Better Multi-Token Drafters by Cancelling the Verifier's Error

Ben Zhao
Iluvatar Labs Inc.
ben@iluvatarlabs.com

Marvin
Iluvatar Labs Inc.
marvinsci.com

1 October 2026

Abstract

A language model with multi-token prediction (MTP) heads has a built-in drafter for speculative decoding: the heads propose tokens beyond the next one, and the main model, acting as verifier, accepts or rejects them. With draft distribution d, the verifier's distribution v for the same token, and target y, a head's cross-entropy gradient through the output layer W is $W^\top(d-y) = W^\top(d-v) + W^\top(v-y)$: a gap term between draft and verifier, plus the verifier's error, which does not involve the head. We cancel the error term in the heads and the trunk by adding a detached-minus-attached copy of each head's state to the verifier's input; no loss value changes. We train Llama-style models of 6M to 59M parameters from scratch with DeepSeek-V3's MTP setting, one MTP module at loss weight 0.3. At 10M parameters and above, error-cancelled MTP lowers the total variation (TV) between draft and verifier by 0.8 to 1.0 percentage points and raises the acceptance rate of the drafted token by 1.1 to 1.9 percentage points, within 0.4% of standard MTP's training speed at the same memory; at 10M its drafts do not differ detectably from those of a KL loss. At 6M the TV difference is not detected. Compared with standard MTP, no next-token cost is detected at any size, and at 59M next-token loss is 0.0016 ± 0.0015 nats lower. Three analyses at 10M show that drafting depends on the trunk learning from the MTP loss while the cancellation's gain does not, that training the shared output layer on next-token loss alone brings no detectable next-token gain at this weight and makes drafts worse, and that a short round of extra head training lets standard MTP catch up, while error-cancelled MTP stays closer to the verifier when given the same.

1 Introduction

Speculative decoding speeds up generation by letting a cheap drafter propose tokens that the full model checks in a single forward pass [1, 2]. MTP heads [3–5] make a convenient drafter: they share the model's trunk, are small, and are learned alongside the next-token objective.

Standard MTP trains each head with cross-entropy on the true future token; written around the verifier's prediction for that token, the head's gradient is the sum of a gap term, which vanishes when the draft matches the verifier, and the verifier's error on that token. The error term does not depend on the head, but the head still receives it and passes it to the trunk. For drafting, the gap term is the one that matters: speculative sampling accepts a drafted token with probability one minus the total variation between draft and verifier at that position [1], so what a drafter should minimise is its distance to the verifier, not to the data [6, 7]. We cancel the error term by adding a quantity that is zero in value, so only gradients change and no loss is added. A KL loss between draft and verifier, as used to distill drafters [6, 8] and to train multi-token or latent predictions [9–11], yields the same head and trunk gradients, but its target at every head position is the verifier's whole distribution rather than one token; in our timing its throughput is 11 to 16% lower than standard MTP's, while the cancellation runs within 0.4% of standard MTP's speed.

We test the cancellation at DeepSeek-V3's MTP setting, a single MTP module at weight 0.3 [3], in models small enough to train with paired seeds at four sizes. Our contributions:

- **The change.** The split of the head gradient into a gap term and a verifier-error term, and an exact implementation of the cancellation: its head and trunk gradients match a KL loss to 3.9×10^{-18} in float64, and it trains within 0.4% of standard MTP's speed.
- **Drafts at small scale.** At 10M and above, error-cancelled MTP's drafts are 0.8 to 1.0 TV points closer to the verifier and accepted 1.1 to 1.9 points more often, and TV point estimates are lower on all five other held-out corpora; at 10M they do not differ detectably from a KL loss's drafts. At 6M the TV difference is not detected.
- **Next-token loss.** No next-token cost compared with standard MTP is detected at any size, and at 59M error-cancelled MTP is 0.0016 ± 0.0015 nats lower. The head's own cross-entropy is higher instead, by 0.086 nats at 6M falling to 0.007 at 59M.
- **Three analyses** at 10M: drafting depends on the trunk learning from the MTP loss, but the cancellation's gain does not: cancelling the error in the head's own gradient is enough; training the shared output layer on next-token loss alone brings no detectable next-token gain at this weight and costs draft quality; and extra head training after pretraining lets standard MTP catch up with error-cancelled MTP as trained, while error-cancelled MTP stays closer to the verifier when both receive it.

2 Method

Head k's hidden state h_k at position i predicts token x_{i+k+1} through the shared output layer W, giving the draft $d = \mathrm{softmax}(Wh_k)$. The verifier for that same token is v, the trunk's next-token distribution at position $i + k$. For one-hot target y:

$$\underbrace{W^\top(d-y)}_{\text{head CE}} = \underbrace{W^\top(d-v)}_{\text{gap}} + \underbrace{W^\top(v-y)}_{\text{verifier error}} \tag{1}$$

The gap term is the gradient of $\mathrm{KL}(v \,\|\, d)$ with v held fixed. The verifier-error term does not involve the head.

To cancel it, we add each head's state, detached minus attached, to the trunk state at position $i + k$, before W:

```
# x: (batch, T) tokens; s: trunk states before W, (batch, T-1, d); s[:, i] predicts x[:, i+1]
# head_states[k-1]: (batch, T-k-1, d); head_states[k-1][:, i] predicts x[:, i+k+1]
for k, h in enumerate(head_states, 1):
    # head loss's per-position weight over the next-token loss's
    c = (lam / D) * (T - 1) / (T - k - 1)
    s = s + F.pad(c * (h.detach() - h), (0, 0, k, 0))   # zero in value; moves position i to i+k
```

The added quantity is zero, so every forward value and loss is unchanged. In the backward pass, the next-token loss sends $-c \cdot W^\top(v - y)$ into h_k, cancelling the error term; the heads and trunk keep the gap term, and W's gradient is unchanged from standard MTP: the heads' full cross-entropy gradient. The factor c is the head loss's per-position weight relative to the next-token loss's: the loss weight λ over the number of heads D, times $(T-1)/(T-k-1)$, the next-token loss's position count over the head loss's. Without it, a head would receive about $(\lambda/D) \cdot \big(\text{gap} - (D/\lambda - 1) \cdot \text{error}\big)$, which at $\lambda = 0.3$ and one head reverses the error term and enlarges it 2.3 times instead of removing it.

In float64, at one head and $\lambda = 0.3$, the head and trunk gradients match a KL loss to 3.9×10^{-18}, W's gradient matches standard MTP exactly, and the losses are bitwise equal.

3 Setup

Models and data. Llama-style transformers [12] with an untied output layer and an 8,192-token BPE vocabulary, trained from scratch on FineWeb-Edu [13] with batches of 64×512 tokens and about 29 tokens per parameter at every size.

Size	Width, layers, attention heads, MLP width	Parameters (without the MTP head)	Tokens	Seeds
6M	192, 6, 4, 512	5.8M	170M	5
10M	256, 8, 4, 683	10.5M	300M	5
28M	384, 12, 6, 1024	27.5M	800M	3
59M	512, 16, 8, 1365	58.7M	1.7B	3

MTP. One MTP module as in DeepSeek-V3 [3]: a transformer layer with its own input projection and norms that reads the shared embedding and output layer. The training loss has DeepSeek-V3's form, next-token cross-entropy plus $(\lambda/D)\sum_k$ of the heads' cross-entropy [3, eq. 25], each averaged over its predicted positions, with one module ($D = 1$) and $\lambda = 0.3$, the weight DeepSeek-V3 uses for the first 10T of its 14.8T training tokens [3, §4.2]; we keep it constant. AdamW [14] with peak learning rate 6×10^{-4}, linear warm-up over about the first 2% of steps and cosine decay to 6×10^{-5}, betas 0.9/0.95, weight decay 0.1 on weight matrices, and gradient clipping at 1.0, untuned; all training in fp32.

Arms. Every arm within a seed shares initialisation and data order.

Arm	Objective
Next-token only	next-token CE, no head
Standard MTP	next-token CE + $\lambda \cdot$ head CE
Error-cancelled MTP	as standard MTP, with the verifier-error term cancelled in head and trunk
KL loss	next-token CE + $\lambda \cdot \mathrm{KL}(v \,\|\, d)$, v held fixed

Error-cancelled MTP and the KL loss give the head and trunk the same gradient; only W's gradient differs: error-cancelled MTP trains W on the head's cross-entropy, the KL loss on its KL.

Metrics. Two primary metrics on 8M held-out FineWeb-Edu tokens: total variation (TV) between the head's draft and the verifier, and next-token cross-entropy. Secondary metrics on the same tokens: forward KL $\mathrm{KL}(v \,\|\, d)$, MTP cross-entropy, and the verifier's entropy. Drafts are also measured on five held-out corpora chosen by established practice: the WikiText-2 test set [15], PG-19 test books [16], and CodeSearchNet Python test functions [17], the Wikipedia, books, and code domains used to evaluate MTP drafting by Gloeckle et al. [4, Table S2]; and CNN/DailyMail test articles [18] and GSM8K test problems with their worked answers [19], the news and grade-school-math sources of speculative-decoding evaluations [1, 6, 20]. TV uses 1M tokens of each, or the whole test set for WikiText-2 (0.36M) and GSM8K (0.25M); next-token cross-entropy is also reported on WikiText-2. On all six corpora: the acceptance rate of the drafted token under speculative sampling at temperature 1, from 256 prompts per corpus, each 256 tokens long and continued for 256 tokens. Three further corpora, arXiv papers [21], Hendrycks MATH [22], and a Project Gutenberg shard [23], measured the same way, are reported in Appendix A.1. TV and acceptance rate are reported in points ($\times 100$).

Statistics. Every comparison is paired by seed and reported as mean $\pm$ 95% t-interval half-width. Only the two primary metrics get a verdict (lower, higher, or not detected); a secondary metric is called higher or lower only where its interval excludes zero. We claim a size trend only when an effect moves across sizes beyond its intervals.

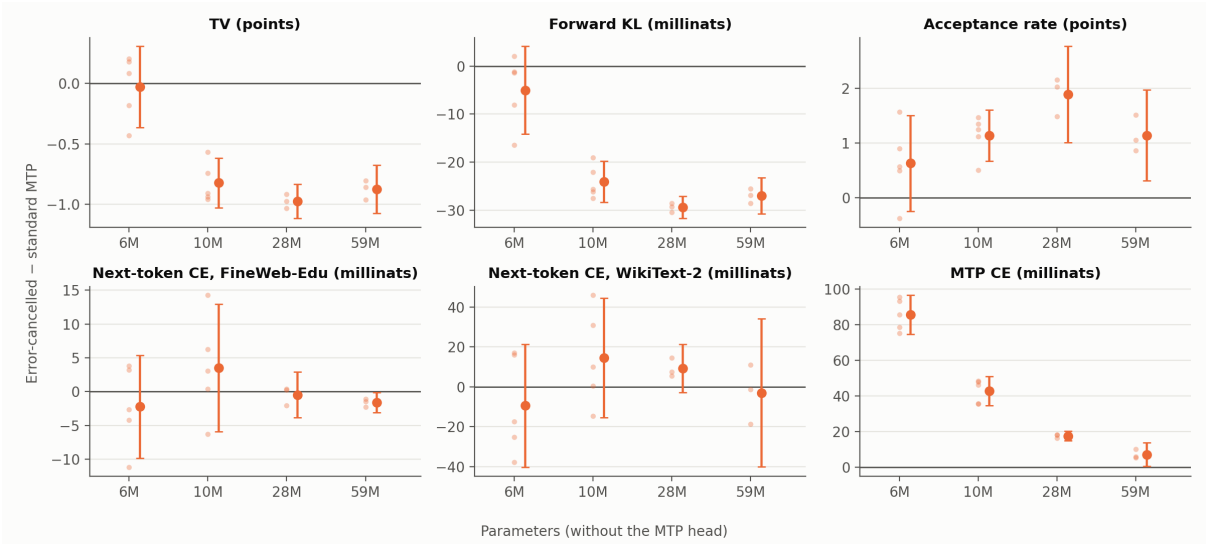


Figure 1. Error-cancelled minus standard MTP at each size, on FineWeb-Edu except where a panel names WikiText-2. Large points are means over seeds with 95% intervals; faint points are seeds.

Experiments. 98 runs and 78 GPU-hours of training: 16 next-token-only runs, 57 MTP runs, and 25 head-training jobs.

Experiment	Arms	Sizes × seeds
Main comparison	next-token only, standard MTP, error-cancelled MTP	6M, 10M × 5; 28M, 59M × 3
Against a KL loss	KL loss	10M × 5
Trunk gradient cut	standard and error-cancelled MTP, MTP gradient stopped at the head's input	10M × 5
W trained by next-token loss only	standard and error-cancelled MTP	10M × 5
Head training after pretraining	fresh KL-trained heads on frozen next-token-only models; existing heads trained further with KL or TV on frozen standard and error-cancelled models	10M × 5

4 Results

4.1 Drafts

Error-cancelled − standard MTP, FineWeb-Edu	6M	10M	28M	59M
TV, points	-0.03 ± 0.34	-0.82 ± 0.21	-0.97 ± 0.14	-0.87 ± 0.20
Verdict	not detected	lower	lower	lower
Forward KL, nats	-0.005 ± 0.009	-0.024 ± 0.004	-0.029 ± 0.002	-0.027 ± 0.004
Acceptance rate, points	$+0.6 \pm 0.9$	$+1.1 \pm 0.5$	$+1.9 \pm 0.9$	$+1.1 \pm 0.8$

At 10M and above, error-cancelled MTP's drafts are closer to the verifier and accepted more often (Figure 1), by about the same amount at 10M, 28M, and 59M. At 6M the TV difference on FineWeb-Edu is not detected. On the other five corpora (Appendix A.1), TV is lower on PG-19 (-0.54 ± 0.19 points) and its interval includes zero on the rest, with point estimates above zero on code and GSM8K; acceptance is higher on WikiText-2 and CNN/DailyMail (intervals excluding zero). At 10M and above, TV point estimates are lower on all six corpora, with intervals excluding zero everywhere except code at 28M and 59M and WikiText-2 at 59M, and acceptance is higher, with intervals excluding zero, on every corpus except code. On code, acceptance varies by up to 10 points between seeds of the same arm, and no difference is detected at any size.

On the three further corpora (Appendix A.1) at 10M and above, TV point estimates are lower everywhere and acceptance point estimates higher everywhere except Hendrycks MATH at 59M, where the three seeds disagree widely (-5.0, -2.4, and $+1.6$ points); intervals include zero for TV on arXiv at 59M and on Hendrycks MATH at 10M and 59M, and for acceptance on arXiv at 28M and 59M and on Hendrycks MATH at 59M. The verifier's entropy shows no detectable change at any size, and its point estimates are all lower, so there is no sign that the gain comes from a flatter verifier. Over training, error-cancelled MTP starts behind on TV at 6M and 10M; it draws level by the end at 6M and passes standard MTP about halfway through training at 10M. It leads from 20% of training at 28M and from the first evaluation at 59M, where its lead narrows from 1.5 points early in training to 0.9 at the end, in all three seeds (Appendix A.2).

Against a KL loss. At 10M, error-cancelled MTP and the KL loss do not differ detectably in drafts or next-token loss (TV -0.01 ± 0.25 points, acceptance rate -0.04 ± 0.42 points, next-token loss -0.0001 ± 0.0039 nats). The TV interval rules out a difference as large as the effect over standard MTP. We trained the KL loss at 10M only.

4.2 Main-model loss

Error-cancelled − standard MTP	6M	10M	28M	59M
Next-token CE, FineWeb-Edu, nats	-0.0022 ± 0.0076	$+0.0035 \pm 0.0094$	-0.0005 ± 0.0034	-0.0016 ± 0.0015
Verdict	not detected	not detected	not detected	lower
Next-token CE, WikiText-2, nats	-0.010 ± 0.031	$+0.015 \pm 0.030$	$+0.009 \pm 0.012$	-0.003 ± 0.037
MTP CE, FineWeb-Edu, nats	$+0.0857 \pm 0.0110$	$+0.0429 \pm 0.0082$	$+0.0177 \pm 0.0027$	$+0.0072 \pm 0.0066$

No next-token difference is detected at 6M, 10M, or 28M. At 59M error-cancelled MTP is 0.0016 ± 0.0015 nats lower, with all three seeds lower; the interval only just excludes zero and the estimates do not move together across sizes, so we do not call it a trend. The MTP loss does differ: with the error term cancelled, the head's cross-entropy is higher, by 0.086 nats at 6M falling to 0.007 at 59M, with no two intervals overlapping.

Against next-token-only training:

Next-token CE minus next-token only, nats	6M	10M	28M	59M
Standard MTP, FineWeb-Edu	$+0.0084 \pm 0.0073$	-0.0049 ± 0.0146	$+0.0028 \pm 0.0024$	$+0.0018 \pm 0.0022$
Error-cancelled MTP, FineWeb-Edu	$+0.0062 \pm 0.0107$	-0.0014 ± 0.0068	$+0.0024 \pm 0.0015$	$+0.0002 \pm 0.0023$
Standard MTP, WikiText-2	$+0.041 \pm 0.023$	-0.020 ± 0.037	$+0.008 \pm 0.012$	$+0.012 \pm 0.033$
Error-cancelled MTP, WikiText-2	$+0.031 \pm 0.029$	-0.005 ± 0.029	$+0.017 \pm 0.008$	$+0.009 \pm 0.051$

At this weight MTP costs little next-token loss. On FineWeb-Edu, standard MTP costs 0.008 nats at 6M and 0.003 at 28M and error-cancelled MTP 0.002 at 28M; no cost is detected at the other sizes. On WikiText-2 the estimates are larger and noisier; their intervals exclude zero for both arms at 6M ($+0.041$ standard, $+0.031$ error-cancelled) and for error-cancelled MTP at 28M ($+0.017$).

4.3 Cost

Every arm timed back to back on one GPU, fp32, batch 64×512, for the 10M model and for the 28M model with a 50,257-token vocabulary (60M parameters):

Tokens/s, peak GiB	Next-token only	Standard MTP	Error-cancelled MTP	KL loss
10M model	501k, 8.4	387k, 10.4	386k, 10.4	345k, 13.4
28M model with a 50,257-token vocabulary (60M)	140k, 30.8	96k, 38.6	96k, 38.6	81k, 56.9

The KL loss has 11% lower throughput than standard MTP at 10M and 16% lower at 28M with the larger vocabulary, and needs 29% and 48% more peak memory; error-cancelled MTP's throughput is at most 0.4% lower, with no more peak memory. The training runs agree: at 10M, peak memory, which also holds the 4.8 GiB of training tokens kept on the GPU, is 15.2 GiB for both standard and error-cancelled MTP and 18.2 GiB for the KL loss.

5 Analysis

All three analyses are at 10M, 5 seeds.

5.1 Drafting needs the trunk to learn from the MTP loss; the cancellation's gain does not

We stopped every MTP gradient at the head's input, so the trunk learns from next-token loss alone while the head trains as before (Figure 3).

10M, 5 seeds	TV, points	Acceptance rate, points	Next-token CE, nats
Standard − standard with trunk cut	-5.26 ± 0.42	$+4.9 \pm 0.5$	-0.0052 ± 0.0109
Error-cancelled − error-cancelled with trunk cut	-4.94 ± 0.47	$+4.3 \pm 0.8$	-0.0033 ± 0.0085
Both with trunk cut: error-cancelled − standard	-1.14 ± 0.15	$+1.7 \pm 0.4$	$+0.0016 \pm 0.0065$

Without the trunk's MTP gradient, drafts are about 5 TV points worse, six times the effect of error cancellation, with no detectable change in next-token loss. With the trunk cut, error cancellation still lowers TV, by 1.14 ± 0.15 points, more than the 0.82 ± 0.21 with the trunk intact (paired difference 0.32 ± 0.19). Its gain therefore does not need the trunk: cancelling the error in the head's own gradient is enough, as the head then learns the gap term without the verifier's error.

5.2 Training the output layer on next-token loss alone: no detectable gain at this weight

Here the head reads W through `W.detach()`, so W learns from next-token loss alone while the head still uses it. We also report the effective rank [24] of $W\,\mathrm{diag}(g)$, which measures how evenly its singular values are spread; g is the gain of the trunk's final norm.

New arm − unmodified arm, 10M	Standard	Error-cancelled
Next-token CE, nats	-0.0024 ± 0.0076	-0.0019 ± 0.0049
TV, points	$+0.86 \pm 0.18$	$+0.63 \pm 0.23$
Acceptance rate, points	-0.8 ± 0.3	-0.6 ± 0.5
Effective rank	$+1.8 \pm 0.3$	$+2.0 \pm 0.2$

At this weight MTP has no detectable next-token cost at 10M (section 4.2), and this change brings no detectable next-token gain. Drafts get worse instead: training W on the head's loss is part of what makes the drafts good. Detaching W raises the effective rank by 1.8 ± 0.3 (standard) and 2.0 ± 0.2 (error-cancelled). Against next-token-only training's 229.0, standard MTP's rank is 2.3 ± 0.3 lower (about 1%), so detaching W closes most of that gap; the detached arms end

0.53 ± 0.14 and 0.34 ± 0.10 lower. In the main comparison, error-cancelled minus standard MTP's effective rank grows with size: -0.24 ± 0.21 at 6M, -0.07 ± 0.14 at 10M, $+2.2 \pm 0.9$ at 28M, and $+9.1 \pm 0.6$ at 59M.

5.3 Head training after pretraining

We froze each trained model's trunk and output layer and trained its heads against the frozen verifier, as drafters are commonly trained after pretraining [11, 25–28], with KL or TV as the loss [6, 7, 29] and the pretraining optimiser and schedule, for 10% of the pretraining token budget drawn from the training data, 5 to 7% of the pretraining time: fresh heads for next-token-only models, and the existing heads, trained further, for standard and error-cancelled models.

Minus error-cancelled MTP as trained, 10M	TV, points	Acceptance rate, points
Fresh KL-trained heads on a next-token-only model	$+21.04 \pm 0.50$	-18.1 ± 0.7
Standard MTP, heads trained further with KL	-0.13 ± 0.19	$+0.1 \pm 0.7$
Standard MTP, heads trained further with TV	-0.70 ± 0.20	$+0.1 \pm 0.5$
Error-cancelled MTP, heads trained further with KL	-0.47 ± 0.04	$+0.6 \pm 0.5$
Error-cancelled MTP, heads trained further with TV	-1.08 ± 0.03	$+0.8 \pm 0.4$

Fresh heads on a next-token-only model draft far worse than heads trained during pretraining, as Gloeckle et al. [4, §3.2] also state. Most of that gap is not the trunk's: these heads are also 14.96 ± 0.32 TV points worse than standard MTP with the trunk cut (section 5.1), whose trunk likewise learns from next-token loss alone but whose head trained throughout pretraining and whose output layer also learned from the head's loss. Extra head training lets standard MTP catch up: with KL-trained heads its drafts do not differ detectably from error-cancelled MTP as trained, and with TV-trained heads they are closer to the verifier. Given the same extra training, error-cancelled MTP stays closer to the verifier, by 0.34 ± 0.21 TV points with KL-trained heads and 0.38 ± 0.20 with TV-trained heads; its acceptance rate is higher by 0.7 ± 0.4 points with TV-trained heads, and the difference with KL-trained heads is not detected ($+0.5 \pm 0.8$). Training heads on TV rather than KL lowers TV by a further 0.6 points (-0.57 ± 0.02 for standard and -0.61 ± 0.03 for error-cancelled MTP), while acceptance rate moves by $+0.0 \pm 0.5$ and $+0.2 \pm 0.4$ points (Appendix A.3).

6 Limitations

- **Scale.** Our models range from 6M to 59M parameters at about 29 tokens per parameter. Larger models and longer training may behave differently, and four sizes cannot establish a trend beyond them.
- **Setup.** Untuned AdamW, an 8K vocabulary, 512-token sequences, 32K-token batches, and a one-layer MTP module on trunks of 6 to 16 layers, so the head is 7 to 19% the size of the trunk's transformer layers. We keep λ at 0.3 throughout, where DeepSeek-V3 lowers it to 0.1 for its last third of training.
- **Statistics.** One training corpus, 3 to 5 seeds, and intervals that cover initialisation and data order only.
- **Speed.** We report the acceptance rate of one drafted token per verifier call, not wall-clock speedup.
- **Benchmarks.** We report held-out loss and draft quality only, no downstream benchmarks.

7 Conclusion

Cancelling the verifier's error in the head's gradient makes the MTP head a better drafter at 10M parameters and above, within 0.4% of standard MTP's training speed. At DeepSeek-V3's MTP setting, drafts at 10M to 59M are 0.8 to 1.0 TV points closer to the verifier and accepted 1.1 to 1.9 points more often, and no next-token cost compared with standard MTP is detected at any size. At 10M the drafts do not differ detectably from a KL loss's, and cancelling the error in the head's own gradient is enough for the gain; a short round of extra head training after pretraining gives standard MTP as much, while error-cancelled MTP stays closer to the verifier when it receives the same.

Appendix

A.1 Drafts by corpus

Error-cancelled minus standard MTP on each evaluation corpus, in points: TV and acceptance rate (AR), with intervals.

Corpus	6M	10M	28M	59M
TV				
FineWeb-Edu	-0.03 ± 0.34	-0.82 ± 0.21	-0.97 ± 0.14	-0.87 ± 0.20
WikiText-2	-0.28 ± 0.42	-1.23 ± 0.61	-1.20 ± 0.07	-1.09 ± 1.10
PG-19	-0.54 ± 0.19	-1.23 ± 0.29	-1.24 ± 0.36	-1.05 ± 0.28
CodeSearchNet Python	$+0.46 \pm 3.15$	-1.17 ± 1.03	-1.52 ± 1.65	-0.95 ± 5.19
CNN/DailyMail	-0.22 ± 0.43	-0.94 ± 0.17	-1.07 ± 0.09	-0.90 ± 0.24
GSM8K	$+0.11 \pm 0.97$	-1.15 ± 0.82	-1.07 ± 0.37	-0.62 ± 0.18
AR				
FineWeb-Edu	$+0.6 \pm 0.9$	$+1.1 \pm 0.5$	$+1.9 \pm 0.9$	$+1.1 \pm 0.8$
WikiText-2	$+1.0 \pm 0.7$	$+1.5 \pm 0.3$	$+1.8 \pm 1.0$	$+1.4 \pm 1.2$
PG-19	$+0.7 \pm 0.7$	$+1.2 \pm 0.4$	$+1.3 \pm 1.1$	$+1.6 \pm 0.6$
CodeSearchNet Python	-0.3 ± 4.4	-0.3 ± 8.3	-0.8 ± 9.1	$+1.1 \pm 5.3$
CNN/DailyMail	$+0.7 \pm 0.4$	$+1.2 \pm 0.5$	$+1.3 \pm 0.8$	$+1.4 \pm 0.8$
GSM8K	$+0.5 \pm 0.5$	$+1.3 \pm 0.5$	$+1.9 \pm 0.2$	$+1.2 \pm 0.2$

Further corpora, with the same protocol:

Corpus	6M	10M	28M	59M
TV				
arXiv	$+0.48 \pm 1.31$	-0.85 ± 0.44	-1.27 ± 0.07	-0.93 ± 2.77
Hendrycks MATH	$+0.83 \pm 1.11$	-0.59 ± 0.80	-1.25 ± 0.90	-0.06 ± 5.35
Project Gutenberg shard	-0.43 ± 0.20	-1.21 ± 0.28	-1.17 ± 0.21	-1.07 ± 0.17
AR				
arXiv	$+1.0 \pm 0.6$	$+1.9 \pm 1.1$	$+1.5 \pm 2.5$	$+1.2 \pm 1.5$
Hendrycks MATH	$+0.7 \pm 0.5$	$+1.8 \pm 0.7$	$+1.7 \pm 1.4$	-2.0 ± 8.3
Project Gutenberg shard	$+0.7 \pm 0.7$	$+1.3 \pm 0.6$	$+1.4 \pm 0.3$	$+1.0 \pm 0.8$

A.2 Training curves

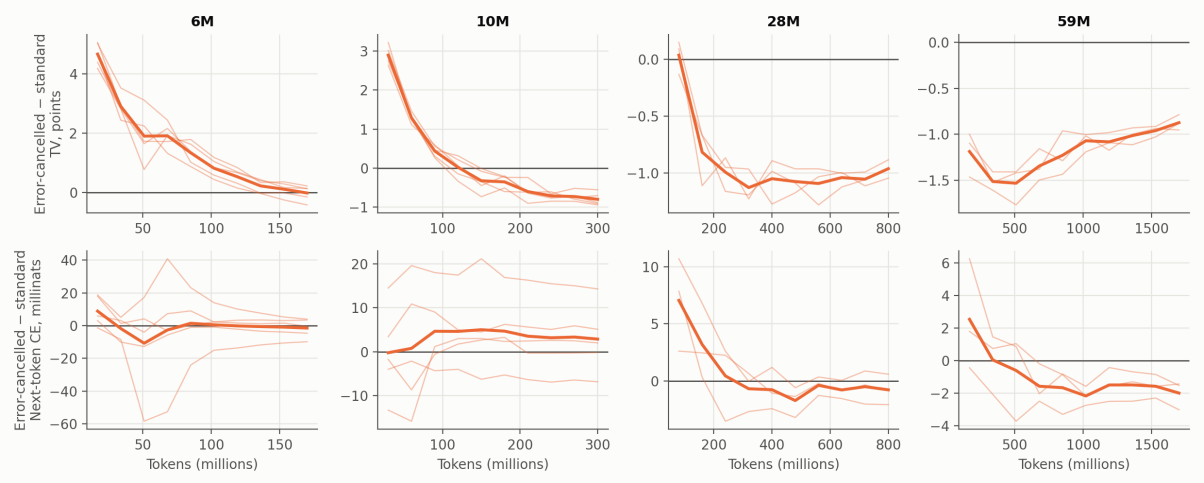


Figure 2. Error-cancelled minus standard MTP over training: TV (top) and next-token cross-entropy (bottom), on 1M held-out FineWeb-Edu tokens at each evaluation. Thin lines are seeds; the thick line is their mean.

A.3 Trunk cut and head training

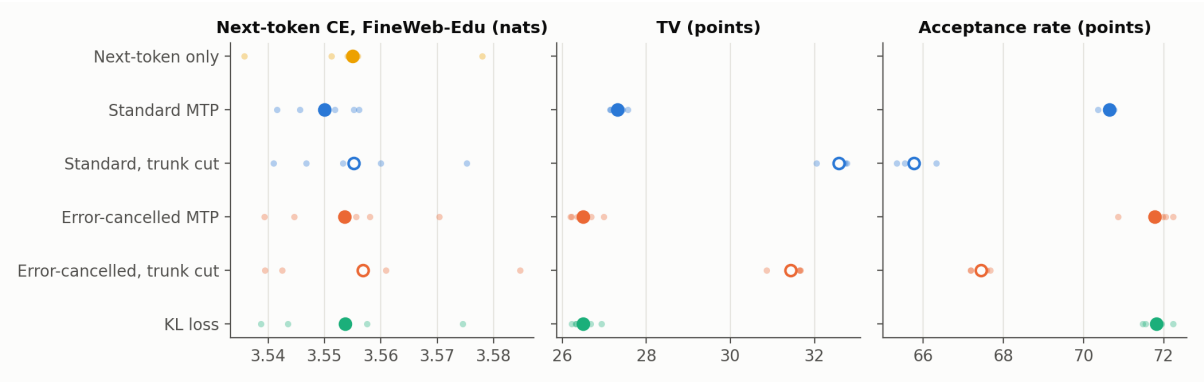


Figure 3. Trunk gradient cut at 10M: each arm's next-token cross-entropy, TV, and acceptance rate. Open circles have the trunk cut; large points are means, faint points are seeds.

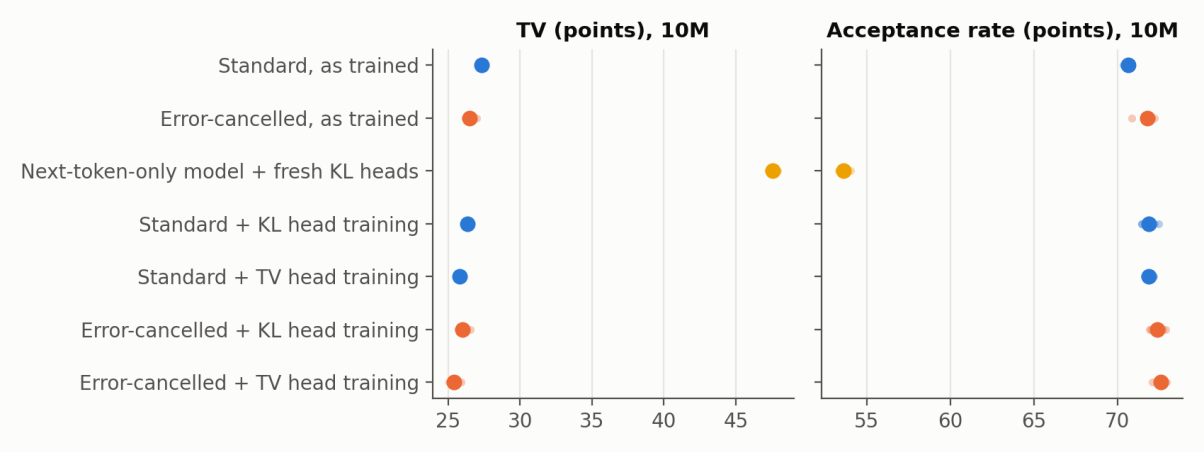


Figure 4. Head training after pretraining at 10M: each arm's TV and acceptance rate. Large points are means, faint points are seeds.

A.4 Implementation checks

Before training, we verified in float64 that the cancellation reproduces a KL loss's head and trunk gradients, to 3.9×10^{-18}; that the speculative sampler's draft distributions equal the training forward pass (to 2.2×10^{-16}); and that the sampler reproduces the verifier's distribution (chi-square $p = 0.32$ over 20,000 rollouts with three drafted tokens per call, which exercises every branch the one-token case uses). The verifier of every seed-1 pretraining run, 17 in all, was checked against a float64 copy (worst TV 3.8×10^{-4}).

References

[1] Leviathan, Kalman, Matias. Fast Inference from Transformers via Speculative Decoding. ICML 2023.

[2] Chen, Borgeaud, Irving, Lespiau, Sifre, Jumper. Accelerating Large Language Model Decoding with Speculative Sampling. arXiv:2302.01318, 2023.

[3] DeepSeek-AI. DeepSeek-V3 Technical Report. arXiv:2412.19437, 2024.

[4] Gloeckle et al. Better & Faster Large Language Models via Multi-token Prediction. ICML 2024.

[5] Stern, Shazeer, Uszkoreit. Blockwise Parallel Decoding for Deep Autoregressive Models. NeurIPS 2018.

[6] Zhou et al. DistillSpec: Improving Speculative Decoding via Knowledge Distillation. ICLR 2024.

[7] Samarin et al. LK Losses: Direct Acceptance Rate Optimization for Speculative Decoding. ICML 2026.
[8] Liu et al. Online Speculative Decoding. ICML 2024.
[9] Zhao et al. Self-Distillation for Multi-Token Prediction. arXiv:2603.23911, 2026.
[10] Teoh et al. Next-Latent Prediction Transformers Learn Compact World Models. NeurIPS 2026.
[11] NVIDIA. Nemotron 3 Ultra: Open, Efficient Mixture-of-Experts Hybrid Mamba-Transformer Model for Agentic Reasoning. arXiv:2606.15007, 2026.
[12] Touvron et al. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971, 2023.
[13] Penedo et al. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale. NeurIPS 2024 Datasets and Benchmarks.
[14] Loshchilov, Hutter. Decoupled Weight Decay Regularization. arXiv:1711.05101, 2017.
[15] Merity, Xiong, Bradbury, Socher. Pointer Sentinel Mixture Models. arXiv:1609.07843, 2016.
[16] Rae, Potapenko, Jayakumar, Lillicrap. Compressive Transformers for Long-Range Sequence Modelling. arXiv:1911.05507, 2019.
[17] Husain, Wu, Gazit, Allamanis, Brockschmidt. CodeSearchNet Challenge: Evaluating the State of Semantic Code Search. arXiv:1909.09436, 2019.
[18] See, Liu, Manning. Get To The Point: Summarization with Pointer-Generator Networks. ACL 2017.
[19] Cobbe et al. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168, 2021.
[20] Xia et al. Unlocking Efficiency in Large Language Model Inference: A Comprehensive Survey of Speculative Decoding. Findings of ACL 2024.
[21] Cohan et al. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. NAACL 2018.
[22] Hendrycks et al. Measuring Mathematical Problem Solving With the MATH Dataset. NeurIPS 2021.
[23] Project Gutenberg. `https://www.gutenberg.org`.
[24] Roy, Vetterli. The Effective Rank: A Measure of Effective Dimensionality. EUSIPCO 2007.
[25] Cai et al. Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads. ICML 2024.
[26] Li, Wei, Zhang, Zhang. EAGLE: Speculative Sampling Requires Rethinking Feature Uncertainty. ICML 2024.
[27] Ankner et al. Hydra: Sequentially-Dependent Draft Heads for Medusa Decoding. arXiv:2402.05109, 2024.
[28] Cai et al. FastMTP: Accelerating LLM Inference with Enhanced Multi-Token Prediction. arXiv:2509.18362, 2025.
[29] Li et al. Breaking Entropy Bounds: Accelerating RL Training via MTP with Rejection Sampling. arXiv:2606.12370, 2026.