Third-derivative regularization for causal language models

Libor “Burny” Burian · 25 November 2026

Abstract

We study whether a language model improves its predictions when training penalizes variation in the curvature of its next-token loss. Independent random signs produce a differentiable estimate of the squared norm of all third partial derivatives in the trainable coordinates. Two matched interventions compare this objective with ordinary AdamW and a signed Hessian-trace penalty. TinyStories: Awaiting paired test results (0 of 8 full-horizon training triplets). Research abstracts: Awaiting paired test results (0 of 8 full-horizon training triplets).

Motivation and related work

Hochreiter and Schmidhuber proposed that flatter minima may generalize better at comparable training error. Coarser weights can preserve the fit, so fewer bits can describe it. Their minimum-description-length argument predicts less expected overfitting. Rescaling can alter sharpness while preserving predictions, so we measure held-out token loss. Hochreiter and Schmidhuber (1994) · Hochreiter and Schmidhuber (1997) · Dinh et al.

The Hessian can change at constant trace. Squared third partials penalize this variation, adapting an order-three spline roughness measure. Earlier work studied trace regularization in recurrent language models and a symmetric perturbation approximation in GPT-2 fine-tuning. Duchon · Liu, Yu and Lin · Zhang, Li and Ju, Table 5

FUT++ uses differences between Hessian estimates at consecutive checkpoints to approximate third weight derivatives and predict the effect of reordering training batches. Here the squared norm enters the loss and changes the optimizer update. Yang et al., Appendix A.1, equations 14–15.

The estimator and its update

$$J(\theta) = L(\theta) + \lambda \cdot \|D^3 L(\theta)\|^2_F$$

For independent Rademacher vectors u and v, form $r = \nabla(u^T H v)$. Squaring and summing its coordinates, then averaging over probes, cancels every cross term with unmatched indices. The expectation is the squared Frobenius norm of the minibatch third derivative. The gradient through r supplies a fourth-derivative contraction to AdamW. The derivative minibatch contains two windows. A Bernoulli gate samples one eighth of updates and multiplies the active contribution by eight. Bujanović and Kressner · Preregistration

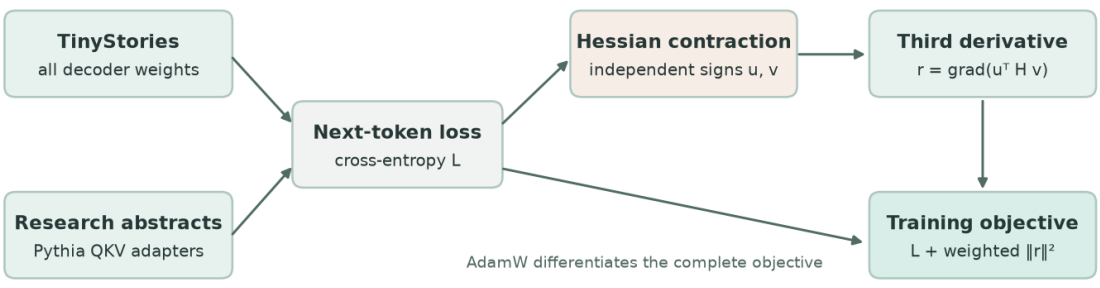


Figure 1. Trainable coordinate sets and the stochastic regularization path.

Experimental design

Full-weight pretraining uses a four-block causal Transformer with width 192, four heads, a gated 512-wide MLP, tied embeddings and 128 targets per window. Stories use a train-only vocabulary of 4,096 tokens. The adapter stage freezes Pythia-410M and trains rank-eight attention QKV adapters with scale alpha 16. The first-submission timestamp sets the split within 2024, grouping versions by base arXiv identifier. January through April supply training text, May supplies development text, and June supplies the held-out test. Eldan and Li · Biderman et al. · Hu et al. · Common Pile

The confirmatory horizons are 16,777,216 target tokens for stories and 524,288 for abstracts. Within a stage, each seed pairs starting weights, text order and optimizer schedule. The pilot calibrates penalty gradients on training data and selects strength at equal development horizons. The squared-Hessian, sign-randomized and combined arms test derivative order, gradient direction and incremental benefit. Initial checkpoints and token unigrams supply untrained references. Run matrix

Each update averages 32 windows for stories or eight for adapters. AdamW uses $\beta_1 = 0.9$, $\beta_2 = 0.95$, matrix weight decay 0.1 and gradient norm clipping at one. Peak learning rates are 0.0003 and 0.0001, respectively, with 5% warmup and cosine decay to 10% of peak. Across 16 batches at the shared pilot initialization, the median ordinary-to-penalty gradient-norm ratio sets the scale. Both regularizers try 1%, 3% and 10% of this scale. Training and selection protocol

Normalize and deduplicate whole documents before fitting the stories tokenizer. Prioritize evaluation records when removing overlaps, and audit word-shingle similarity between splits. Pack documents with an end-of-sequence token. Full evaluation counts each valid target once. During training, the development curve uses a fixed subset of windows. Data preparation and overlap audit

Corpus	Split	Retained documents	Packed tokens
TinyStories	dev	3,106	715,644
TinyStories	test	21,990	4,884,131
TinyStories	train	154,766	35,478,911
Research abstracts	dev	21,115	5,124,072
Research abstracts	test	20,199	4,912,988
Research abstracts	train	77,181	18,617,236

Packed-token counts include document separators. The number of valid next-token targets is recorded separately during evaluation.

Statistical analysis

Corpus negative log-likelihood sums token losses and divides by the valid target count. For each baseline, paired differences subtract baseline NLL from third-order NLL at the same token budget. The registered stage decision requires both one-sided exact paired sign tests to pass Holm correction at 0.05 across eight seed pairs. Ties are removed from sign-test denominators. A distribution-free order-statistic interval quantifies uncertainty in the median paired difference. Equal-time analyses use the latest logged development checkpoint within the shortest final training time of that seed triple. Bouthillier et al. · Decision rule

Results

Numerical accuracy on the pretrained checkpoint

For identical Pythia weights and probe signs, changing the FP32 CUDA batching gave a 48.7% relative discrepancy in the training gradient. In FP64 the CPU and GPU gradients agreed to relative error 1.4e-05. The initial FP32 pilot is retained as a diagnostic. The amended adapter arms compute their penalties in FP64 on the CPU and use CUDA for the ordinary loss and optimizer updates. Pre-test amendment and raw checks

The ledger contains 25 runs and 24 completed summaries. Raw records · Per-run ledger

TinyStories. Awaiting paired test results (0 of 8 full-horizon training triplets).

Development selected trace ratio 0.010 and third-order ratio 0.010. All candidate scores

The selected-arm pilot speeds project 1.69 training hours for this full confirmatory stage. This estimate covers optimization time. Measured timing basis

In the development pilot at 2,097,152 training targets, third-order training recorded 0.00004 nats per target token lower against Ordinary AdamW, 0.00024 nats per target token lower against Trace + third derivative, 0.00175 nats per target token lower against Squared Hessian norm, 0.00032 nats per target token lower against Randomized gradient signs, 0.00034 nats per target token lower against Hessian trace. The comparison uses one seed and strengths selected on development.

Research abstracts. Awaiting paired test results (0 of 8 full-horizon training triplets).

Development selected trace ratio 0.010 and third-order ratio 0.010. All candidate scores

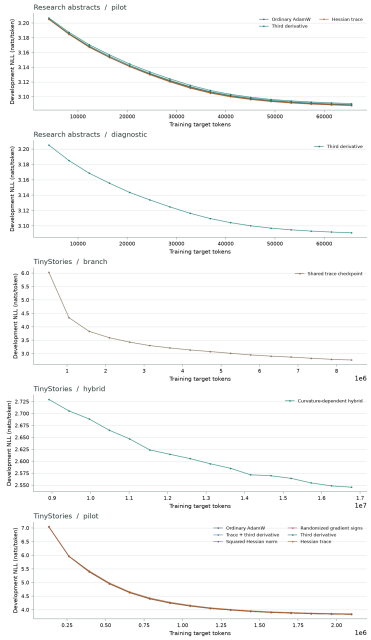
The selected-arm pilot speeds project 9.54 training hours for this full confirmatory stage. This estimate covers optimization time. Measured timing basis

In the development pilot at 65,536 training targets, third-order training recorded 0.00030 nats per target token higher against Ordinary AdamW, 0.00027 nats per target token higher against Hessian trace. The comparison uses one seed and strengths selected on development.

Scores across methods

Corpus	Phase	Method	Ratio	Runs	Tokens/run	Dev NLL	Test NLL
Research abstracts	control	Initial checkpoint	0.000	1	0	3.2517	—
Research abstracts	control	Unigram	0.000	1	0	7.3306	—
Research abstracts	pilot	Ordinary AdamW	0.000	1	65,536	3.1139	—
Research abstracts	pilot	Third derivative	0.010	1	65,536	3.1142	—
Research abstracts	diagnostic	Third derivative	0.030	1	65,536	3.1168	—
Research abstracts	pilot	Third derivative	0.030	1	65,536	3.1149	—
Research abstracts	pilot	Third derivative	0.100	1	65,536	3.1163	—
Research abstracts	pilot	Hessian trace	0.010	1	65,536	3.1140	—
Research abstracts	pilot	Hessian trace	0.030	1	65,536	3.1141	—
Research abstracts	pilot	Hessian trace	0.100	1	65,536	3.1143	—
TinyStories	branch	Shared trace checkpoint	0.010	1	8,388,608	2.7577	—
TinyStories	control	Initial checkpoint	0.000	1	0	8.3437	—
TinyStories	control	Unigram	0.000	1	0	6.0036	—
TinyStories	hybrid	Curvature-dependent hybrid	0.010	1	16,777,216	2.5459	—
TinyStories	pilot	Ordinary AdamW	0.000	1	2,097,152	3.8770	—
TinyStories	pilot	Trace + third derivative	0.010	1	2,097,152	3.8772	—
TinyStories	pilot	Squared Hessian norm	0.030	1	2,097,152	3.8787	—
TinyStories	pilot	Randomized gradient signs	0.010	1	2,097,152	3.8773	—
TinyStories	pilot	Third derivative	0.010	1	2,097,152	3.8769	—
TinyStories	pilot	Third derivative	0.030	1	2,097,152	3.8778	—
TinyStories	pilot	Third derivative	0.100	1	2,097,152	3.8827	—
TinyStories	pilot	Hessian trace	0.010	1	2,097,152	3.8773	—
TinyStories	pilot	Hessian trace	0.030	1	2,097,152	3.8802	—
TinyStories	pilot	Hessian trace	0.100	1	2,097,152	3.8979	—

Table 1. Medians over available completed runs sharing corpus, method, development strength and token horizon. NLL is in nats per target token. Dashes mark missing observations. The paired seed analyses use individual final test scores.



Development NLL by logged token count. Each line is a run; panels separate training stages and phases.

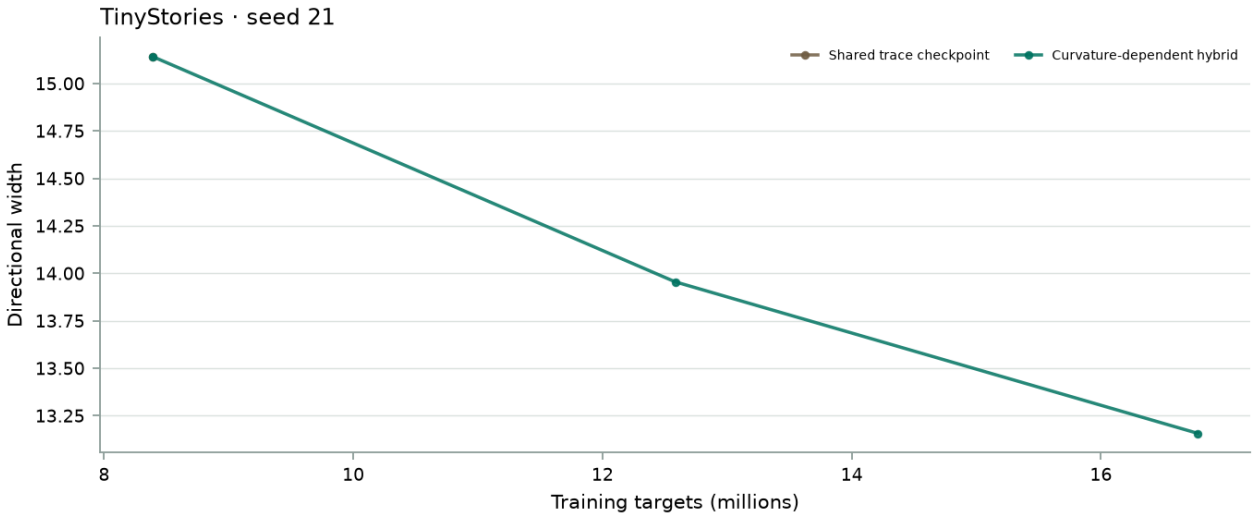
Discussion and limitations

A wide minimum permits large weight displacements at a fixed loss increase. If curvature starts small and changes slowly over the displacement, the loss stays low. For the loss w^4, the squared third derivative grows quadratically with distance from the bottom, so a small descent step can pull displaced weights back. A steep quadratic has constant curvature and receives zero third-derivative penalty everywhere. Finding wider basins remains a geometric hypothesis. Dinh et al. · Derivation and examples

Predictive utility requires lower held-out loss against both baselines. A reduction in the derivative estimate alone describes local geometry. Parameterization changes its magnitude, including the low-rank factorization used for abstracts. The estimate targets expected two-window minibatch roughness. The small datasets, limited seed count and two tokenization schemes bound the interpretation. More tokens per GPU hour can change the practical ranking. Dinh et al. · Scope of inference

Registered curvature-dependent schedule

The TinyStories follow-up keeps the trace active and increases the weight on squared third derivatives once training loss and measured curvature are low. Retaining the highest weight reached could oppose later drift if curvature changes faster at the edge. Direct division can reverse the incentive. For the quartic loss above, the squared ratio of third to second derivative is $4/w^2$ away from zero and grows toward the flat bottom. The final appendix specifies both schedules and the width test. Frozen follow-up design



1 of 48 full hybrid arms and 1 of 8 shared checkpoints have completed. Geometric endpoint: awaiting eight paired width measurements. All observed widths, development scores and fit differences

Conclusion

The outstanding paired evaluations determine which claims can be made about prediction quality at the registered horizons. The run ledger preserves every available observation and implementation failure.

Appendix · Computational cost

NVIDIA GeForce RTX 5070 Laptop GPU

Corpus	Phase	Method	Ratio	Runs	Targets/run	Train (s)	Targets/s	Peak GPU (MiB)
Research abstracts	control	Initial checkpoint	0.000	1	0	—	—	2,337.3
Research abstracts	control	Unigram	0.000	1	0	—	—	0.8
Research abstracts	pilot	Ordinary AdamW	0.000	1	65,536	39.5	1,659	2,346.3
Research abstracts	pilot	Third derivative	0.010	1	65,536	346.5	189	2,346.3
Research abstracts	diagnostic	Third derivative	0.030	1	65,536	72.3	907	6,369.6
Research abstracts	pilot	Third derivative	0.030	1	65,536	349.3	188	2,346.3
Research abstracts	pilot	Third derivative	0.100	1	65,536	339.7	193	2,346.3
Research abstracts	pilot	Hessian trace	0.010	1	65,536	150.8	435	2,346.3
Research abstracts	pilot	Hessian trace	0.030	1	65,536	153.6	427	2,346.3
Research abstracts	pilot	Hessian trace	0.100	1	65,536	152.2	431	2,346.3
TinyStories	branch	Shared trace checkpoint	0.010	1	8,388,608	120.6	69,581	608.2
TinyStories	control	Initial checkpoint	0.000	1	0	—	—	138.0
TinyStories	control	Unigram	0.000	1	0	—	—	0.1
TinyStories	hybrid	Curvature-dependent hybrid	0.010	1	16,777,216	381.5	43,973	617.3
TinyStories	pilot	Ordinary AdamW	0.000	1	2,097,152	16.9	123,871	608.2
TinyStories	pilot	Trace + third derivative	0.010	1	2,097,152	51.1	41,030	619.3
TinyStories	pilot	Squared Hessian norm	0.030	1	2,097,152	25.0	83,790	608.2
TinyStories	pilot	Randomized gradient signs	0.010	1	2,097,152	41.2	50,914	608.6
TinyStories	pilot	Third derivative	0.010	1	2,097,152	49.4	42,481	608.2
TinyStories	pilot	Third derivative	0.030	1	2,097,152	45.5	46,072	608.6
TinyStories	pilot	Third derivative	0.100	1	2,097,152	47.8	43,901	608.2
TinyStories	pilot	Hessian trace	0.010	1	2,097,152	28.7	73,185	608.2
TinyStories	pilot	Hessian trace	0.030	1	2,097,152	30.8	68,106	608.2
TinyStories	pilot	Hessian trace	0.100	1	2,097,152	29.7	70,730	608.2

Times and rates are per-run medians within each hardware, method, phase, strength and target budget. The clock sums synchronized training steps, including batch transfer and the optimizer update and any host FP64 probe while the GPU is reserved. Evaluation, checkpointing and uploads take additional time. Peak memory is PyTorch’s recorded maximum

allocated GPU tensor memory during the run, including evaluation. Untrained controls have no training speed.

Development NLL at matched GPU training time

For each seed the cutoff is the shortest completed training time across AdamW, trace and third-order training. Fixed-window development evaluations occur at discrete checkpoints. We take the latest checkpoint at or before the cutoff. Each cell gives NLL in nats per target token, followed by its lag behind the cutoff in seconds. Larger lags make the time comparison coarser. The archived per-seed table also records the token count at each checkpoint. Per-seed timing records

Awaiting a complete, timed three-method confirmatory seed with eligible development checkpoints.

Appendix · Output examples

For each stage and method, the first registered checkpoint and sample in run-ID order is shown in excerpt. The experiment ledger contains every full saved continuation and its corresponding reference when recorded. All outputs

Research abstracts · Initial checkpoint · control, seed 0

Prompt. One of the central objectives of contemporary neuroimaging research is to create predictive models

Reference. that can disentangle the connection between patterns of functional connectivity across the entire brain and various behavioral traits. Previous studies have shown that models trained to predict behavioral features from the individual's functional

Prediction. that will be used by neuroscience researchers to identify cases of a disease or to select appropriate therapeutic intervention. The research of Peter Seager and colleagues has demonstrated that the brain can

Research abstracts · Unigram · control, seed 0

Prompt. One of the central objectives of contemporary neuroimaging research is to create predictive models

Reference. that can disentangle the connection between patterns of functional connectivity across the entire brain and various behavioral traits. Previous studies have shown that models trained to predict behavioral features from the individual's functional

Prediction. neural. is novel In used describe sacrific the theruction To their zo than chronic current the the factor an(simultaneouslyrelated exchange purpose,) was} standards of radio.

Research abstracts · Ordinary AdamW · pilot, seed 0

Prompt. One of the central objectives of contemporary neuroimaging research is to create predictive models

Reference. that can disentangle the connection between patterns of functional connectivity across the entire brain and various behavioral traits. Previous studies have shown that models trained to predict behavioral features from the individual's functional

Prediction. that will accurately predict the behavior of patient-specific neurological disorders such as epilepsy. In this paper, we present a novel dataset of EEG data recorded from children with focal subcortical epilepsy

Research abstracts · Third derivative · pilot, seed 0

Prompt. One of the central objectives of contemporary neuroimaging research is to create predictive models

Reference. that can disentangle the connection between patterns of functional connectivity across the entire brain and various behavioral traits. Previous studies have shown that models trained to predict behavioral features from the individual's functional

Prediction. that will accurately predict the behavior of patient-specific neurological disorders such as epilepsy. In this paper, we present a novel dataset of EEG data recorded from children with focal subcortical epilepsy

Research abstracts · Hessian trace · pilot, seed 0

Prompt. One of the central objectives of contemporary neuroimaging research is to create predictive models

Reference. that can disentangle the connection between patterns of functional connectivity across the entire brain and various behavioral traits. Previous studies have shown that models trained to predict behavioral features from the individual's functional

Prediction. that will be used to assess the cognitive performance of neurological disorders. We will focus on three major areas: (1) the nervous system as a whole, (2) the brain regions that

TinyStories · Initial checkpoint · control, seed 0

Prompt. Once upon a time, there was a clever little dog named Max. Max loved

Reference. to run and play with his friends in the park. One day, Max was running very fast when he fell and hurt his knee. Max went to his friend, the wise old owl,

Prediction. break painted pleaseerp gif neat lead bro comfort rid shr search impressedosairs try ants oss note daughter woodLater rest independent stars stonesiver mixaybe Le spoon readMe fingersiew embar bached mail

TinyStories · Unigram · control, seed 0

Prompt. Once upon a time, there was a clever little dog named Max. Max loved

Reference. to run and play with his friends in the park. One day, Max was running very fast when he fell and hurt his knee. Max went to his friend, the wise old owl,

Prediction. it said her mom girlan he said that fell flut back's hard. a them into work protectt hug?" his and little teeth "What sky some Timmy and on down As's very

TinyStories · Curvature-dependent hybrid · hybrid, seed 21

Prompt. Once upon a time, there was a clever little dog named Max. Max loved

Reference. to run and play with his friends in the park. One day, Max was running very fast when he fell and hurt his knee. Max went to his friend, the wise old owl,

Prediction. to play on the swings and slide. His owner liked to run and play outside. Lily was happy to see his dog and play. One day, Max found a big hole in his house

TinyStories · Ordinary AdamW · pilot, seed 0

Prompt. Once upon a time, there was a clever little dog named Max. Max loved

Reference. to run and play with his friends in the park. One day, Max was running very fast when he fell and hurt his knee. Max went to his friend, the wise old owl,

Prediction. to play on the cat.

TinyStories · Trace + third derivative · pilot, seed 0

Prompt. Once upon a time, there was a clever little dog named Max. Max loved

Reference. to run and play with his friends in the park. One day, Max was running very fast when he fell and hurt his knee. Max went to his friend, the wise old owl,

Prediction. to play on the cat.

TinyStories · Squared Hessian norm · pilot, seed 0

Prompt. Once upon a time, there was a clever little dog named Max. Max loved

Reference. to run and play with his friends in the park. One day, Max was running very fast when he fell and hurt his knee. Max went to his friend, the wise old owl,

Prediction. to play on the cat.

TinyStories · Randomized gradient signs · pilot, seed 0

Prompt. Once upon a time, there was a clever little dog named Max. Max loved

Reference. to run and play with his friends in the park. One day, Max was running very fast when he fell and hurt his knee. Max went to his friend, the wise old owl,

Prediction. to play on the cat.

TinyStories · Third derivative · pilot, seed 0

Prompt. Once upon a time, there was a clever little dog named Max. Max loved

Reference. to run and play with his friends in the park. One day, Max was running very fast when he fell and hurt his knee. Max went to his friend, the wise old owl,

Prediction. to play on the cat.

TinyStories · Hessian trace · pilot, seed 0

Prompt. Once upon a time, there was a clever little dog named Max. Max loved

Reference. to run and play with his friends in the park. One day, Max was running very fast when he fell and hurt his knee. Max went to his friend, the wise old owl,

Prediction. to play on the cat.

References and reproducibility

Hochreiter and Schmidhuber (1994), Simplifying neural nets by discovering flat minima

Hochreiter and Schmidhuber (1997), Flat Minima

Zhang, Li and Ju, Noise Stability Optimization for Finding Flat Minima: A Hessian-based Regularization Approach

Duchon, Splines minimizing rotation-invariant semi-norms in Sobolev spaces

Liu, Yu and Lin, Regularizing deep neural networks with stochastic estimators of Hessian trace

Bujanović and Kressner, Norm and trace estimation with random rank-one vectors

Yang et al., Estimating the effects of sample training orders for large language models without retraining

Eldan and Li, TinyStories

Biderman et al., Pythia

Hu et al., Low-rank adaptation of large language models

Common Pile, Filtered arXiv abstracts

Bouthillier et al., Accounting for variance in machine learning benchmarks

Dinh et al., Sharp minima can generalize for deep nets

Rebuild tables, figures, the explorer and this paper with **uv run thirdloss report**. Code and versioned protocol

Appendix · Registered TinyStories schedules

The follow-up asks whether changing the third-derivative coefficient in response to low curvature helps retain a wider low-loss neighborhood. Eight paired seeds compare six arms at equal training tokens.

Shared training objective

$$J_t = L_t + (I_t/p)\,[\alpha_t\, h_t + \lambda_t\, P_{3,t}]$$

$$h_t = a^{\mathrm{T}} H_t a, \quad P_{3,t} = \|T_t[u, v, \cdot]\|^2$$

L_t is the full update's next-token loss. H_t and T_t are the Hessian and third derivative of the unregularized two-window loss in the trainable weights. The trace probe a and the two independent third-derivative probes u, v have random sign entries. I_t is the Bernoulli gate with activation probability p = 1/8. Both coefficients are held fixed in each weight update.

Fixed-token ramp and controls

$$s_t = \min(1, \max(0, (n_t - n_0)/R)), \quad \lambda_t = \lambda_{\max}\, s_t$$

Here n_t counts training targets before update t, n_0 is the switch point, and R is the positive ramp length in tokens. The trace coefficient stays at α. Trace alone sets $\lambda_t = 0$. Stronger trace sets $\lambda_t = 0$ and $\alpha_t = \alpha + \Delta\alpha\, s_t$, with Δα calibrated to the added penalty-gradient norm. The always-on comparison sets $\lambda_t = \lambda_{\max}$ from the start.

Curvature-dependent coefficient

$$c_t = [(1/(md)) \textstyle\sum_{j \in S(t)} \|H_j z_j\|^2]^{1/2}$$

$$\lambda_t = \max(\lambda_{t-1}, \lambda_{\max} \max(0, 1 - c_t/c_*))$$

S(t) contains the last m training-only measurements before the update. Each z_j is a random sign probe and d is the number of trainable weights. Start λ at zero. Enable this rule after warm-up once the loss criterion is met and the window is full. Below the fixed positive threshold c_*, smaller curvature can raise λ up to $\lambda_{\max}$. Retaining the peak keeps the penalty active when curvature rebounds. A control replays other seeds' coefficient histories in a prespecified permutation.

Directional width

$$r_{\pm}(\theta, e) = \inf \{\, r > 0 \mid L_{\text{probe}}(\theta \pm re) - L_{\text{probe}}(\theta) \geq \delta \,\}$$

$$W_\delta(\theta) = \text{median}_e\, [r_+(\theta, e) + r_-(\theta, e)]$$

Use fixed unit directions e, fixed training examples, unregularized loss L_{probe}, and a positive loss tolerance δ. Compare changes in W_δ from a common trace-trained checkpoint at matched token counts. Held-out development loss tests prediction separately. Derivation · Controls and readouts

AI statement

Codex wrote the codebase and paper from the exact specification provided by a human. Human cleanup and rewriting are in progress.