Base Items Overfit, New Items Underfit: The Hidden Cost of Joint Training in Incremental Adaptation

Rishabh Agrawal

Amazon

NeurIPS 2026

Models keep getting new entries

Sequential recommendation

New products appear every day; each needs a new row in the item embedding table.
Models and data: SASRec on Taobao (86.8M interactions) and MerRec (305.9M).

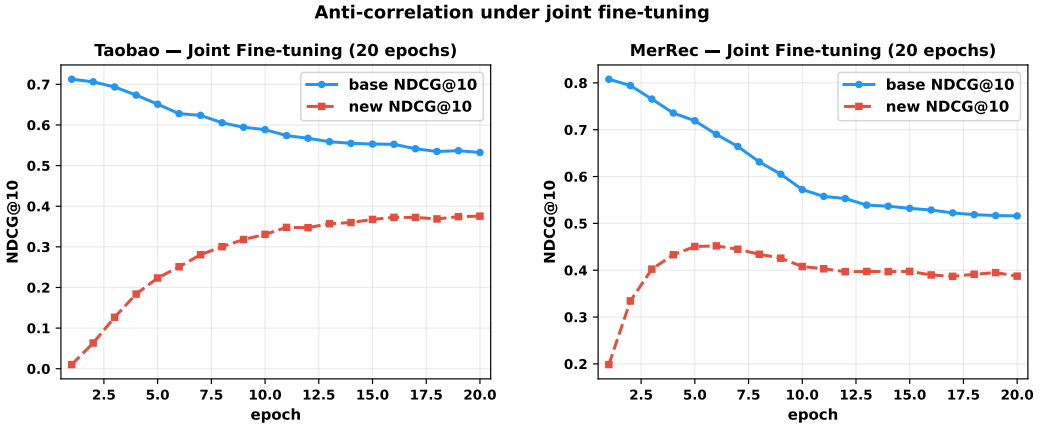
LLM vocabulary expansion

A language model adapted to a new domain gets new tokens; each needs a new row in the token embedding table.
Models and data: GPT-2 (124M) and Pythia (160M to 12B) on BillSum and PubMed, 2000 new tokens.

In both, the standard recipe adds the new rows, then **jointly fine-tunes** the old (base) rows, the new rows and the shared encoder.

Question: what does joint training do to the *old* entries while the new ones learn?

The hidden cost: base gets worse while new still improves



Joint fine-tuning, first incremental window. On Taobao, base NDCG@10 **drops 25%** over 20 epochs while new-item NDCG@10 rises from 0.01 to 0.38.

No single epoch is good for both populations.

Why early stopping cannot fix it

Stop early (standard validation early stopping)

On Taobao, joint fine-tuning stops at **epoch 1**, with new-item NDCG@10 = **0.010**.

On Taobao, 90% of validation targets are base items, so the score peaks before new items learn anything.

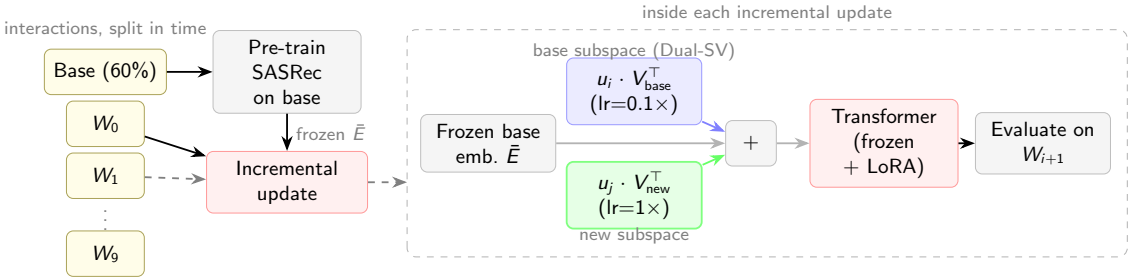
Train longer (20 epochs)

New items reach 0.38, but base-only NDCG@10 falls by **0.18** (Taobao) and **0.29** (MerRec).

LLMs show the same pattern: base-token perplexity rises after a few epochs.

Either new items underfit, or base items overfit.

Experimental setup (sequential recommendation)

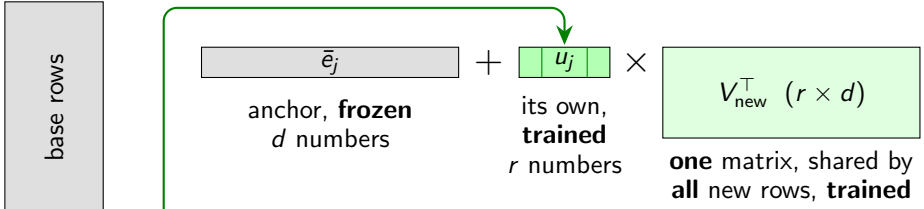


Freeze-SV	base rows frozen from the start; new rows low-rank
Freeze1-SV	one warmup epoch (everything trains), then base rows frozen
Dual-SV	base rows also get their own low-rank correction (V_{base}, blue), at 0.1× the learning rate

Method: what each embedding row is made of

embedding table

one new row: $e_j = \bar{e}_j + u_j V_{\text{new}}^{\top}$



- The shared V_{new} pools signal from every new item, so rare items borrow strength.
- Base rows: frozen in Freeze-SV, and after one warmup epoch in Freeze1-SV. In Dual-SV each base row gets its own $u_i V_{\text{base}}^{\top}$, so base and new rows never share embedding parameters; only the LoRA encoder is shared.
- Sizes: SASRec d=150, r=128; Pythia-1.4B d=2048, r=1024.

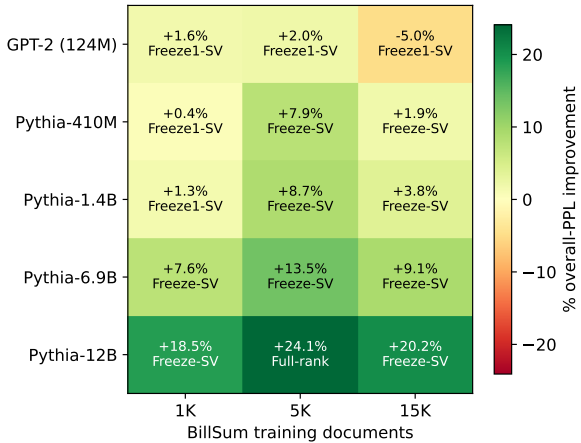
Sequential recommendation: with and without early stopping

	Taobao		MerRec	
Method	new	base	new	base
With early stopping, mean over 10 windows				
Joint fine-tuning	0.087	0.705	0.529	0.709
EWC (λ tuned)	0.363	0.680	0.571	0.639
ADER	0.095	0.695	0.511	0.634
Dual-SV (ours)	**0.514**	0.690	**0.667**	**0.746**
No early stopping, first window, epoch 20				
Joint fine-tuning	0.378	0.535	0.384	0.500
EWC (λ tuned)	0.357	0.693	0.498	0.663
ADER	0.376	0.614	0.478	0.625
Dual-SV (ours)	**0.442**	0.665	0.466	**0.677**

At 5 and 10 epochs, Dual-SV has the most new-item quality on both datasets (Taobao 0.486, 0.461; MerRec 0.499, 0.475).

NDCG@10. Early stopping: validation overall NDCG@10, patience 5. Dual-SV: V_{base} rank 32.

LLM vocabulary expansion



% overall-PPL gain of the best frozen variant over joint training, BillSum.

- Freezing the base wins on overall perplexity in **14 of 15** BillSum and **13 of 15** PubMed cells, from GPT-2 to Pythia-12B.
- Exceptions: GPT-2 at the larger data sizes, and Pythia-160M: a capacity effect. From 410M up, freezing wins in every cell.
- The gain grows with scale: at 12B on BillSum-15K, Dual-SV **3.27** vs joint **4.06** PPL.

Ablations

Shared low-rank delta (LoRA on the embedding table): one low-rank update for old and new rows together, nothing frozen.
It loses base quality: base NDCG@10 at epoch 20 is 0.597 vs. **0.665** for Dual-SV (Taobao), 0.560 vs. **0.677** (MerRec).

Full-rank new rows (base frozen): each new token trains its own full embedding vector instead of $\bar{e}_j + u_j V_{\text{new}}^\top$.
Our low-rank rows give equal or lower perplexity in all 9 BillSum settings (3 models $\times$ 3 data sizes), e.g. Pythia-1.4B: **4.068** vs. 4.094.

Full-catalog ranking: each target ranked against every item (0.9M Taobao, 3.9M MerRec) instead of 100 random items.
New-item NDCG@10 at epoch 20 is **0.156** for Dual-SV vs. 0.010 for joint fine-tuning (Taobao), **0.136** vs. 0.053 (MerRec).

Recommendation results: first window, no early stopping; Dual-SV: V_{base} rank 32.

Takeaways

- We show that joint training has a hidden cost: **base items overfit while new items still underfit**, and early stopping cannot pick a good epoch.
- We give each population its own low-rank subspace (Freeze-SV, Freeze1-SV, Dual-SV), a change to how the embedding table is parameterized.
- In a standard pipeline it pays off: under the same early stopping, **6**$\times$ more new-item quality on Taobao at near-equal base quality, and better on both on MerRec; on LLMs, freezing wins in every cell from 410M to 12B (**19%** lower perplexity at 12B).

Thank you