Test-Time Training for Sequential Recommendation via Efficient Sample Filtering

CPSC 532X: Adaptation & Adaptive Computation Course Project Report
Professor: Evan Shelhamer

Author: Johnson Chen (Student #: 85784080)

Abstract

Sequential recommender systems face a fundamental challenge: models trained on historical data become outdated as user preferences evolve. Recently, Test-Time Training (TTT) was introduced to address this by continuously adapting models during deployment (TTT4Rec), but comes with high computational costs. This project aim to compensate this by exploring Efficient Test-Time Adaptation (ETA) for recommender systems, introducing Sample Filtering that selectively skips redundant or unreliable updates. We find that standard entropy filters fail in sparse recommendation domains and propose a Top-K Entropy approach tailored for this setting. On the Amazon-Video-Games dataset, our method reduces backward gradient passes by 13.73% while maintaining recommendation accuracy (Hit@10: 0.0868) statistically comparable to the TTT4Rec baseline (Hit@10: 0.0872).

1. Introduction

Recommender systems have become the backbone of our digital experiences. At their core, these systems aim to predict what a user will want next based on the context and their past interactions. Sequential Recommender Systems (SRS) have become a major field of development, as they model the order of user actions to capture evolving interests. These models are typically trained on historical data and frozen during deployment.

However, user preferences and interests change constantly. As preferences shift and new items enter the catalog, these static models gradually lose accuracy. They remain frozen until the next expensive retraining cycle, unable to adapt to emerging trends or changing user needs.

To address this, we explore Test-Time Adaptation (TTA), a technique originally developed for computer vision. TTA methods adapt pre-trained models to new data distributions during deployment. Test-Time Training (TTT) achieves this through self-supervised tasks, allowing the model to improve its confidence without human labels.

Recently, TTT4Rec brought this idea to recommender systems. By allowing the model to adapt its internal hidden state for each new user interaction, it achieves significantly better results without the computational burden of full retraining.

In this project, we build upon TTT4Rec to examine if we could improve model efficiency. While TTT4Rec adapts the model's state to handle context changes, we propose adding ETA (Efficient Test-Time Adaptation) to filter out unreliable and redundant test samples before model adaptation, reducing computational overhead and potential overfitting to noise.

2. Related Work

Test-Time Training (TTT) in Recommendation

Traditional sequential recommender systems like SASRec and BERT4Rec use static parameters after training, making them vulnerable to shifts in user behavior over time. TTT4Rec addresses this with a dual-loop architecture where the model's hidden state becomes a learnable parameter. By optimizing a self-supervised

reconstruction loss on incoming test sequences, TTT4Rec continuously adapts during inference. However, TTT4Rec processes all test data, which increases computational cost and may incorporate noisy samples.

Efficient Test-Time Adaptation (ETA)

In computer vision, the ETA framework uses sample filtering to improve TTT efficiency. It filters out unreliable (high entropy) or redundant (high similarity) samples to prevent noisy gradients and reduce computation. While effective for image classification, ETA assumes a dense output space. This assumption breaks down in recommender systems, which operate in sparse, high-dimensional action spaces. Our work adapts ETA to address this challenge.

3. Methodology

3.1 Base Architecture: TTT4Rec

We use the TTT4Rec framework with a Transformer backbone as our base model. The core component is the TTT Layer, which updates the model's hidden states via gradient descent on self-supervised tasks before generating the final prediction.

3.2 Proposed Module: ETA

To reduce the computational cost of the TTT layer, we insert an ETA module into the inference pipeline. This module acts as a gate, evaluating each test sample's forward-pass features to decide whether to perform the backward-pass update. We use two filtering criteria, following the original ETA approach:

1. Reliability (Entropy): High-entropy predictions indicate uncertainty, making their gradients potentially biased and unreliable. We use an entropy-based weighting scheme that identifies reliable samples and emphasizes their contributions. The entropy-based weight is:

$$S^{ent}(x) = \frac{1}{exp[E(x;\Theta) - E_0]} \cdot \mathbb{I}_{\{E(x;\Theta)<E_0\}}(x)$$

where E_0 is a pre-defined entropy threshold.

2. Redundancy (Diversity): To avoid training on redundant samples, we maintain a moving average of the model's output features. If the cosine similarity between the current sample and the moving average exceeds a threshold, the sample is considered redundant and the update is skipped. For a test sample x at iteration $t > 1$, we compute the cosine similarity between its prediction $f_\Theta(x)$ and the moving average m^{t-1}:

$$S^{div}(x) = \mathbb{I}_{\{cos(f_\Theta(x),m^{t-1})<\epsilon\}}(x)$$

where ϵ is a pre-defined cosine similarity threshold.

3.3 Adaptation to Context Change: Active Sample Filtering (ETA-Rec)

While ETA established its status in computer vision, our empirical analysis shows that its core assumptions might be incompatible for sequential recommendation. We hypothesized that the key issue is the fundamental difference in output spaces:

In dense classification tasks (e.g., CIFAR-10, ImageNet), the label space is relatively small ($|\mathcal{Y}| \approx 10^3$), and confident predictions show a peaked distribution where $p_{max} \to 1$, resulting in near-zero entropy.

In contrast, recommender systems operate in massive action spaces ($|\mathcal{V}| \approx 10^4$ to 10^7). For the Amazon Video Games dataset, the vocabulary size is $|\mathcal{V}| = 10,673$ with $> 99.9\%$ sparsity. This extreme sparsity dilutes the probability distribution and creates noise. Additionally, recommender systems aim to provide a ranked list of possible next items rather than a single label, therefore even confident predictions would have their probability mass distributed across multiple plausible items.

Proposed Solution: Top-K Filtering

We propose ETA-Rec, an adaptation of ETA for sequential recommendation using Top-K filtering. Instead of computing entropy over all items, we calculate it only on the Top-K most probable items. A sample is discarded if its Top-K entropy exceeds a threshold, indicating uncertainty that could destabilize the model. Similarly, cosine similarity is computed on top-K items to focus the comparison on the most relevant predictions.

We set the entropy threshold E_0 as the median entropy of the first test batch, calibrating to the actual operating range rather than using a theoretical maximum. The cosine similarity threshold ϵ_0 remains consistent with the original EATA paper.

4. Experimental Setup

We evaluated our approach on the Amazon-Video-Games dataset, a well-established benchmark with high sparsity and sequential user behaviors. We compared three model configurations:

- **Static Baselines:** Standard sequential models (SASRec, BERT4Rec).
- **Adaptive Baseline:** The original TTT4Rec model (adapting on 100% of samples).
- **ETA-Rec (Ours):** TTT4Rec integrated with our Top-K Active Filtering module.

We measure performance using Hit Ratio@10 (HR@10) and NDCG@10 for accuracy. For efficiency, we introduce the Filtering Rate, defined as the percentage of test samples where the backward propagation step was successfully skipped.

All hyperparameters except E_0 are kept consistent with the original papers to ensure fair comparison. Further hyperparameter optimization is a limitation that should be addressed in future work.

5. Results & Analysis

5.1 Performance Comparison

Table 1 and Figure B in the appendix show the performance comparison between the adaptive baseline and our ETA implementations. Compared to the baseline TTT4Rec (HR@10: 0.0872), naively applying the standard entropy filter from computer vision caused a severe performance drop (HR@10: 0.0237). However, our adapted ETA-Rec ($K = 50$, HR@10: 0.0868) restores performance to baseline levels.

Table 1: Recommendation Performance

Model	Hit@10	NDCG@10	Filter Rate (Test)
Best Static Baseline			
SASRec	0.0841	0.0401	-
Adaptive Baselines			
TTT4Rec (Paper)	0.0879	0.0425	-
TTT4Rec (Replication)	0.0872	0.0421	-
Our Method			
TTT4Rec + ETA (Naive)	0.0237	0.0116	-
ETA-Rec ($K = 10$)	0.0865	0.0416	8.08%

Model	Hit@10	NDCG@10	Filter Rate (Test)
ETA-Rec ($K = 50$)	0.0868	0.0418	13.73%

5.2 Efficiency Analysis

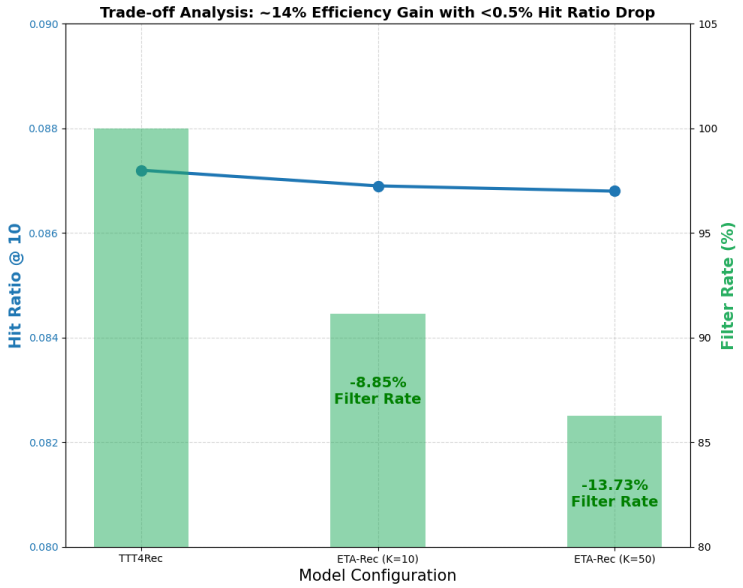

Figure 1: Trade-off Analysis

Figure 1 shows that our ETA-Rec ($K = 50$) achieved a 13.73% reduction in backward passes during the test phase while maintaining statistically insignificant difference in accuracy (0.5% reduction). In large recommender system datasets, this represents a considerable reduction in gradient computations.

6. Discussion & Limitations

While our method successfully reduces gradient computations, we need to address the project limitations. Computing Top-K entropy adds overhead to the forward pass, therefore real computational gains depend on whether the saved backward pass exceeds this overhead. Additionally, due to scope and time constraints, we did not implement the Fisher Regularization component from the original EATA framework. Without this regularization, this model lacks the ability to prevent catastrophic forgetting of long-term user preferences. Since we evaluated the model performance on one single dataset, future work is recommended to evaluate this method on additional datasets for validation. Finally, hyperparameter tuning was not conducted rigorously, and performance could improve further upon optimization on the training set to maximize the tradeoff stability.

7. Conclusion

This project demonstrates that efficient TTA methods can be transferred from computer vision to recommender systems to improve performance, but successful implementation requires domain-specific challenges like data sparsity to be addressed. By implementing a Top-K ETA-Rec, we reduced the computational burden of Test-Time Training by 13.73% without compromising recommendation accuracy. Future work should focus on integrating anti-forgetting regularization to ensure long-term stability in continuous adaptation scenarios.

References

1. Yang, Z., Wang, Y., & Ge, Y. (2024). TTT4Rec: A Test-Time Training Approach for Rapid Adaption in Sequential Recommendation. arXiv preprint arXiv:2409.19142.

2. Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., & Tan, M. (2022). Efficient test-time model adaptation without forgetting. International Conference on Machine Learning (pp. 16888-16905). PMLR.

3. Zhang, C., Zhang, X., Shi, T., Xu, J., & Wen, J. R. (2025). Test-Time Alignment for Tracking User Interest Shifts in Sequential Recommendation. arXiv preprint arXiv:2504.01489.

Appendix

A. Full Performance Table

Table A1: Complete Recommendation Performance

Model	Hit@10	Hit@50	NDCG@10	NDCG@50	Filter Rate (Test)	Filter Rate (Valid)
Static Baselines						
GRU4Rec	0.0495	0.1428	0.0246	0.0446	-	-
SASRec	0.0841	0.2107	0.0401	0.0677	-	-
BERT4Rec	0.0259	0.0845	0.0126	0.0251	-	-
Mamba4Rec	0.0711	0.1732	0.0394	0.0615	-	-
Adaptive Baselines						
TTT4Rec (Paper)	0.0879	0.2178	0.0425	0.0707	-	-
TTT4Rec (Replication)	0.0872	0.2165	0.0421	0.0701	-	-
Our Method						
TTT4Rec + ETA (Naive)	0.0237	0.0746	0.0116	0.0225	-	-
ETARec ($K = 10$)	0.0865	0.2157	0.0416	0.0697	8.08%	26.21%
ETARec ($K = 50$)	0.0868	0.2166	0.0418	0.0700	13.73%	32.92%

B. Additional Figures

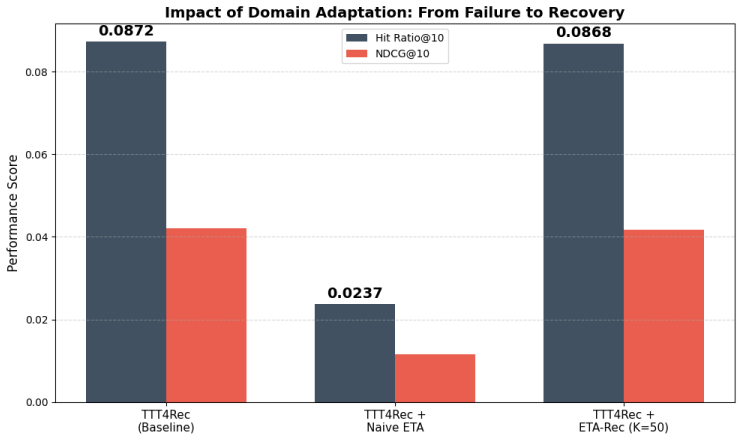

Figure 2: ETA Domain Adaptation