Continuous Autoregressive Language Models

Generate a token block with one backbone step

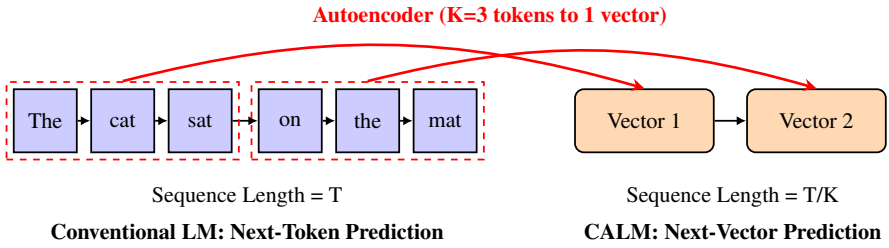
Paper authors: Chenze Shao, Darren Li, Fandong Meng, Jie Zhou
WeChat AI, Tencent; Qiuzhen College, Tsinghua University

arXiv:2510.27688v1 | 31 October 2025
Research discussion | 14 October 2026

Robust latent codes + a one-pass generator + a sampling-based toolkit.

Paper: arXiv:2510.27688v1 Official implementation

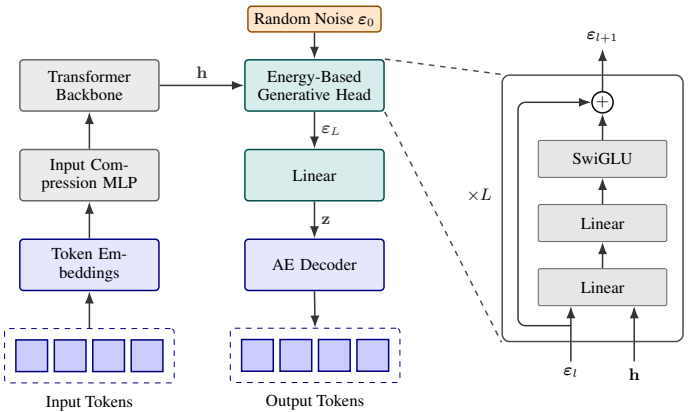
Increase the amount of text predicted per step



- A length-T sequence becomes T/K backbone positions.
- Enumerating every discrete block would need up to $|\mathcal{V}|^K$ classes.
- A continuous vector represents the block; a small decoder emits its tokens.

Paper, Sec. 1, pp. 1–3; original Fig. 1, p. 2. Figure uses K=3; main experiments use K=4.

Continuous prediction uses a discrete feedback loop



1. Compress K token embeddings.
2. Run the causal backbone once.
3. Sample a vector from $G(\mathbf{h}, \varepsilon)$.
4. Decode K tokens and feed them back.

One shared latent variable carries within-block dependence.

Paper, Sec. 3.3.3, pp. 6–8; original Fig. 2, p. 7. AE decoder is frozen.

The autoencoder must reconstruct and tolerate noise

Context-free encoder

Token embeddings + position-wise FFN

Flatten: $Kd_{\mathrm{AE}} \to d_{\mathrm{AE}}$

FFN + projection to $(\boldsymbol{\mu}, \boldsymbol{\sigma})$

$$\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))$$

Parallel decoder

Latent $\to d_{\mathrm{AE}} \to Kd_{\mathrm{AE}}$

Reshape + FFN + tied vocabulary projection

Argmax at all K positions

$K = 4$, $d_{\mathrm{AE}} = 512$, $l = 128$.

$> 99.9\%$ token reconstruction accuracy, with posterior std. about 0.3.

Paper, Sec. 2, pp. 3–4. Reconstruction accuracy measures the AE, not next-block prediction.

Redundancy makes the latent space easier to model

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ae}} + \beta \sum_{j=1}^{l} \max(\lambda_{\text{KL}}, \mathcal{L}_{\text{KL},j})$$

$$\mathcal{L}_{\text{KL},j} = \tfrac{1}{2}(\mu_j^2 + \sigma_j^2 - 1 - \log \sigma_j^2)$$

- Variational regularization: $\beta = 0.001$.
- KL floor $\lambda_{\text{KL}} = 0.5$: removes the incentive to reduce small KL terms further.
- Token masking and latent dropout: both $p = 0.15$ during AE training.
- Dropout is disabled during subsequent LM training and inference.

Paper, Sec. 2.2, p. 4, Eqs. (2)–(4). The floor applies to the loss; it is not a hard lower bound on actual KL.

Energy scoring learns a distribution from samples

$$S(P, \mathbf{y}) = \underbrace{\mathbb{E}\|\mathbf{x}' - \mathbf{x}''\|^{\alpha}}_{\text{sample diversity}} - 2 \quad \underbrace{\mathbb{E}\|\mathbf{x} - \mathbf{y}\|^{\alpha}}_{\text{fidelity to the target}}$$

- $\mathbf{x}, \mathbf{x}', \mathbf{x}''$ are independent draws from the generator.
- Higher expected score is better; training minimizes its negative.
- Strictly proper for $0 < \alpha < 2$, under finite-moment conditions.
- Default $\alpha = 1$. At $\alpha = 2$, only the predictive mean is identified.

One-pass generative head

Condition residual MLP blocks on **h**; transform uniform noise into **z**. No iterative denoising is required to obtain a base sample.

Paper, Sec. 3.3, pp. 6–7, Eqs. (8)–(9). Mean-only interpretation: discussion derivation.

Reuse the backbone state across multiple head samples

At each position i, estimate the negative energy score:

$$\ell_i = \frac{2}{NM} \sum_{n=1}^{N} \sum_{m=1}^{M} \|\mathbf{z}_{i,m} - \tilde{\mathbf{z}}_{i,n}\| - \frac{1}{N(N-1)} \sum_{n \neq k} \|\tilde{\mathbf{z}}_{i,n} - \tilde{\mathbf{z}}_{i,k}\|.$$

- $N = 8$ model samples $\tilde{\mathbf{z}}$: require head evaluations.
- $M = 100$ target samples $\mathbf{z}$: drawn from the known AE posterior.
- A shared Transformer state conditions all model samples.
- Multiple posterior targets reduce the variance of the training signal.

Paper, Sec. 3.3.2, p. 6, Eq. (10); displayed as the per-position loss.

Training separates representation learning from modeling

Two training stages

1. Train the AE with reconstruction and regularization.
2. Freeze the AE. Train the backbone and head with energy loss.

True history tokens provide the input. The AE encoder provides target posteriors.

Generation

1. Compress the previous token block.
2. Predict one latent vector.
3. Decode the next block in parallel.

Input compression is a separate trainable module, not the frozen AE encoder.

Base generation advances the backbone once per block of K tokens.

Paper, Secs. 2, 3.3.3 and 7.1, pp. 3–8 and 14–15.

A collision estimates the Brier uncertainty penalty

$$\mathrm{Brier}(P, y) = 2P(y) - \sum_x P(x)^2$$

$$\mathbb{E}_{y\sim Q}[\mathrm{Brier}(P, y)] = -\|P - Q\|_2^2 + \|Q\|_2^2$$

Two independent samples are enough for an unbiased component estimate

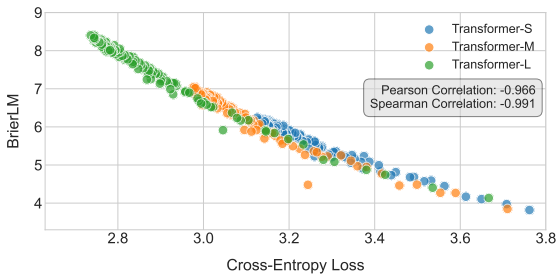
$$\widehat{B} = \mathbf{1}\{X_1 = y\} + \mathbf{1}\{X_2 = y\} - \mathbf{1}\{X_1 = X_2\}$$

- Exact n-gram matching treats the whole n-gram as one outcome.
- Rewards reference matches while penalizing concentrated predictions.

Paper, Sec. 4.2, pp. 8–9, Eqs. (12)–(14). This score is higher-is-better.

BrierLM combines short continuation scores

$$\text{BrierLM} = 100 \left(\prod_{n=1}^{4} \text{Brier-}n \right)^{1/4}$$



- Strong alignment with cross-entropy on baseline checkpoints.
- Pearson: -0.966; Spearman: -0.991.
- Evaluates short continuations given a true prefix.

Discussion: component unbiasedness does not survive nonlinear aggregation; negative components need a defined treatment.

Paper, Sec. 4.2, p. 9, Eq. (15) and original Fig. 3. Formal caveat and counterexample in backup.

Repeated discrete samples implement low temperature

$$P_T(x) = \frac{P(x)^{1/T}}{Z_T}, \qquad 0 < T < 1, \quad x = \text{a decoded token block.}$$

- If $T = 1/n$: draw n independent blocks and accept only if all match.
- If $1/T = n + \alpha$: add a Bernoulli factory with acceptance $P(x)^{\alpha}$.
- Per-trial accepted mass: $P(x)^n P(x)^{\alpha} = P(x)^{1/T}$.

Exactness can be expensive

At $T = 1/n$, the expected sample count is n/Z_T. A uniform distribution over S blocks costs nS^{n-1} calls on average.

Paper, Secs. 5.1–5.2, pp. 10–11; Theorems 1–2; Appendix A.1–A.2, pp. 30–32. Uniform example: derivation.

A finite batch trades exactness for useful sample reuse

Draw N blocks, count each outcome, and weight matching subsets:

$$w_x = \binom{c_x}{n}, \qquad T = 1/n, \qquad P(x \mid \text{batch}) = \frac{w_x}{\sum_z w_z}.$$

Example: $n = 2$

Block	Count	Weight
A	3	3
B	2	1
Other	1	0

$$P(A \mid \text{batch}) = 3/4.$$

- If no n-fold match exists, reduce the matching order until one exists.
- Finite N is biased: random ratios and fallback alter the target.
- As $N \to \infty$, the output converges to P_T.

Paper, Sec. 5.3, pp. 11–12, Algorithm 2 and Theorem 3; proof on p. 33.

The comparison fixes backbone positions, not token context

Data	Pile uncopyrighted; about 230B tokens with Llama 3 tokenization
Evaluation	WikiText-103; BrierLM ↑
AE training	15B-token subset; 30K steps; about 75M parameters
LM training	250K steps, about 2M tokens per batch
Context	2048 backbone positions: 2048 tokens for AR, 8192 for CALM at $K = 4$
Ablations	12 layers, width 768, 3 head blocks; 50K training steps

Main-table costs include AE parameters, AE training, and encoding/decoding.

Paper, Sec. 7.1, pp. 14–15; Table 1 caption; Secs. 7.3–7.4, pp. 16–17.

Fewer FLOPs buy scores close to discrete baselines

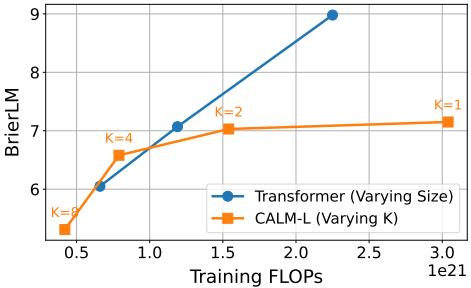
Model	Params	Train 10^{20} FLOPs	Infer/token 10^{8} FLOPs	BrierLM ↑
Transformer-S	281M	6.6	4.4	6.05
Transformer-M	465M	11.9	7.9	7.07
Transformer-L	849M	22.5	15.0	8.98
CALM-M ($K = 4$)	371M	3.7	2.9	5.72
CALM-L ($K = 4$)	735M	7.7	4.6	6.58
CALM-XL ($K = 4$)	1.82B	19.5	9.4	8.53

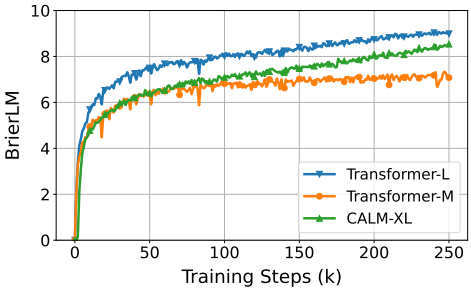
CALM-M versus Transformer-S

43.9% fewer training FLOPs; 34.1% fewer inference FLOPs.
BrierLM is lower by 0.33 points. End-to-end latency is not reported.

Paper, Table 1 and Sec. 7.2, p. 15. Percentages calculated from the reported values.

Bandwidth helps until the prediction task becomes too hard





$K = 1$: overhead without fewer steps.
$K = 4$: useful trade-off; $K = 8$: larger score loss.

CALM-XL improves more slowly at first, then closes part of the gap to Transformer-L.

The left plot uses training FLOPs; the right plot uses training steps.

Paper, Sec. 7.2, pp. 15–16; original Figs. 4–5, p. 16. Larger models supporting larger K is a hypothesis.

The full AE regularization recipe improves downstream quality

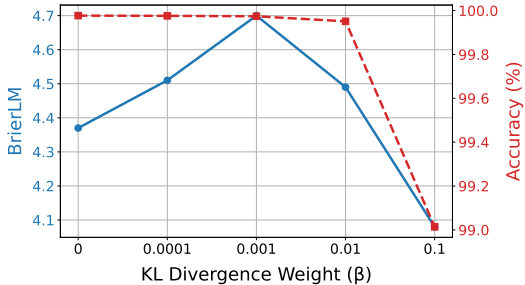
AE regularization	BrierLM
Reconstruction only	3.99
Ordinary KL	3.48
Clipped KL	4.13
Clipped KL + token dropout	4.55
Clipped KL + latent dropout	4.46
Clipped KL + both	**4.70**

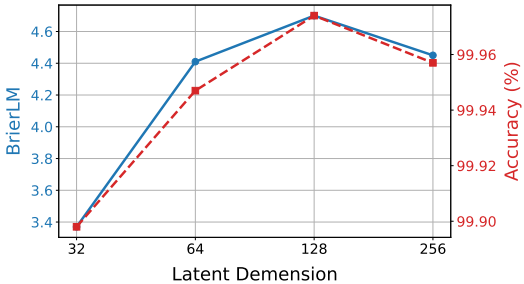
- Ordinary VAE: 71 of 128 dimensions collapsed.
- KL clipping restores useful capacity.
- Both dropout forms help.

Faithful reconstruction alone does not guarantee a learnable latent space.

Paper, Sec. 7.3, pp. 16–17; Table 2, p. 17. Fixed small LM, 50K steps.

Representation quality peaks at moderate regularization





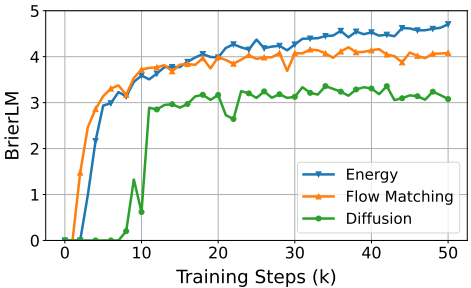
$\beta = 0.001$ works best here. At 0.1, reconstruction falls to about 99%.

$l = 128$ works best; dropout also changes with dimension: 0.05 to 0.20.

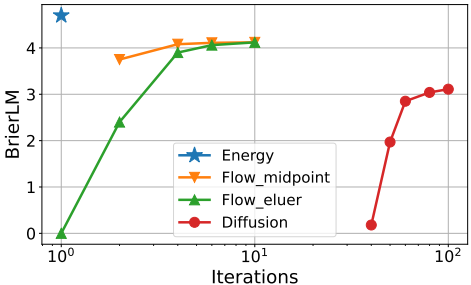
Blue/left: downstream BrierLM. Red/right: AE token reconstruction accuracy.
Larger AE depth, width, or training data produced no significant improvement.

Paper, Sec. 7.3, pp. 16–17; original Figs. 6–7, p. 17.

The energy head combines quality with one-pass sampling



Flow starts faster; the energy head reaches a higher final BrierLM.



Flow midpoint is useful at 2–4 iterations. Energy needs one forward pass.

Diffusion and flow also use 8 training replicas of each conditional state.
Solver iterations and network evaluations need not be identical.

Paper, Sec. 7.4, pp. 17–18; original Figs. 8–9, p. 18. Training-curve evaluation uses 100 iterative steps.

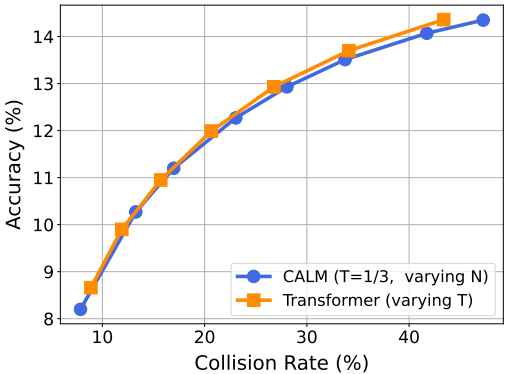
Discrete feedback is a major part of the result

Backbone input	Brier-1	Brier-4	BrierLM
Decoded discrete tokens	21.81	1.25	**4.70**
Continuous latent vector	17.43	0.73	3.25
Sum of both representations	21.17	1.12	4.40

- Discrete input gains 1.45 BrierLM points over continuous input.
- Authors' explanation: the backbone struggles to unpack the compact code.
- The AE decoder exposes useful token structure before the next backbone step.

Paper, Sec. 7.4 and Table 5, p. 19. Selected columns; Brier-n components shown on the paper's percent scale.

Temperature control consumes a candidate budget



- Match model checkpoints by BrierLM.
- CALM: fix $T = 1/3$, vary candidate count N.
- $N \approx 100$: behavior near AR temperature 0.6.
- $N \approx 200$: behavior near AR temperature 0.5.

Similar reference-match/collision trade-offs; extra head and decoder evaluations still cost compute.

Paper, Sec. 7.5, pp. 19–20; original Fig. 11, p. 20. Both axes aggregate n=1–4; collision is inverse diversity.

The evidence supports a compute trade-off with clear conditions

- **Compute:** reported FLOPs include AE overhead. Actual latency with temperature sampling remains to be measured.
- **Quality:** WikiText-103 short-continuation scores, with different raw-token context windows. BrierLM needs careful treatment of negative components.
- **Scaling:** $K = 4$ is useful at the tested sizes; larger K and joint scaling laws remain open.
- **Algorithms:** log-probability-based policy optimization and KL distillation require adaptation.

Paper, Secs. 7–8, pp. 14–21. Metric caveat: discussion analysis of Eq. (15) and official aggregation code.

Three ideas to take into the discussion

1. **Scale the prediction unit.** A larger K makes each backbone call account for more generated text.
2. **Learn a generatable representation.** Robustness and redundancy matter beyond AE reconstruction accuracy.
3. **Budget the entire sampling pipeline.** Fewer backbone steps coexist with head, decoder, and temperature-control costs.

CALM supplies an integrated prototype for blockwise continuous-output language modeling.

Shao, Li, Meng and Zhou. Continuous Autoregressive Language Models. arXiv:2510.27688v1 (2025).

Versioned paper Official code. Main talk ends here; backup slides follow.

Backup: more samples have diminishing returns

Model samples; $M = 100$

N	BrierLM	Cost
2	4.37	0.82×
4	4.53	0.91×
8	4.70	1.00×
12	4.72	1.13×

Target samples; $N = 8$

M	BrierLM	Cost
1	4.50	0.92×
16	4.56	0.94×
100	4.70	1.00×
200	4.67	1.07×

$N = 8 \rightarrow 12$: only $+0.02$ score for 13% extra cost.

$M = 100 \rightarrow 200$: score decreases; the table is not strictly monotonic in M.

Paper, Sec. 7.4 and Table 3, p. 18. Costs relative to N=8, M=100.

Backup: squared energy distance only identifies a mean

$$S_2(P, \mathbf{y}) = \mathbb{E}\|X - X'\|^2 - 2\mathbb{E}\|X - \mathbf{y}\|^2$$
$$= -2\|\mathbb{E}X - \mathbf{y}\|^2.$$

$$\mathbb{E}_Q S_2(P, Y) = -2\|\mathbb{E}_P X - \mathbb{E}_Q Y\|^2 - 2\,\mathrm{tr}\,\mathrm{Cov}_Q(Y).$$

α	0.75	1	1.25	1.5	1.75	2
BrierLM	Fail	4.70	4.42	4.46	4.30	0

- $\alpha < 1$: authors report unstable gradients.
- $\alpha = 2$: many distributions have the same optimal mean.

Paper, Eq. (9), p. 6; Table 4, p. 19. Equations expanded here for teaching; finite second moments assumed.

Backup: exact temperature sampling cost

Write $1/T = n + \alpha$, with $0 \le \alpha < 1$.

$$\mathbb{E}[N_{\text{total}}] = \frac{n + \mathbf{1}(\alpha > 0) \sum_x P(x)^{1/T - 1}}{Z_T}.$$

- Stage 1 always costs n draws; candidate x occurs with mass $P(x)^n$.
- Stage 2, given x, costs $P(x)^{\alpha - 1}$ draws on average.
- Divide expected per-trial cost by the success probability Z_T.

$$\mathbb{E}[N_{\text{total}}] \le \begin{cases} (n+1)/Z_T, & 0 < T \le 1/2, \\ (1 + |\mathcal{X}|^{2 - 1/T})/Z_T, & 1/2 < T < 1. \end{cases}$$

Paper, Theorem 2 and Corollary 2.1, p. 11; Appendix A.2, pp. 31–32. T=1 uses the base sampler directly.

Backup: the composite score needs a restricted domain

Discussion counterexample to an unrestricted product formula

True continuation: always AAAA. Model continuation: always BBBB.

For every $n = 1, 2, 3, 4$: $P(y) = 0$ and $\sum_x P(x)^2 = 1$, so $B_n = -1$.

Direct multiplication gives $100\,((-1)^4)^{1/4} = 100$.

- The geometric mean normally requires nonnegative components.
- Official aggregation clamps the *product*; even numbers of negative components remain possible.
- Positive population components retain the true optimum, but nonlinear plug-in estimates are not generally unbiased.

Analysis of paper Eq. (15), p. 9; this is not a claim that the reported positive-component results exhibit the failure.

Official code, commit 0249935: train/train_calm.py, lines 566–574.

Backup: why the batch approximation is consistent

For a fixed finite block space, counts satisfy $C_x/N \to P(x)$.

$$\frac{\binom{C_x}{n}}{N^n} = \frac{1}{n!}\prod_{j=0}^{n-1}\frac{C_x - j}{N} \longrightarrow \frac{P(x)^n}{n!}.$$

$$\frac{\binom{C_x}{n}}{\sum_z \binom{C_z}{n}} \longrightarrow \frac{P(x)^n}{\sum_z P(z)^n}.$$

- A positive-probability outcome eventually appears at least n times; fallback probability vanishes.
- Normalized weights are bounded in $[0, 1]$, giving convergence of their expectations.
- Neither step makes the finite-N procedure exact.

Paper, Theorem 3 and Appendix A.3, p. 33. Vanishing-fallback argument supplied in this discussion.

Backup: block temperature differs from token temperature

CALM's exact target at one block boundary is

$$P_T(x_{1:K} \mid h) \propto \left[\prod_{j=1}^{K} p(x_j \mid h, x_{<j}) \right]^{1/T}.$$

Scaling each token's logits instead gives

$$\prod_{j=1}^{K} \frac{p(x_j \mid h, x_{<j})^{1/T}}{\sum_v p(v \mid h, x_{<j})^{1/T}}.$$

- The second expression has normalization factors that depend on the generated prefix.
- Similar accuracy–collision curves do not establish identical distributions.
- At $K = 8$, the public Brier evaluator directly scores only the first four positions of each block.

Discussion derivation from Sec. 5; comparison in Fig. 11, p. 20. Evaluation detail: official modeling_calm.py, commit 0249935.