

Peer Review

Review of: "Metadata Conditioning Accelerates Language Model Pre-training"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The paper proposes MeCo (Metadata Conditioning then Cooldown), a two-stage pre-training recipe. For the first 90% of training, each document is prefixed with its source domain in the template URL: `en.wikipedia.org\n\n[document]`, with the loss computed only over document tokens; for the last 10%, the model sees plain documents, so that it works normally at inference without any metadata (§2). The central claim is that this small change speeds up pre-training: a 1.6B Llama-style model trained on 160B DCLM tokens with MeCo reaches the same 10-task average as a standard model trained on 240B tokens (Table 1, column Avg., rows MeCo and + 80B tokens, both 56.7), which the authors describe as 33% less data. They report gains across four model sizes (Table 17) and three corpora (Table 18).

The second claim is about steering. Because the model has learned to associate domains with styles of text, prepending a chosen URL at inference (real or invented) can raise task accuracy (Tables 2 and 3) and lower toxicity when the prompt starts with `en.wikipedia.org` (Figure 6). Ablations in §5 argue that what matters is grouping documents by source rather than the meaning of the URL: hashed URLs give the same average as real ones (Table 5, row Hashed URLs, Avg. 56.7).

I enjoyed reading this. The idea is simple enough to add to any pre-training pipeline in an afternoon, the comparison is run at a scale where the result actually matters, and the authors release models, data, and code (footnote 1). The hashed-URL ablation is the kind of experiment I wish more papers ran because it changes how one should think about why the method works.

2. Whether the conclusion holds on the data

Several things are done well and deserve to be said first. The baselines are fair: the standard model uses the same improvements the authors apply to MeCo (no cross-document attention, sequences starting at document boundaries, §2), and Table 12 shows the effect of the first of these for both arms. The two reference points in Table 1 (data selection and 240B tokens) are honestly labelled as "references instead of fair comparisons." The authors also warn in footnote 3 that they do not claim MeCo beats data selection in general. I checked all deltas printed under Table 1, Table 3, and Table 4 against the rows above them, and each one is correct. I also recomputed the Avg. column of Table 1: the ten task scores of Standard sum to 557.1 and those of MeCo to 567.1, giving 55.71 and 56.71, which matches the printed 55.7 and 56.7.

Where I think the text says a little more than the numbers carry:

The average gain is concentrated in two tasks. The total difference of 10.0 points over ten tasks (567.1 – 557.1) comes mostly from OBQA (Table 1, delta row, +5.2) and TruthfulQA (+3.3). Together, that is 8.5 of the 10.0. The remaining eight tasks add 1.5 points, or about 0.19 per task, and three of them go down (CSQA –1.0, PIQA –0.9, SIQA –1.6). The same pattern appears on C4 (Table 18, block C4): TruQA moves from 32.0 to 39.5, a gain of 7.5, while the sum of all ten deltas is 11.0 (avg. 51.6 vs 50.5), so without TruQA, the other nine tasks gain 3.5 in total. I do not think this undermines the paper. It does mean that the sentence in §3.4 that MeCo gives "consistent and significant gains ... both on the average accuracies and individual tasks" is stronger than Table 18 shows, since on C4, CSQA, HSwag, and WG all drop.

Variance is estimated for the cooldown, not for a full run. §B.2 and Table 13 are a welcome attempt at quantifying noise. But the three runs share the same first 90% of training and differ only in the data for the last 10%, so the Avg. standard deviation of 0.1 (Table 13, row Std., column Avg.) is a lower bound on run-to-run variance. Initialization, data order, and the main-stage data are all held fixed. I would state this plainly in §3.1 and in the Limitations, instead of "the variance of our experiments should be low".

Evaluation sample size. §3.1 says 1,000 examples are sampled per task. For a single task with accuracy near 50%, the binomial standard error is $\sqrt{0.5 \times 0.5 / 1000} \approx 0.016$, i.e., about 1.6 points. Per-task deltas of around one point in Tables 1, 17, and 18 are inside that range; the OBQA +5.2 is well outside it. The paired design helps, but I could not find a paired test or bootstrap interval anywhere in the paper. The claim that 1,000 examples give "the same reliable results as full evaluation" is not supported by a number in the text.

Scale trend. From Table 17, the Avg. gains are: 600M, $51.7 - 51.5 = 0.2$; 1.6B, 1.0; 3B, $59.8 - 59.2 = 0.6$; 8B, $58.4 - 57.8 = 0.6$. The text in §3.3 correctly calls this "somewhat noisy" and "qualitative," which I appreciate. Still, the 1.6B model has the largest gain, so "greater improvements for larger models" is not what the four numbers show. The 600M gain of 0.2 is also only twice the Table 13 standard deviation, which (as noted) is itself a lower bound.

The 33% figure. It rests on one comparison: MeCo at 160B against a single standard run at 240B, both 56.7. Table 16 gives a second, independent view. The standard 160B checkpoint scores 55.7; MeCo reaches 55.4 at 112B and 56.1 at 128B. On that curve, MeCo matches the 160B baseline somewhere between 112B and 128B tokens, i.e., a saving of 20% to 30%. That is still a very good result. I would present the range next to the single 33% number and keep "up to" in the Conclusion.

Steering. Table 2 is convincing on its own terms: conditional inference changes the standard model by 0.1 and MeCo by 0.5, and $57.2 - 55.7 = 1.5$ matches the text. The toxicity result in Figure 6 is reported only as a chart, and the text gives no numbers or intervals. With 4,096 samples per condition, an interval would be cheap to add.

3. Reproducibility

This is well above the norm for pre-training papers. Released artifacts (footnote 1), optimizer and schedule (Table 6), architectures (Table 7), GPU hours (Table 9; the 1,536 hours for the main 1.6B model agree with "roughly 2 days on 32 H100 GPUs" in §A.5, since $1536 / 32 = 48$ hours), the topic-generation prompt with greedy decoding (Table 10), every inference URL (Table 11), and the full per-task results (Tables 16 to 18, 20, and 21) are all present. Toxicity sampling parameters are stated ($T = 0.7$, top-p = 0.9, 4,096 sequences of 10 to 128 tokens, Detoxify unbiased model). The topic model is named precisely as Llama-3.1-8B-Instruct.

What I could not find in the text:

- The training sequence length (context window). Table 6 gives batch size in tokens (4M) but not how many tokens per sequence, which matters for the packing rule in §2.
- Random seeds for training and for drawing the 1,000 evaluation examples per task, and the OLMES and Detoxify versions used.
- Training framework and numeric precision.
- For C4 and RefinedWeb, whether the cooldown data is disjoint from the main-stage data; §A.3 guarantees this only "for all our DCLM experiments".

- How the intermediate checkpoints of Figure 2 and Table 16 are produced for the standard model. MeCo checkpoints get a 16B cooldown on top; it is not said whether the standard checkpoints are taken mid-schedule at a high learning rate or are also annealed. The fairness of the curve depends on this.

As rendered on Qeios, Table 19 (full results for conditional inference, referred to in the caption of Table 2 and in Appendix D) does not appear in the text, and Figures 1 to 6 carry no numeric values. I could not verify the per-task conditional-inference numbers or the toxicity scores for this reason.

4. *What specifically to fix*

1. **§1, contribution 2 (and the matching example in the Abstract).** The 6% zero-shot CommonsenseQA gain is attributed to `factquizmaster.com`, described as "not a real URL". In Table 3, the +6.2 on CSQA comes from `www.factmonster.com`, which §4.1 calls a real trivia website, while `factquizmaster.com` is the custom URL for OpenBookQA in Table 11 (CSQA uses `www.quizsmart.com`). Please align the example with the table.
2. Restore Table 19 on the Qeios version.
3. **§5.2, "Model-generated topics".** The text asks for a "two-word or three-word topic", but the prompt in Table 10 says "within 4 words". One of the two should change, and a short length distribution of the generated topics would settle it.
4. **§B.2 / Table 13.** Say explicitly that the three runs share the first 90% of training, so the reported ± 0.1 excludes initialization and main-stage data variance. If compute allows even one extra 600M run from a different seed, it would give a much better anchor than the current reference point.
5. Add paired bootstrap confidence intervals (resampling the 1,000 evaluation items) for the Avg. differences in Tables 1, 17, and 18. This is cheap because the predictions already exist, and it would let a reader see which per-task deltas are distinguishable from noise.
6. **§3.4, last sentence of the first paragraph.** Soften "consistent and significant gains ... on individual tasks"; Table 18 shows drops on C4 for CSQA (55.8 $\rightarrow$ 54.3), HSwag (64.9 $\rightarrow$ 63.6), and WG (60.0 $\rightarrow$ 58.9). It would also help to report the Avg. with TruthfulQA excluded, since that one task drives much of the C4 gain.
7. **§3.3.** Replace "greater improvements for larger models" with a description that fits Table 17, where the 1.6B gain (1.0) exceeds both the 3B and 8B gains (0.6 each). The 8B model also differs in learning rate, weight decay, token count, and cooldown fraction (12.5%, §A.3), so it cannot sit on the same trend line without a caveat.

8. §3.2 / Figure 2. State how the standard intermediate checkpoints are obtained and whether their learning rate was annealed. Consider reporting the token saving read off Table 16 (between 112B and 128B MeCo tokens to reach the standard 160B score) beside the 33% headline.
9. Figure 6. Put the mean toxicity values, with confidence intervals over the 4,096 samples, into a table. A single sentence on how many samples crossed a toxicity threshold of 0.5 would make "several-fold" in §1 checkable.
10. The Abstract says MeCo "adds no computational overhead"; §1 and §7 say "negligible". The URL prefix occupies context positions that carry no loss, so "negligible" is the accurate word. Reporting the average prefix length in tokens for DCLM would close the question.
11. Table 4. The four columns shown omit CSQA, where 100% URL collapses from 64.8 to 37.2 (Table 20). That is the most striking evidence for why cooldown is needed; I would include it in the main-text table.
12. §4.1. The note that URLs were chosen without trial and error is good practice. To show that the gain is not tied to particular strings, I would add two or three paraphrased URLs per task and report the spread.
13. §B.1 explains the small Table 12 gap for MeCo (56.7 vs 56.6) by saying URLs help the model "learn the noisy cross-document attention". That is a hypothesis; please phrase it as one.

5. Recommendation

Minor revisions. The method is simple and cheap. It is tested at a serious scale with fair baselines, and the artefacts are released. The main conclusion, that metadata conditioning with a short cooldown improves data efficiency, is supported by Tables 1, 16, 17, and 18. The items above are about calibrating the wording to the numbers (concentration of gains in OBQA and TruthfulQA, the narrow variance estimate, the scale trend), fixing one example that contradicts Table 3, and restoring Table 19 on this platform. None of them requires new pre-training runs, although an additional seed would strengthen the paper considerably.

Declarations

Potential competing interests: No potential competing interests to declare.