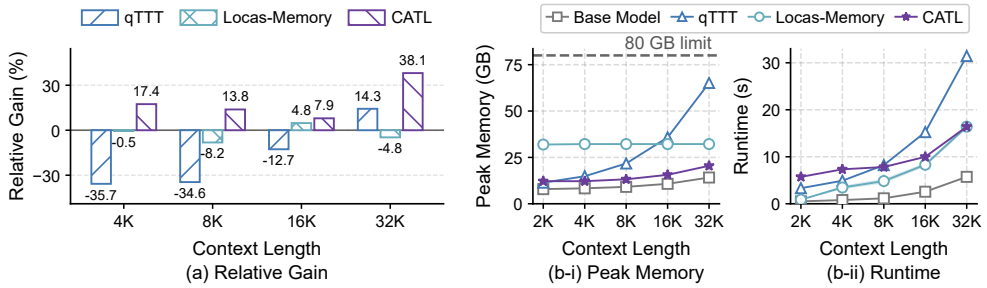


Figure 3: **Relative gains and inference-time efficiency across context lengths. (a)** Relative accuracy gains over the base model on BABILong with Qwen3-8B-Base. **(b-i)** Peak GPU memory usage and **(b-ii)** wall-clock time measured on Qwen3-4B-Base.

Inference-time efficiency. Figure 3(b-i) shows that CATL's peak memory grows only gradually with context length, remaining well below the fixed allocation of Locas-Memory and the steeply increasing footprint of qTTT at longer lengths, with only modest overhead over the base model. Its runtime advantage becomes more pronounced as the context grows: at 32K, CATL remains close to Locas-Memory while requiring approximately half the wall-clock time of qTTT (Figure 3(b-ii)). These results demonstrate that CATL maintains favorable inference-time efficiency as context length increases, supporting practical test-time adaptation to long contexts.

5 Related Work

This section provides a concise overview of related work (see Appendix A for a detailed review).

Test-time learning. Test-time learning (TTL) adapts model parameters using learning signals derived from the current test instance (Sun et al., 2020; Hardt & Sun, 2024). For long-context modeling, existing methods broadly either integrate TTL into the model architecture to compress history into learned parameterized states (Sun et al., 2025; Behrouz et al., 2025; Tandon et al., 2025), or retain a pretrained Transformer and adapt a subset of its parameters on the input context (Bansal et al., 2026; Feng et al., 2026; Chen et al., 2026; Zhu et al., 2026). Several online-adaptation methods process long contexts chunk-by-chunk and encode each chunk into shared adaptation parameters (Wang et al., 2024; Chen et al., 2026). However, successive updates can interfere and overwrite information written by earlier chunks (Yu et al., 2020; Wang et al., 2024).

Continual learning. CL studies learning from sequentially arriving data while retaining previously acquired knowledge (Wu et al., 2024). Existing approaches mitigate forgetting through rehearsal (Lopez-Paz & Ranzato, 2017; Hacohen & Tuytelaars, 2025), parameter regularization (Kirkpatrick et al., 2017; Zenke et al., 2017; Zheng et al., 2026), architectural specialization (Rusu et al., 2016; Le et al., 2026), or subspace constraints on subsequent updates (Farajtabar et al., 2020; Saha et al., 2021; Wang et al., 2021; Qiu et al., 2025). Long-context TTL naturally exhibits a similar sequential-learning structure, making subspace-based CL particularly relevant because it can reduce interference without replaying earlier data. Yet, conventional subspace-based CL is ill-suited to TTL because standard projection suppresses both conflicting updates and compatible directions that could reinforce previous writes under limited adaptation steps.

6 Conclusion

This paper introduces CATL, a plug-and-play TTL method for learning from long contexts while mitigating interference across successive context updates. CATL casts sequential test-time adaptation as an online CL process and maintains a compact historical write memory to selectively attenuate conflicting components while preserving compatible and novel directions. Experiments demonstrate that CATL improves long-context adaptation while retaining favorable inference-time efficiency. Future work could explore more compact or adaptive management of the historical write memory for substantially longer contexts, as well as extending CATL to larger model scales and broader test-time adaptation settings.

AI USE STATEMENT

We used generative AI tools to assist with language polishing and to improve the clarity and presentation of the manuscript. All AI-assisted revisions were carefully reviewed and edited by the authors. The authors take full responsibility for the final content of this paper, including its text, claims, and conclusions.

ETHICS STATEMENT

This work does not involve human subjects, private or personally identifiable data, or deployment in high-stakes settings. Our experiments use existing benchmarks and pretrained language models for methodological evaluation. We do not identify specific ethical risks introduced by this work beyond the general risks associated with the use of large language models.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce our experiments in Appendix C, including detailed descriptions of the datasets, test-time learning setup, evaluation protocols, and prompt templates. The complete CATL algorithm is provided in Appendix B.

REFERENCES

Chenxin An, Jun Zhang, Ming Zhong, Lei Li, Shansan Gong, Yao Luo, Jingjing Xu, and Lingpeng Kong. Why does the effective context length of LLMs fall short? In *International Conference on Learning Representations*. OpenReview.net, 2025.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2025.

Rachit Bansal, Aston Zhang, Rishabh Tiwari, Lovish Madaan, Venkata Sai Surya Subramanyam Duvvuri, Devvrit Khatri, David Brandfonbrener, David Alvarez-Melis, Prajjwal Bhargava, Mihir Kale, et al. Let's (not) just put things in context: Test-time training for long-context LLMs. In *International Conference on Learning Representations*, volume 2026, pp. 112130–112153, 2026.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems*, 2025.

Zeming Chen, Angelika Romanou, Gail Weiss, and Antoine Bosselut. PERK: long-context reasoning as parameter-efficient test-time learning. In Carl Vondrick, Bharath Hariharan, Colin Raffel, Lerrel Pinto, Diyi Yang, and Aleksandra Faust (eds.), *International Conference on Learning Representations*. proceedings.iclr.cc, 2026.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pp. 3762–3773. PMLR, 2020.

Guhao Feng, Shengjie Luo, Kai Hua, Ge Zhang, Wenhao Huang, Di He, and Tianle Cai. In-place test-time training. In *International Conference on Learning Representations*, volume 2026, pp. 114452–114472, 2026.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *CoRR*, abs/2312.00752, 2023.

Guy Hacohen and Tinne Tuytelaars. Predicting the susceptibility of examples to catastrophic forgetting. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In *International Conference on Learning Representations*. OpenReview.net, 2024.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: what's the real context size of your long-context language models? *CoRR*, abs/2404.06654, 2024.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Y. Sorokin, and Mikhail Burtsev. BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. In *Advances in Neural Information Processing Systems*, 2024.

Minh Le, Bao-Ngoc Dao, Huy Nguyen, Quyen Tran, Anh Nguyen, and Nhat Ho. One-prompt strikes back: Sparse mixture of experts for prompt-based continual learning. In *International Conference on Learning Representations*. proceedings.iclr.cc, 2026.

Changhun Lee, Minsang Seok, Jungyu Jin, Younghyun Cho, and Eunhyeok Park. SEAL: scaling to emphasize attention for long-context retrieval. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28942–28955. Association for Computational Linguistics, 2025.

Zhan Ling, Kang Liu, Kai Yan, Yifan Yang, Weijian Lin, Ting-Han Fan, Lingfeng Shen, Zhengyin Du, and Jiecao Chen. LongReason: A synthetic long-context reasoning benchmark via context expansion. *CoRR*, abs/2501.15089, 2025.

Xiaoran Liu, Yuerong Song, Zhigeng Liu, Zengfeng Huang, Qipeng Guo, Zhaoxiang Liu, Shiguo Lian, Ziwei He, and Xipeng Qiu. Beyond real: Imaginary extension of rotary position embeddings for long-context LLMs. In *International Conference on Learning Representations*, volume 2026, pp. 127245–127264, 2026.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pp. 6467–6476, 2017.

Sidi Lu, Zhenwen Liang, Dongyang Ma, Yan Wang, Haitao Mi, and Dong Yu. Locas: Your models are principled initializers of locally-supported parametric memories. *CoRR*, abs/2602.05085, 2026.

OpenCompass Contributors. OpenCompass: A universal evaluation platform for foundation models. `https://github.com/open-compass/opencompass`, 2023.

Xuan Ouyang, Zefan Cai, and Junjie Hu. Test-time training with next-token prediction. *CoRR*, abs/2606.21803, 2026.

Ziqing Qiao, Yinuo Xu, Chaojun Xiao, Zhou Su, Zihan Zhou, Yingfa Chen, Xiaoyue Xu, Xu Han, and Zhiyuan Liu. Rethinking the role of efficient attention in hybrid architectures. *CoRR*, abs/2606.15378, 2026.

Haomiao Qiu, Miao Zhang, Ziyue Qiao, Weili Guan, Min Zhang, and Liqiang Nie. SplitLoRA: Balancing stability and plasticity in continual learning through gradient space splitting. *CoRR*, abs/2505.22370, 2025.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Gobinda Saha and Kaushik Roy. Continual learning with scaled gradient projection. In *Thirty-Seventh AAAI Conference on Artificial Intelligence*, pp. 9677–9685. AAAI Press, 2023.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*. OpenReview.net, 2021.

Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248. PMLR, 2020.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): RNNs with expressive hidden states. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Arnuv Tandon, Karan Dalal, Xinhao Li, Daniel Koceja, Marcel Rød, Sam Buchanan, Xiaolong Wang, Jure Leskovec, Sanmi Koyejo, Tatsunori Hashimoto, Carlos Guestrin, Jed McCaleb, Yejin Choi, and Yu Sun. End-to-end test-time training for long context. *CoRR*, abs/2512.23675, 2025.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Xin Teng, Canyu Zhang, Shaoyi Zheng, Danyang Zhuo, Tianyi Zhou, and Shenji Wan. InfoFlow KV: Information-flow-aware KV recomputation for long context. *arXiv preprint arXiv:2603.05353*, 2026.

Blerta Veseli, Julian Chibane, Mariya Toneva, and Alexander Koller. Positional biases shift as inputs approach context window limits. *CoRR*, abs/2508.07479, 2025.

Tomoya Wakayama. A decision-theoretic view of test-time training: When, how far, and which directions to adapt. *CoRR*, abs/2606.15569, 2026.

Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 184–193. Computer Vision Foundation / IEEE, 2021.

Yan Wang, D. Ma, and Deng Cai. With greater text comes greater necessity: Inference-time training helps long text generation. *CoRR*, abs/2401.11504, 2024.

Ziqi Wang, Xiusi Chen, Gaotang Li, Heng Ji, and Tong Zhang. When test-time training fails: A critical analysis of robustness and hyperparameter sensitivity. *Trans. Mach. Learn. Res.*, 2026, 2026.

Davis Wertheimer, Aozhong Zhang, Derrick Liu, Penghang Yin, and Naigang Wang. Frayed RoPE and long inputs: A geometric perspective. *CoRR*, abs/2603.18017, 2026.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. Continual learning for large language models: A survey. *CoRR*, abs/2402.01364, 2024.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In *International Conference on Learning Representations*. OpenReview.net, 2025.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. In *International Conference on Learning Representations*. OpenReview.net, 2025.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems 37*, 2024.

Xi Ye, Wuwei Zhang, Fangcong Yin, Howard Yen, and Danqi Chen. DySCO: Dynamic attention-scaling decoding for long-context language models. *arXiv preprint arXiv:2602.22175*, 2026.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to evaluate long-context models effectively and thoroughly. In *International Conference on Learning Representations*. OpenReview.net, 2025.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 2020.

Xiaopeng Yuan, Zebin Wang, Suwen Wang, Zongxin Yang, Haohan Wang, and Yushun Dong. EASE-TTT: evidence-aligned selective test-time training for long-context question answering. *CoRR*, abs/2606.06906, 2026.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995. PMLR, 2017.

Wuwei Zhang, Fangcong Yin, Howard Yen, Danqi Chen, and Xi Ye. Query-focused retrieval heads improve long-context reasoning and re-ranking. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 23791–23805. Association for Computational Linguistics, 2025.

Yaoyue Zheng, Yin Zhang, Joost van de Weijer, Gido van de Ven, Shaoyi Du, Xuetao Zhang, and Zhiqiang Tian. Revisiting weight regularization for low-rank continual learning. In *International Conference on Learning Representations*, volume 2026, pp. 42895–42919, 2026.

Xinyu Zhu, Zhe Xu, Xiaohan Wei, Yunchen Pu, Fei Tian, Chonglin Sun, Kaushik Rangadurai, Hua Zhi, Frank Shyu, Sandeep Pandey, Luke Simon, Yu Meng, and Xi Liu. Self-guided test-time training for long-context LLMs. *CoRR*, abs/2607.09415, 2026.

A Related Work

Test-time learning. The central idea of TTL is to update model parameters using a learning signal derived from the current test instance, allowing the model to adapt during inference (Sun et al., 2020; Hardt & Sun, 2024). For long-context modeling, existing TTL methods broadly follow two directions. A model-design line integrates TTL into the sequence-model architecture and its training procedure, using learned parameterized states to compress history and reduce the computation and memory costs of self-attention (Sun et al., 2025; Behrouz et al., 2025; Tandon et al., 2025). An online-adaptation line instead retains a pretrained Transformer and updates a restricted subset of its parameters on the input context to improve retrieval, understanding, and reasoning (Bansal et al., 2026; Feng et al., 2026; Chen et al., 2026; Zhu et al., 2026).

Within online adaptation, several methods divide long contexts into chunks and encode their content into shared adaptation parameters by minimizing the next-token prediction loss (Wang et al., 2024; Chen et al., 2026). While chunk-wise processing makes online adaptation more tractable for long inputs, successive chunks repeatedly modify the same shared parameters without explicitly protecting information written by earlier chunks. Consequently, updates induced by different chunks can conflict (Yu et al., 2020; Wang et al., 2024), potentially overwriting previously encoded information as adaptation proceeds. In contrast, CATL explicitly maintains a compact memory of historical writes and uses their subspace, orientation, and importance to selectively attenuate conflicting components of subsequent updates while preserving compatible and novel directions.

Long contexts in LLMs. Context windows of frontier LLMs have rapidly expanded into million-token regimes (Xu et al., 2026; Team et al., 2026). However, the ability to accept such inputs does not imply that the information they contain can be reliably utilized. Evaluations on realistic understanding and reasoning tasks show that performance can deteriorate as context length increases or tasks require global information integration (Bai et al., 2025; Yen et al., 2025; Ling et al., 2025), motivating research on both effective context utilization and efficient long-context processing. For effective utilization, one line studies long-context degradation through the lens of positional effects. It identifies imbalanced exposure to relative distances (An et al., 2025), length-dependent positional biases (Veseli et al., 2025), and the geometry of rotary position embeddings (RoPE) (Wertheimer et al., 2026) as factors that limit reliable access to distant evidence, and develops methods that shift or extend positional representations to mitigate these effects (Liu et al., 2026).

Another line focuses on retrieval-specialized attention mechanisms. Mechanistic analyses identify attention heads that play important roles in factual retrieval (Wu et al., 2025), while subsequent methods identify query-responsive heads or modulate attention weights to emphasize relevant evidence during retrieval and decoding (Zhang et al., 2025; Lee et al., 2025; Ye et al., 2026). Meanwhile, long contexts pose substantial computational and memory challenges. The KV cache grows linearly with sequence length during autoregressive decoding, while full self-attention incurs quadratic computation during prefilling. Recent methods address these costs by tailoring KV-cache allocation to different attention-head functions (Xiao et al., 2025) or selectively recomputing KV states (Teng et al., 2026). A complementary architectural direction replaces dense attention with recurrent dynamics to reduce the computational and memory costs of long-sequence modeling, as in DeltaNet (Yang et al., 2024) and Mamba (Gu & Dao, 2023). Yet, controlled studies of hybrid attention and recurrent architectures suggest that long-range retrieval remains primarily supported by their full-attention layers (Qiao et al., 2026). Unlike these approaches, which primarily modify positional representations, attention computation, memory management, or sequence-model architectures, our work retains an existing full-attention backbone and improves long-context utilization through test-time parametric adaptation.

Continual learning. Continual learning (CL) studies how models can acquire knowledge from sequentially arriving data while retaining knowledge learned from earlier tasks, a growing concern for LLMs that must incorporate new information without repeated full retraining (Wu et al., 2024). Existing CL methods mitigate forgetting through several complementary mechanisms. Rehearsal-based approaches selectively revisit stored examples from previous tasks (Lopez-Paz & Ranzato, 2017; Hacohen & Tuytelaars, 2025), regularization-based approaches protect parameters important to earlier tasks (Kirkpatrick et al., 2017; Zenke et al., 2017; Zheng et al., 2026), and architecture-based approaches allocate or route specialized capacity to reduce interference across different

Algorithm 1 Conflict-Aware Test-Time Learning (CATL)

Input: Pretrained model $f_{\theta,W_{\mathrm{pre}}}$ with frozen parameters θ and adaptable fast weights W_{pre}; ordered chunks $\mathcal{C} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_M)$; query q; optimizer Opt; hyperparameters $(T, \eta, \epsilon_{\mathrm{th}}, \alpha)$.
Output: Answer $\hat{y}$.

$W_{0,T} \leftarrow W_{\mathrm{pre}}$; $\mathcal{M}_0 = (S_0, R_0, \Lambda_0) \leftarrow (\varnothing, \varnothing, \varnothing)$
for $i = 1, \ldots, M$ **do**
  $W_{i,0} \leftarrow W_{i-1,T}$; reset optimizer state
  for $t = 1, \ldots, T$ **do**
    $D_{i,t} \leftarrow \mathsf{Opt}(\nabla_W \mathcal{L}_i(W_{i,t-1}))$
    Obtain $\widetilde{D}_{i,t}$ using $\mathcal{M}_{i-1}$ via Eq. (16)
    $W_{i,t} \leftarrow W_{i,t-1} - \eta \widetilde{D}_{i,t}$
  end for
  Form the applied chunk write Q_i via Eq. (5)
  $\mathcal{M}_i \leftarrow \mathcal{M}_{i-1}$
  if $\|Q_i\|_F > 0$ **then**
    Initialize or expand S_i via Eqs. (6) and (9)
    Accumulate R_i via Eq. (11)
    Update Λ_i via Eqs. (7) and (13)
  end if
end for
return $\hat{y} \leftarrow \mathsf{Decode}_{\theta,W_{M,T}}(\mathcal{C}, q)$

tasks (Rusu et al., 2016; Le et al., 2026). Another line controls the geometry of learning updates by deriving protected subspaces from earlier gradients, activations, or feature covariances and constraining subsequent updates to reduce interference with previously acquired knowledge (Farajtabar et al., 2020; Saha et al., 2021; Wang et al., 2021; Qiu et al., 2025). Long-context TTL naturally exhibits a sequential-learning structure. This connection motivates leveraging CL techniques to stabilize sequential test-time adaptation. Among them, subspace-based CL is particularly suitable because it constrains the geometry of subsequent updates without requiring replay of previous data, which would incur additional computation at inference time. Yet, conventional subspace-based CL cannot be directly adopted for TTL, as standard projection indiscriminately suppresses update components within the historical subspace. In TTL, where only a few adaptation steps are available, this may suppress compatible directions that reinforce previous writes along with conflicting ones.

B Complete Algorithm of CATL

Algorithm 1 summarizes CATL for one test instance. In the algorithm, W denotes the MLP down-projection fast weights in each selected layer, while S, R, and Λ denote the corresponding historical memory. The NTP gradient is computed jointly over all selected fast weights. The historical memory is fixed during the T updates on a chunk and refreshed only after that chunk is complete.

C Experimental Details

C.1 Dataset Details

BABILong (Kuratov et al., 2024)[1] is a question-answering benchmark that evaluates reasoning over facts embedded in long background texts. We use three tasks: QA1 (single-hop) identifies a character's latest location from one supporting fact (e.g., "Mary went to the garden."; Q: "Where is Mary?"; A: *garden*); QA2 (two-hop) infers an object's current location from pickup or drop events and the character's location using two supporting facts (e.g., "Mary grabbed the football there. Mary went back to the garden."; Q: "Where is the football?"; A: *garden*); and QA3 (three-hop) determines where an object was before a specified location using three supporting facts (e.g., "Mary got the football. Mary went back to the kitchen. Mary journeyed to the garden."; Q: "Where was the football before the garden?"; A: *kitchen*).

[1] https://huggingface.co/datasets/RMT-team/babilong

RULER (Hsieh et al., 2024) is a synthetic long-context benchmark with controllable input length and task complexity. We generate evaluation samples using the RULER dataset generators provided by OpenCompass (OpenCompass Contributors, 2023). To evaluate information aggregation across long contexts, we use common-word extraction (CWE), which identifies a fixed set of common words among increasingly many uncommon words, and frequent-word extraction (FWE), which identifies the three most frequent words in the context. We also use its SQuAD and HotpotQA question-answering tasks, which place answer-bearing paragraphs among distractor paragraphs and test single-hop and multi-hop QA, respectively. Task-specific prompt templates are provided in Appendix C.4.

C.2 Test-Time Learning Setup

We update only the MLP down-projection matrices in the final six Transformer layers of models with at most 4B parameters and the final four layers of the 8B models, while keeping all remaining parameters frozen. To limit the number of adaptation steps and the associated TTL overhead as context length increases, we use chunk sizes of 512, 1024, 2048, and 4096 tokens for the 4K, 8K, 16K, and 32K context-length settings, respectively; variable-length inputs follow the same token-length bucket policy. Chunks are processed sequentially without overlap. Unless otherwise specified, we fix the number of updates per chunk to $T = 2$, the scaling coefficient to $\alpha = 5$, and the energy threshold to $\epsilon_{\text{th}} = 0.98$ across all evaluated models and datasets. To balance adaptation to the current context with preservation of pretrained knowledge, we use configuration-specific learning rates within $\eta \in [1\text{e}-5,\ 1\text{e}-4]$, held constant throughout each run. We use the Adam optimizer and clip gradients to a global ℓ_2 norm of 1.0 before computing optimizer updates. Each test instance is processed independently with a batch size of one, using BF16 mixed-precision computation while maintaining the adapted weights, optimizer states, and historical write memory in FP32. Optimizer states are reset at each chunk boundary, whereas adapted weights and historical memory persist across chunks within the same instance. Before each new instance, we restore the pretrained weights and clear the historical memory. Sensitivity to the key hyperparameters is examined in Appendix D.2.

C.3 Evaluation Protocol

Generation protocol. We use a unified generation protocol across all datasets. For experiments run through our evaluation pipeline, responses are produced with the standard Hugging Face Transformers `generate` interface[2]. We use greedy decoding (`do_sample=False`), with a shared limit of 128 new tokens and early stopping at the end-of-sequence token.

RULER scoring. We follow the official scoring protocol, which uses case-insensitive substring matching. For common-word and frequent-word extraction, an example receives the fraction of reference words appearing in the generated response, allowing partial credit. For the SQuAD and HotpotQA tasks, an example receives a score of one if any accepted reference answer appears in the response, and zero otherwise. We report the mean example score, expressed as a percentage, for each task and across the evaluated tasks.

BABILong scoring. For QA1–QA3, we follow the label-level accuracy metric implemented in the officially released BABILong evaluation code (Kuratov et al., 2024). We identify location labels in the first generated sentence, discard text following generated context or example markers, and exclude location labels already present in the question. A response is correct only when the remaining predicted labels consist solely of the gold location. Responses with no identifiable label or multiple candidate labels are counted as incorrect. Accuracy is computed separately for each task and context length.

C.4 Prompt Templates

Here, we list the prompts used in this paper across different datasets:

For the RULER benchmark, we use different prompts for different tasks:

[2] `https://github.com/huggingface/transformers/releases/tag/v5.5.3`.

Prompt for Common-Word Extraction (CWE)

Below is a numbered list of words.
In these words, some appear more often than others.
Memorize the ones that appear most often.
1. {word_1} 2. {word_2} . . . N. {word_N}
Question: What are the 10 most common words in the above list? Answer: The top 10 words that appear most often in the list are:

Prompt for Frequent-Word Extraction (FWE)

Read the following coded text and track the frequency of each coded word.
Find the three most frequently appeared coded words. {coded_text}
Question: Do not provide any explanation. Please ignore the dots '....'.
What are the three most frequently appeared words in the above coded text? Answer: According to the coded text above, the three most frequently appeared words are:

Prompt for SQuAD and HotpotQA

Answer the question based on the given documents. Only give me the answer and do not output any other words.

The following are given documents.

Document 1:
{document_1}

Document 2:
{document_2}

. . .

Document N:
{document_N}

Answer the question based on the given documents. Only give me the answer and do not output any other words.

Question: {question} Answer:

For the BABILong benchmark, we use the following prompt:

Prompt for BABILong: QA1–QA3

{context}

{question}

The correct answer is:

C.5 Baseline Implementation Details

qTTT. As the official source code was unavailable at the time of our experiments, we implement qTTT following the original paper and adopt the LoRA-based adaptation practices described in prior work (Yuan et al., 2026; Zhu et al., 2026). We freeze the base model and apply LoRA adapters to the query projection matrices W_Q in all attention layers. Following the paper's default adaptation schedule, we perform 32 updates on uniformly sampled 128-token context spans using the next-token prediction objective and a fixed learning rate of $1e{-}5$, while keeping the KV cache from a single context prefill fixed. After adaptation, we reuse the context KV cache to generate the final answer.

Table 4: Optimizer sensitivity of CATL across Qwen3 base models on RULER at 32K context length. Scores are average accuracy (%) (↑ indicates gains over the corresponding base model in percentage points). Column-wise best scores are shown in **bold**.

Method	Qwen3-1.7B	Qwen3-4B	Qwen3-8B
Base Model	43.62	60.39	69.35
CATL + Adam	**46.14** ↑2.52	**63.51** ↑3.12	**71.74** ↑2.39
CATL + AdamW	45.71 ↑2.09	63.08 ↑2.68	71.63 ↑2.28
CATL + SGD (Momentum)	45.43 ↑1.81	61.73 ↑1.33	70.98 ↑1.63
CATL + Muon	45.24 ↑1.62	62.45 ↑2.06	71.36 ↑2.01

Locas-Memory. We implement Locas-Memory following the original paper and use their official released open-source implementation, adapting it to our long-context question-answering setting. Following the streaming configuration in the authors' official implementation, we use 2,048-token windows with a stride of 1,024 tokens and perform one optimization step per window using the next-token prediction loss on the final 1,024 tokens. The learning rate is set to $4\mathrm{e}{-3}$, following the original paper. For each example, we independently initialized and updated the memory, then used it together with the input context and question to generate the final answer.

D Additional Analyses

D.1 Optimizer Sensitivity Analysis

CATL regulates optimizer-proposed update directions and is therefore not restricted to a specific optimizer. To assess this compatibility, we evaluate CATL with Adam, AdamW, momentum SGD, and Muon on RULER at 32K across three Qwen3 base models. As shown in Table 4, CATL consistently outperforms the corresponding base model across all tested optimizer and backbone combinations. These results support CATL's applicability across different optimizers, beyond the default Adam configuration.

D.2 Hyperparameter Sensitivity Analysis

Starting from the default configuration used in the main experiments, we vary one hyperparameter at a time and evaluate all three Qwen3 base models on RULER at 32K, with chunk size fixed at 4096. As shown in Table 5, CATL exhibits limited sensitivity to its memory-specific hyperparameters within the tested ranges. Across the three backbones, varying $\alpha \in \{1, 5, 10\}$ changes the mean score by at most 0.43 percentage points, and varying $\epsilon_{\mathrm{th}} \in \{0.90, 0.95, 0.98\}$ changes it by approximately 0.93 percentage points, measured as the difference between the highest and lowest scores in each sweep. All tested values of these two parameters retain improvements over the base model. For the learning-rate sweep, the default RULER setting (η = 1e-5) provides competitive performance across all three backbones.[3] These results characterize sensitivity on RULER at 32K context length, rather than establish a universally optimal learning rate across benchmarks.

The default configuration provides competitive performance and a balanced choice across model scales. It achieves the best score on the 1.7B backbone, the second-best score on the 4B backbone, and remains within 0.37 percentage points of the best score on the 8B backbone. The improved performance with $T = 4$ on the 4B and 8B models suggests that additional adaptation steps can help the models incorporate context information more effectively, although this requires twice as many adaptation updates as the default $T = 2$. The default configuration therefore offers a practical trade-off between predictive performance and adaptation time cost.

[3]Since RULER's word-frequency tasks require accurate aggregation across the full context, a relatively small learning rate may help preserve this capability while adapting to the current context.

Table 5: Hyperparameter sensitivity of CATL across Qwen3 base models on RULER at 32K context length. The shaded row denotes the default configuration. Each subsequent row changes only the **bold** hyperparameter value. The best and second-best scores across all listed configurations are shown in **bold** and underlined, respectively.

Setting	T	α	η	ϵ_{th}	RULER 32K (%)		
					1.7B	**4B**	**8B**
Base Model	–	–	–	–	43.62	60.39	69.35
Default CATL	2	5	1e-5	0.98	**46.14**	63.51	71.74
Varying α	2	**1**	1e-5	0.98	45.71	63.25	71.66
	2	**10**	1e-5	0.98	45.98	63.14	71.76
Varying ϵ	2	5	1e-5	**0.90**	45.67	63.06	71.61
	2	5	1e-5	**0.95**	45.65	62.58	71.70
Varying T	**1**	5	1e-5	0.98	45.89	61.88	71.26
	4	5	1e-5	0.98	45.03	**64.53**	**72.11**
Varying η	2	5	**1e-6**	0.98	44.39	60.62	69.94
	2	5	**5e-6**	0.98	45.95	62.13	71.21
	2	5	**5e-5**	0.98	43.90	63.08	67.71
	2	5	**1e-4**	0.98	42.84	52.44	65.13

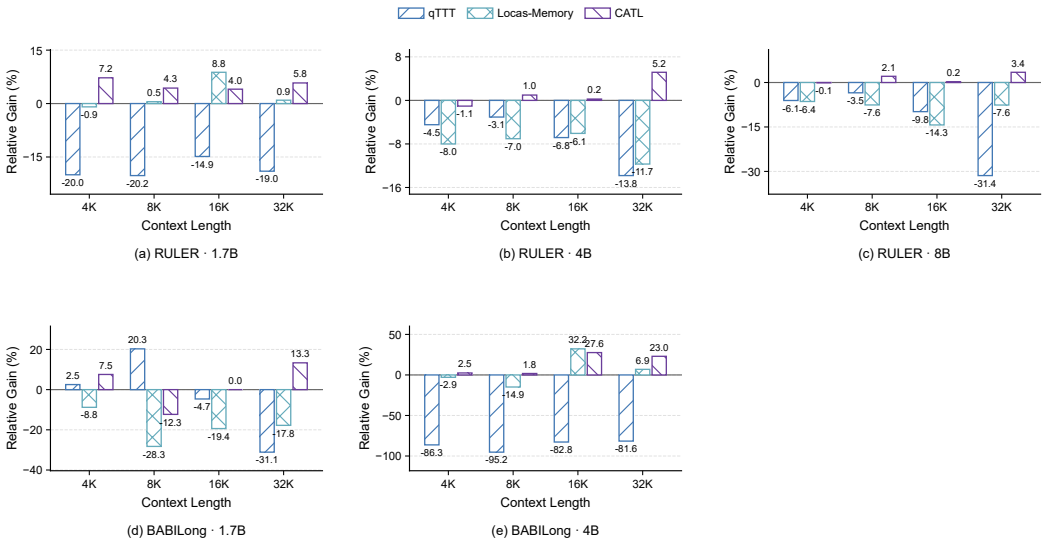


Figure 4: Relative gains over the base model across context lengths. Results are shown for the 1.7B, 4B, and 8B Qwen3 base models on RULER and the 1.7B and 4B variants on BABILong.

E Relative Gains across Context Lengths

Figure 4 shows that CATL achieves positive relative gains and outperforms both competing TTL methods at 32K in all five additional model–benchmark configurations. This extends the long-context advantage observed in Figure 3(a) across model scales and benchmarks.

For example, qTTT's relative gain on RULER with the 8B backbone deteriorates sharply beyond 8K, reaching -31.4% at 32K, where CATL achieves its largest gain in this setting. CATL also maintains positive gains across all lengths on BABILong with the 4B backbone, attaining a 23.0% gain at



(a) RULER · 1.7B (b) RULER · 4B (c) RULER · 8B

Figure 5: Post-hoc loss-increase probabilities under CATL on RULER at 32K, using Qwen3 base models with 1.7B, 4B, and 8B parameters.

32K compared with 6.9% for Locas-Memory. Together, these comparisons demonstrate CATL's consistent long-context benefits across different backbones and benchmarks.

F Post-hoc Visualization of Cross-Chunk Interference

To examine whether CATL's selective attenuation of conflicting update components supports the retention of previously acquired information, we inspect earlier-chunk NTP losses during sequential adaptation. We use the same RULER contexts at 32K for all three Qwen3 base models. We measure each earlier chunk's loss immediately before and after adapting to each subsequent chunk. Figure 5 shows predominantly low loss-increase probabilities across model scales. This pattern is consistent with CATL's design: selectively attenuating conflicts while preserving compatible and orthogonal directions supports stable accumulation of context information in the fast weights.