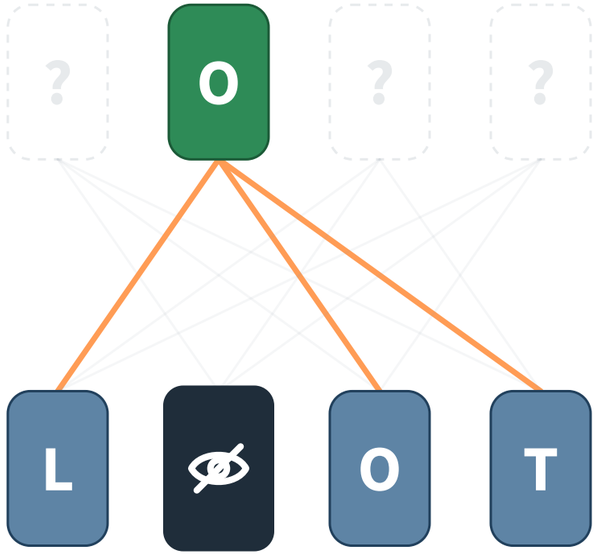
Empowering Masked Diffusion Models to Self-Correct with Leave-One-Out Transformers

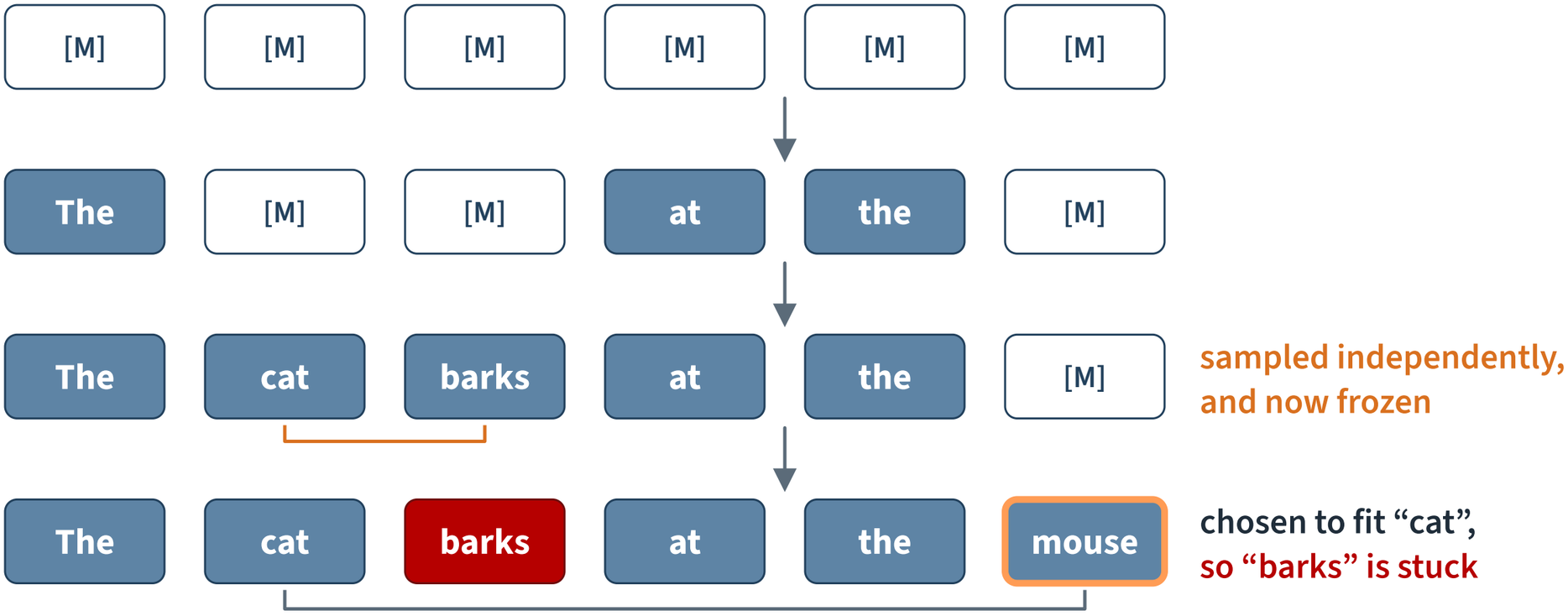
NeurIPS 2026

Sofian Zalouk, Vincent Counathe, Paul Jünger, Daniel Cao, Daniel Zlotnick, Kilian Q. Weinberger, Christopher De Sa

Cornell University

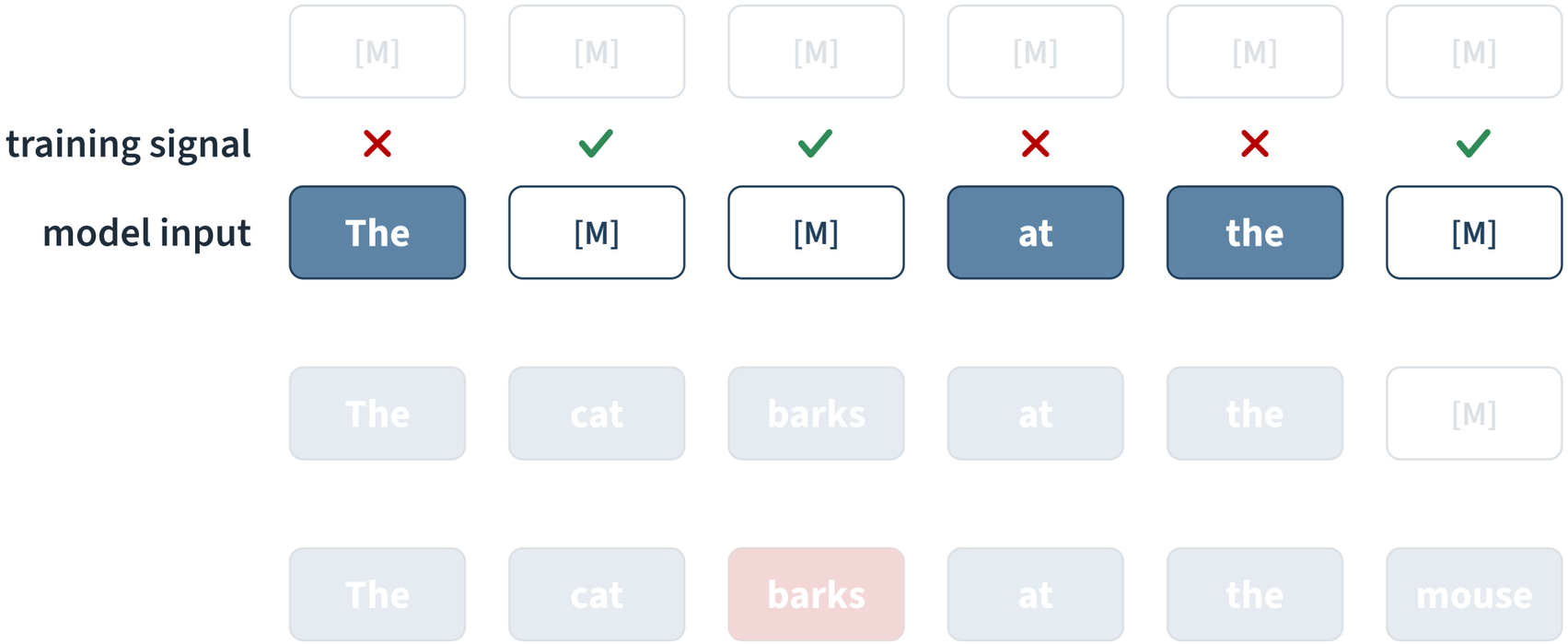


Masked Diffusion Models Cannot Self-Correct



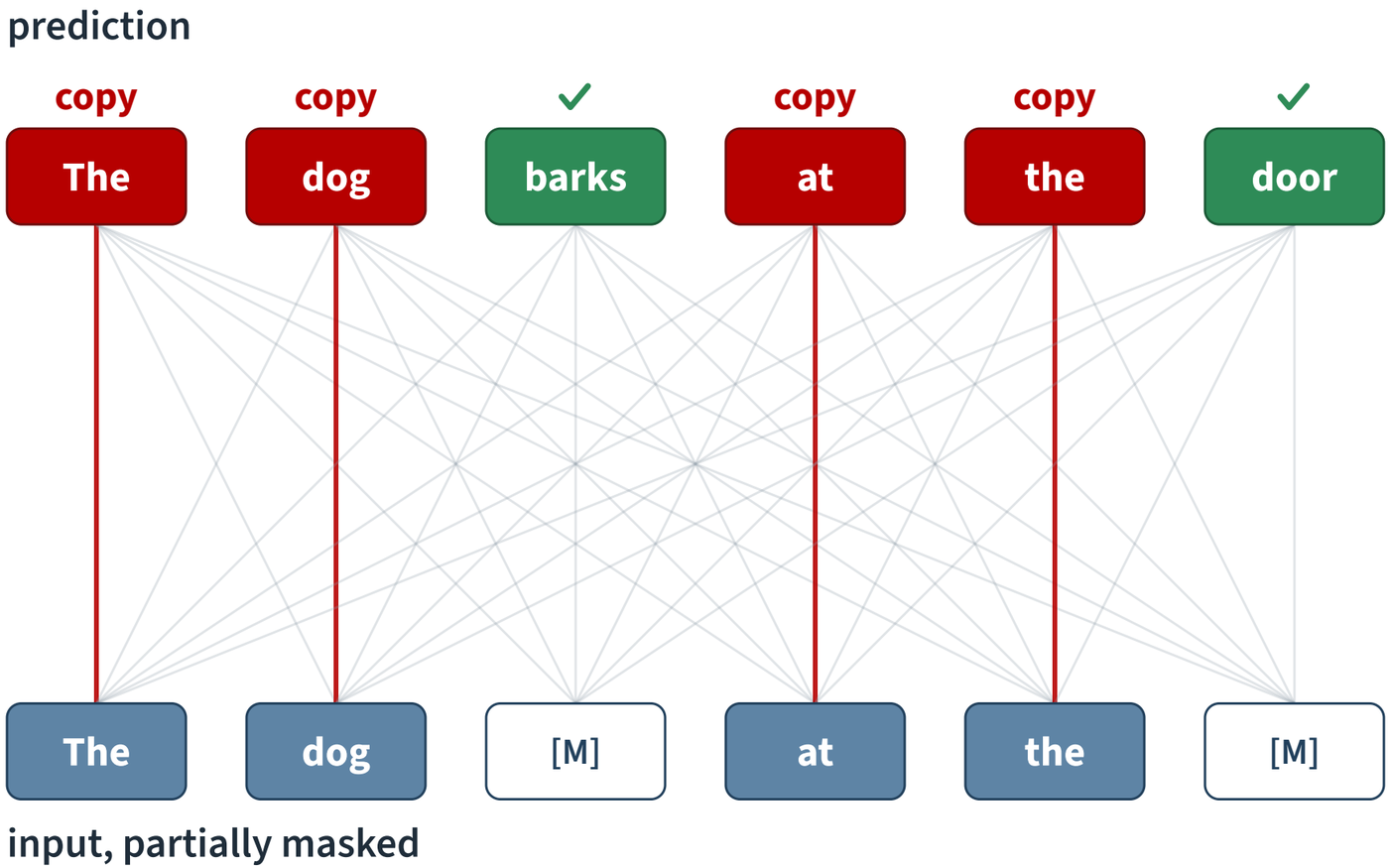
Mistakes propagate through the generation process.

Sparse Training Signal



Only masked tokens contribute to the loss, so the model never learns to revise a visible token.

Idea: Leave-One-Out Prediction

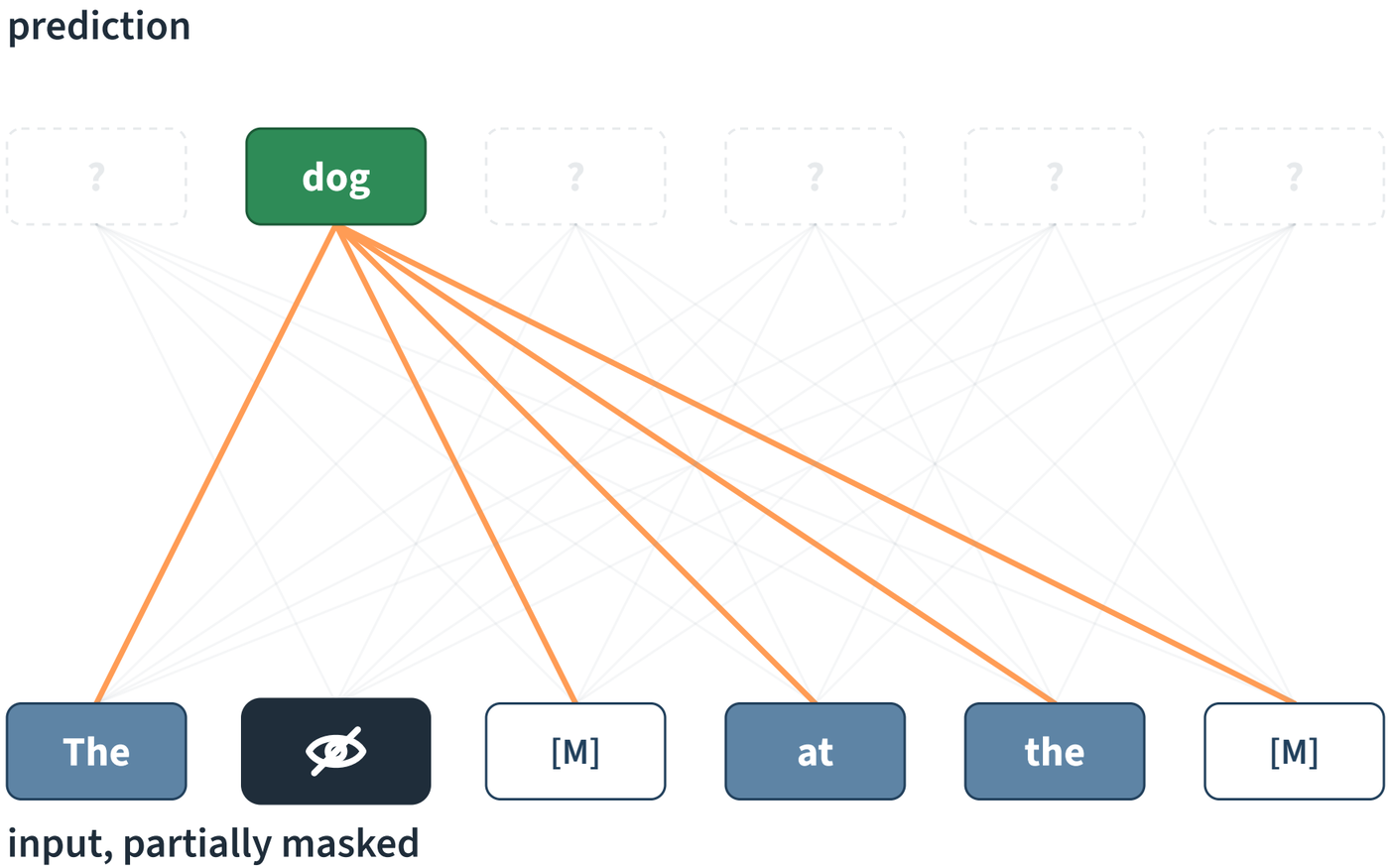


Visible tokens are just copied

Output i sees input i, so there is nothing to learn.

Why not train on the visible tokens too? In a bidirectional transformer, every output can see its own input.

Idea: Leave-One-Out Prediction



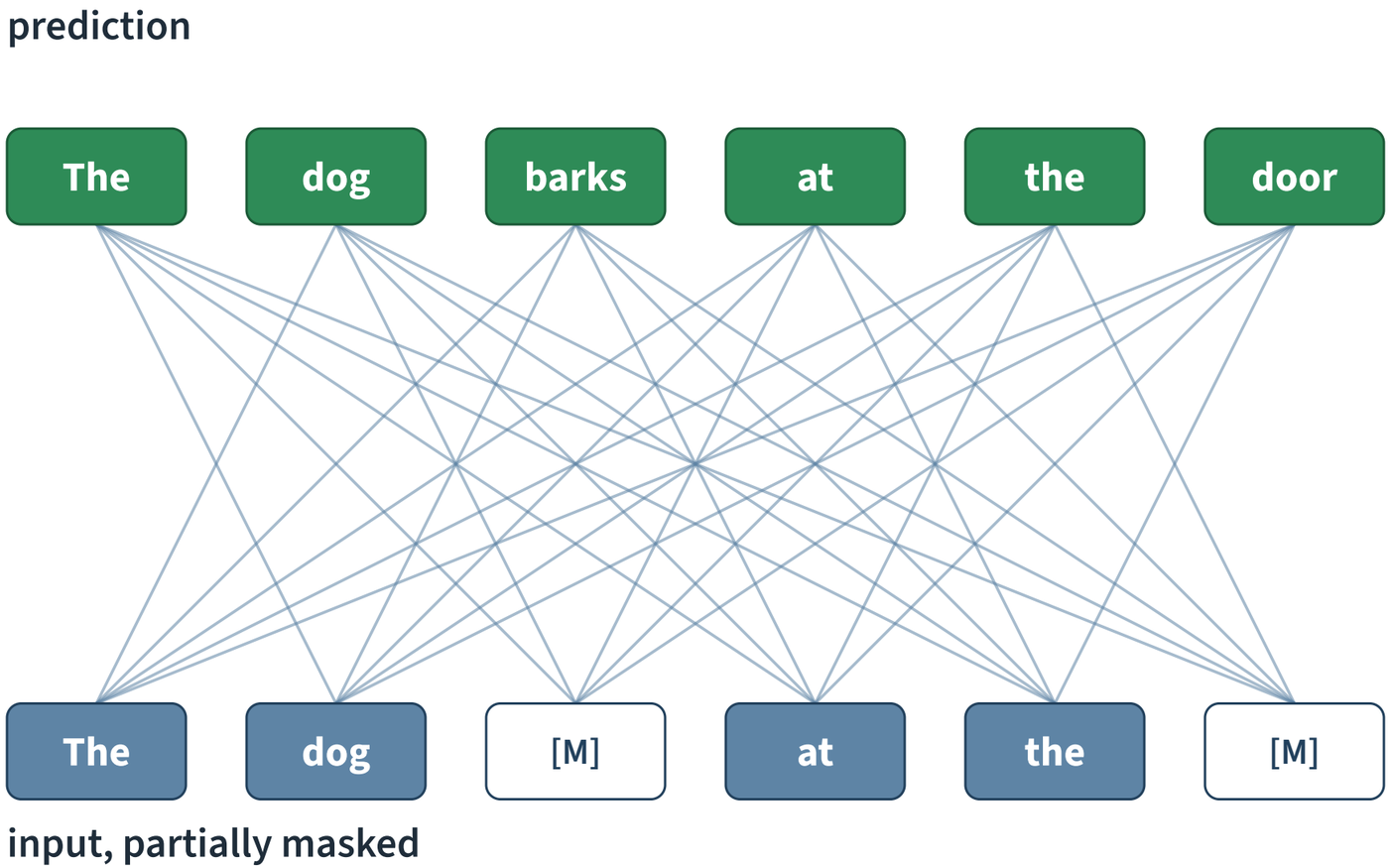
Leave-one-out conditional

$$p_\theta\left(x_i \mid \mathbf{x}_{-i}\right)$$

Hide input i, then predict it from the rest.

Hide one token, and predict it from the surrounding context.

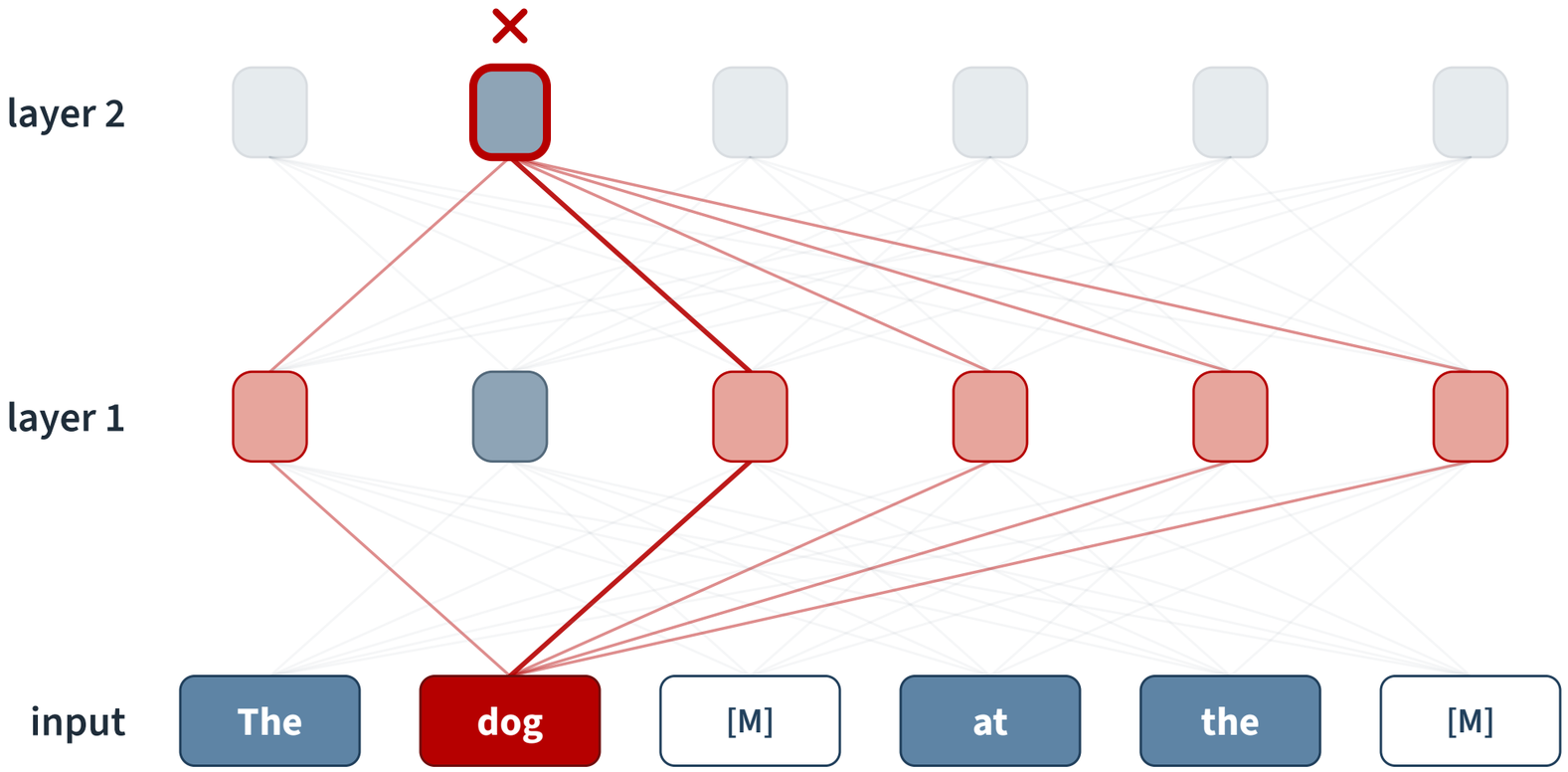
Idea: Leave-One-Out Prediction



- **Training:** a signal at every position
- **Sampling:** re-check every token

That gives a training signal at every token, and a way to revise tokens while sampling.

Challenge: Information Leakage

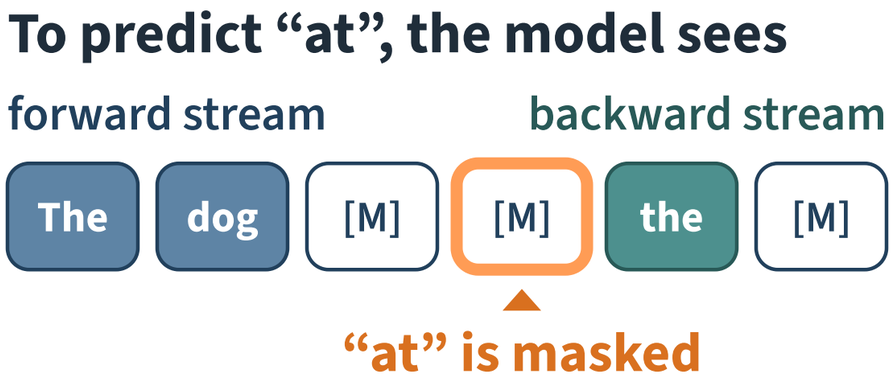


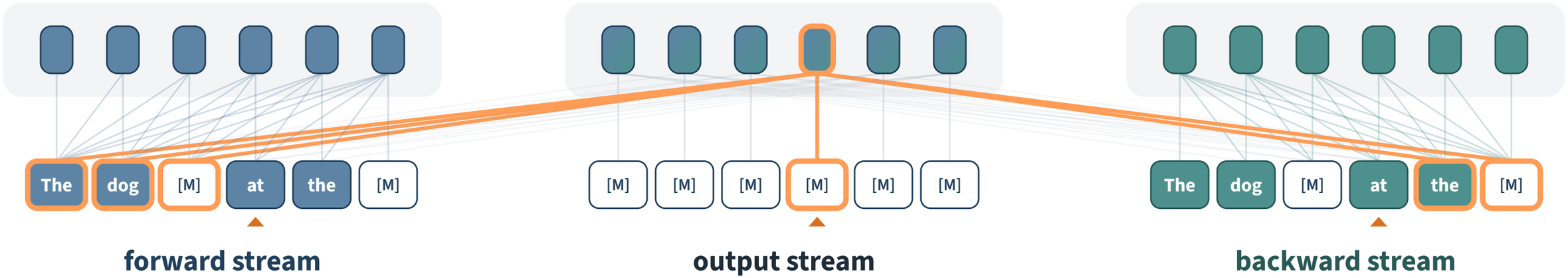
Two layers: leak

“dog” reaches position 2 through its neighbours.

With two layers it breaks: the model can see the input.

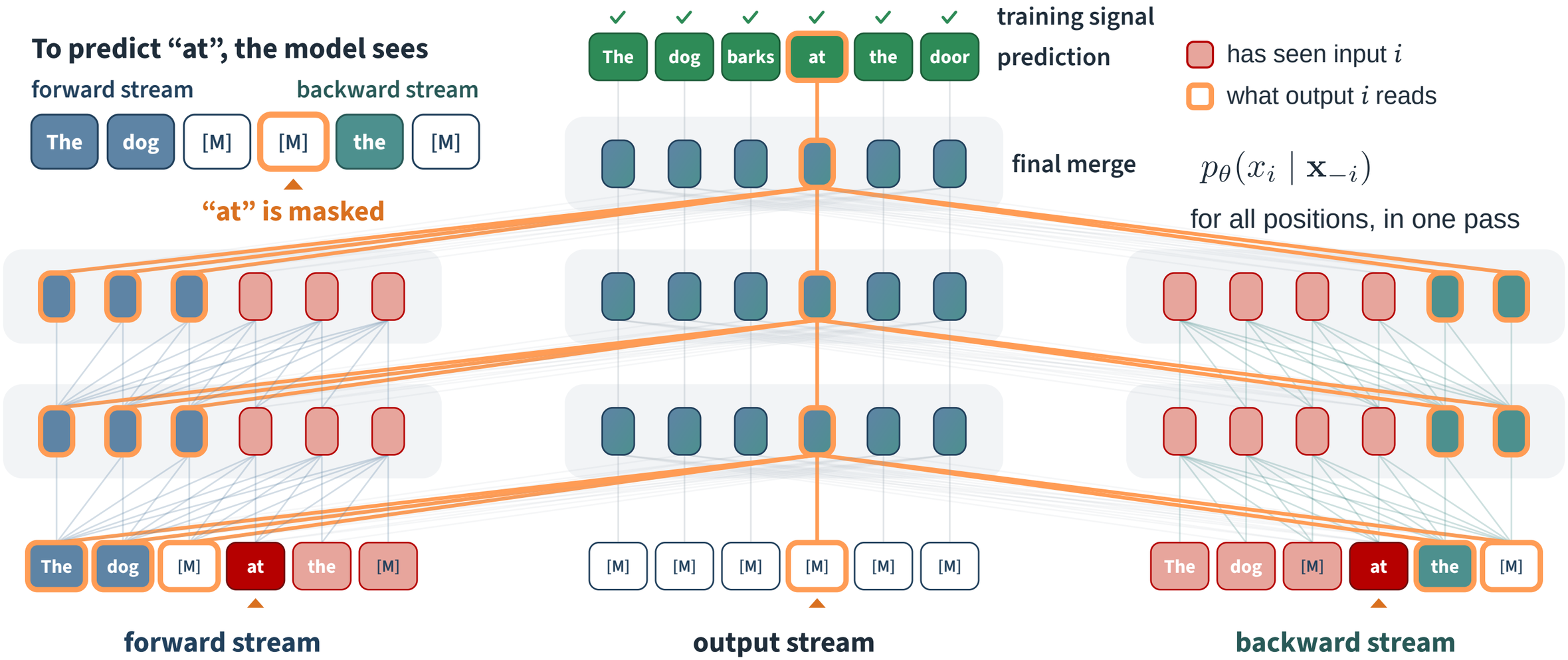
Leave-One-Out Transformer (LOOT) Architecture





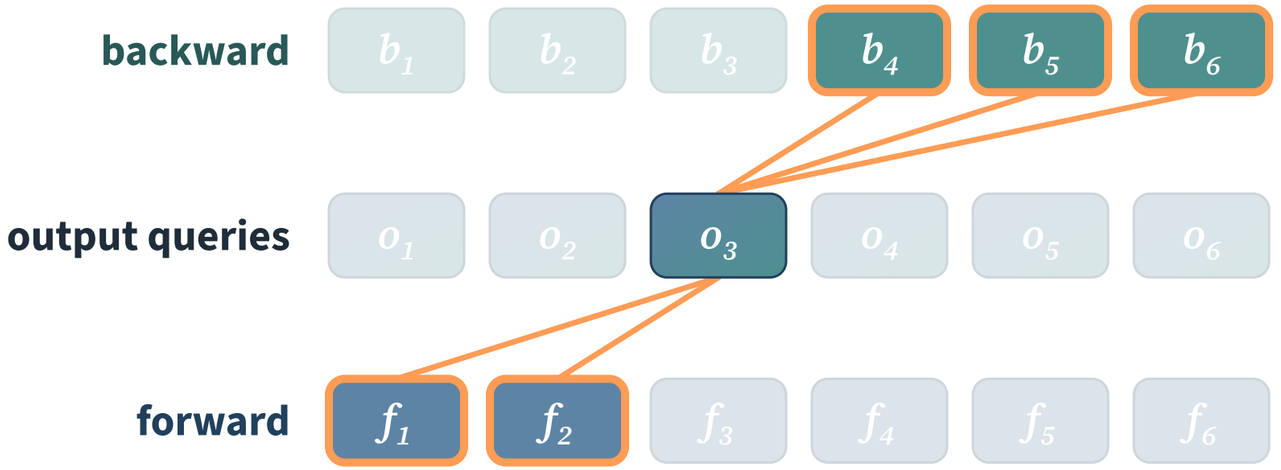
Put the three streams together: to predict “at”, the model sees every input except its own. From its point of view, “at” is masked.

Leave-One-Out Transformer (LOOT) Architecture



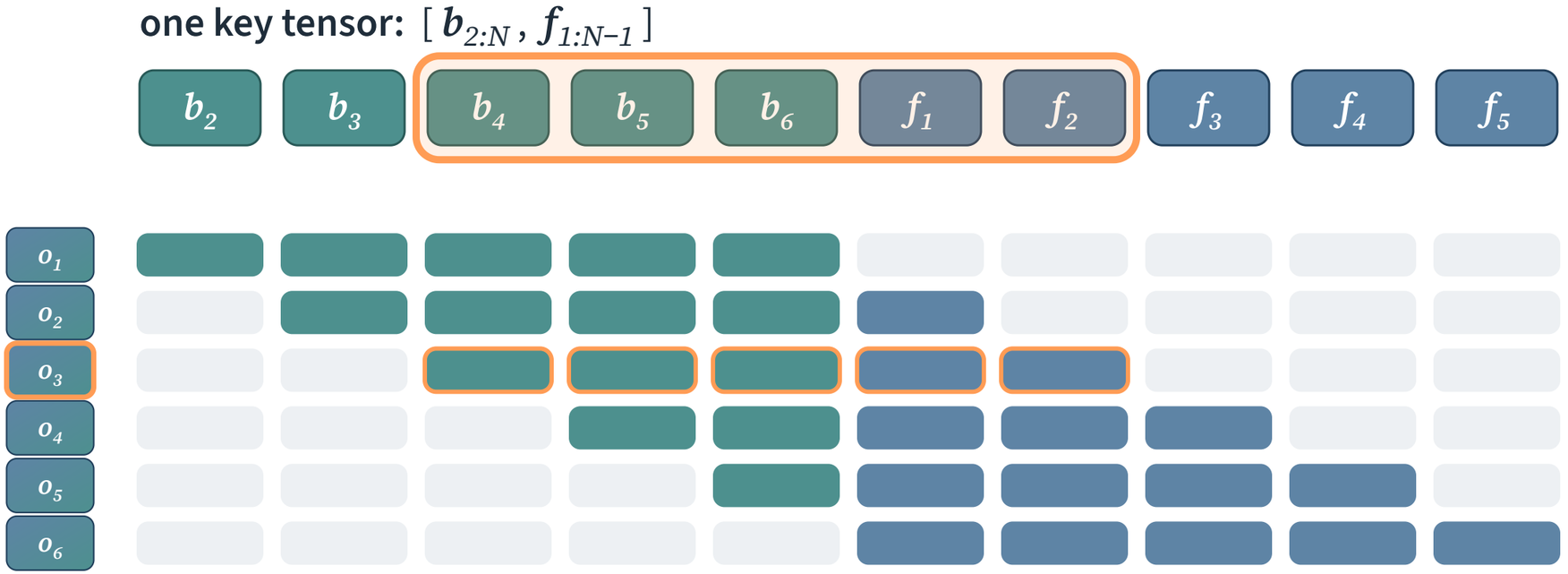
One forward pass predicts every token, masked or visible, without seeing its own input. So every position can be supervised.

Efficient Merging with Sliding-Window Attention



Each output query attends to two separate sets: backward states to its right, forward states to its left.

Efficient Merging with Sliding-Window Attention



- **2 FlashAttention calls** per layer
- **Shared weights:** same parameter count as a standard MDM
- **2.24×** the FLOPs of a standard MDM pass

Ordinary sliding-window attention: no custom kernels.

LOOT Training Objective

Standard MDM

$$\mathcal{L}_{\text{MDM}} = \mathbb{E}_{t,\mathbf{x},M_t}\left[\frac{\alpha'_t}{1-\alpha_t}\sum_{i\in M_t}\log p_\theta\left(x_i \mid \mathbf{x}[M_t]\right)\right]$$



2 of 6 scored

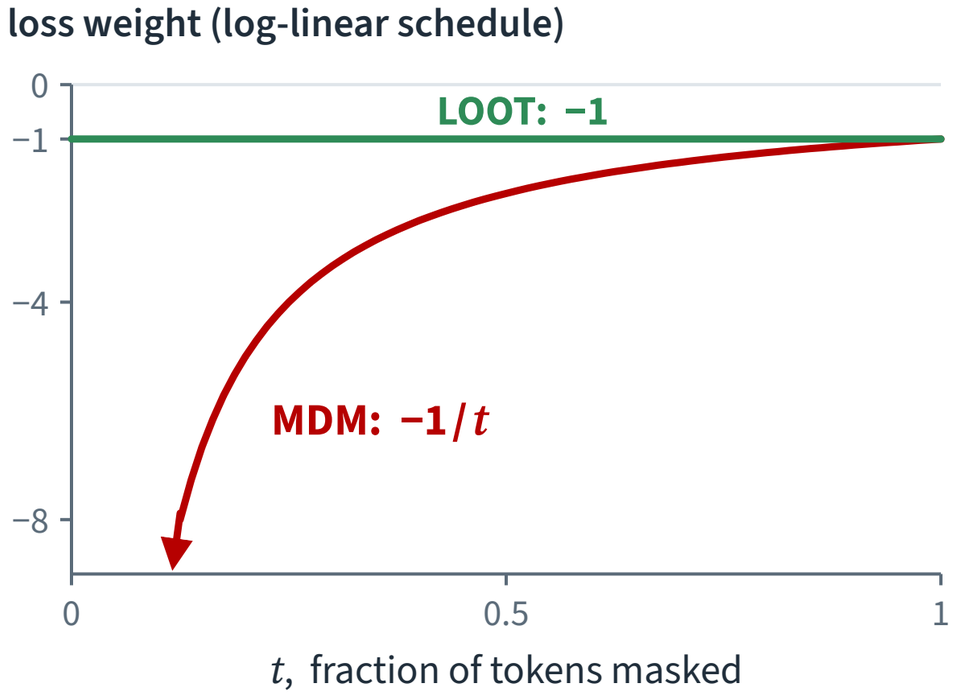
= in expectation (leave-one-out property)

LOOT

$$\mathcal{L}_{\text{LOOT}} = \mathbb{E}_{t,\mathbf{x},M_t}\left[\alpha'_t\sum_{i=1}^{N}\log p_\theta\left(x_i \mid \mathbf{x}[M_t]\right)\right]$$



6 of 6 scored

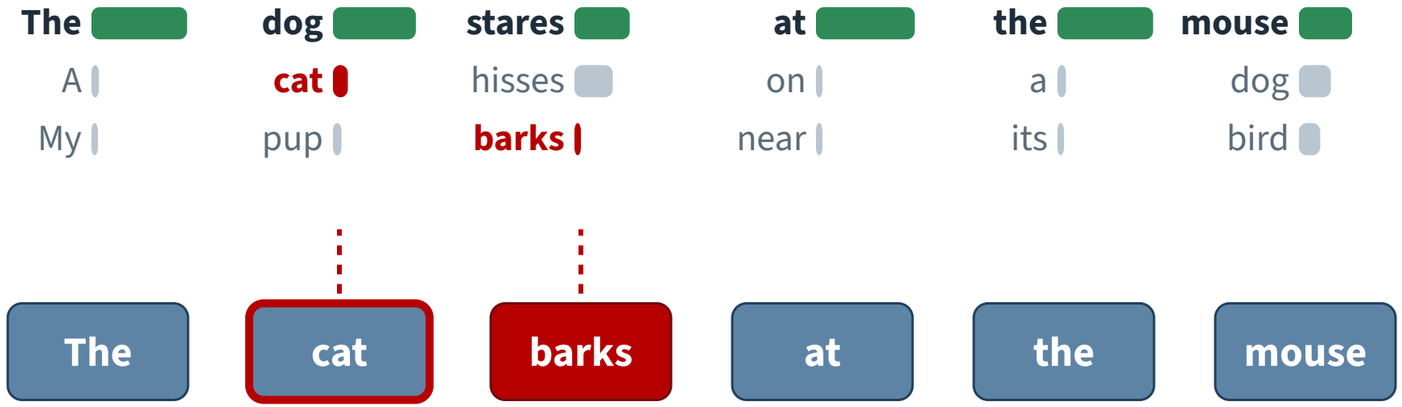


- **−3.1 PPL** vs. half-and-half loss
- **−2.8 PPL** vs. standard MDM loss

Same bound, lower-variance estimate, and about 3 points better perplexity in our ablations.

Self-Correction with Leave-One-Out Conditionals

Leave-one-out conditionals from one forward pass (illustrative)

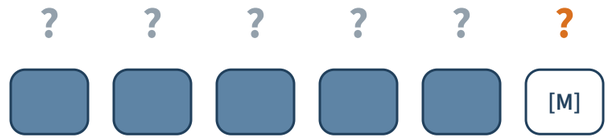


LOOT predicts replacements for every token.

Self-Correction with Leave-One-Out Conditionals

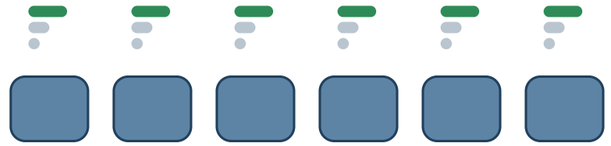


ReMDM



has to guess what to remask

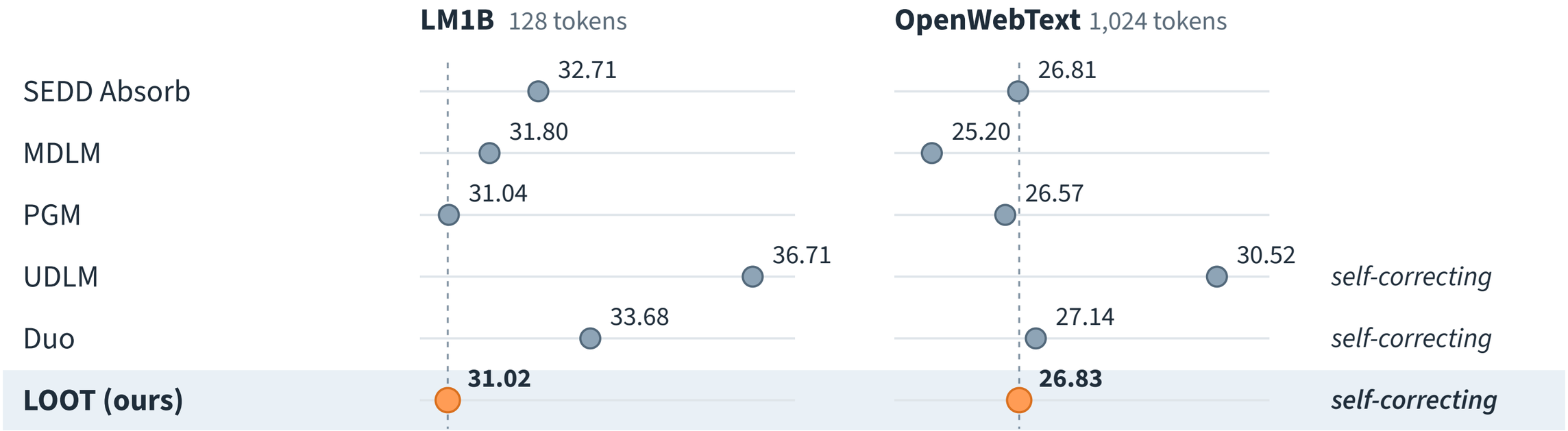
LOOT



a fresh conditional
for every token

Remasking samplers must choose which tokens to remask, using heuristics instead of up-to-date conditionals. LOOT gets those conditionals for every token, at every step.

Results: Language Modeling



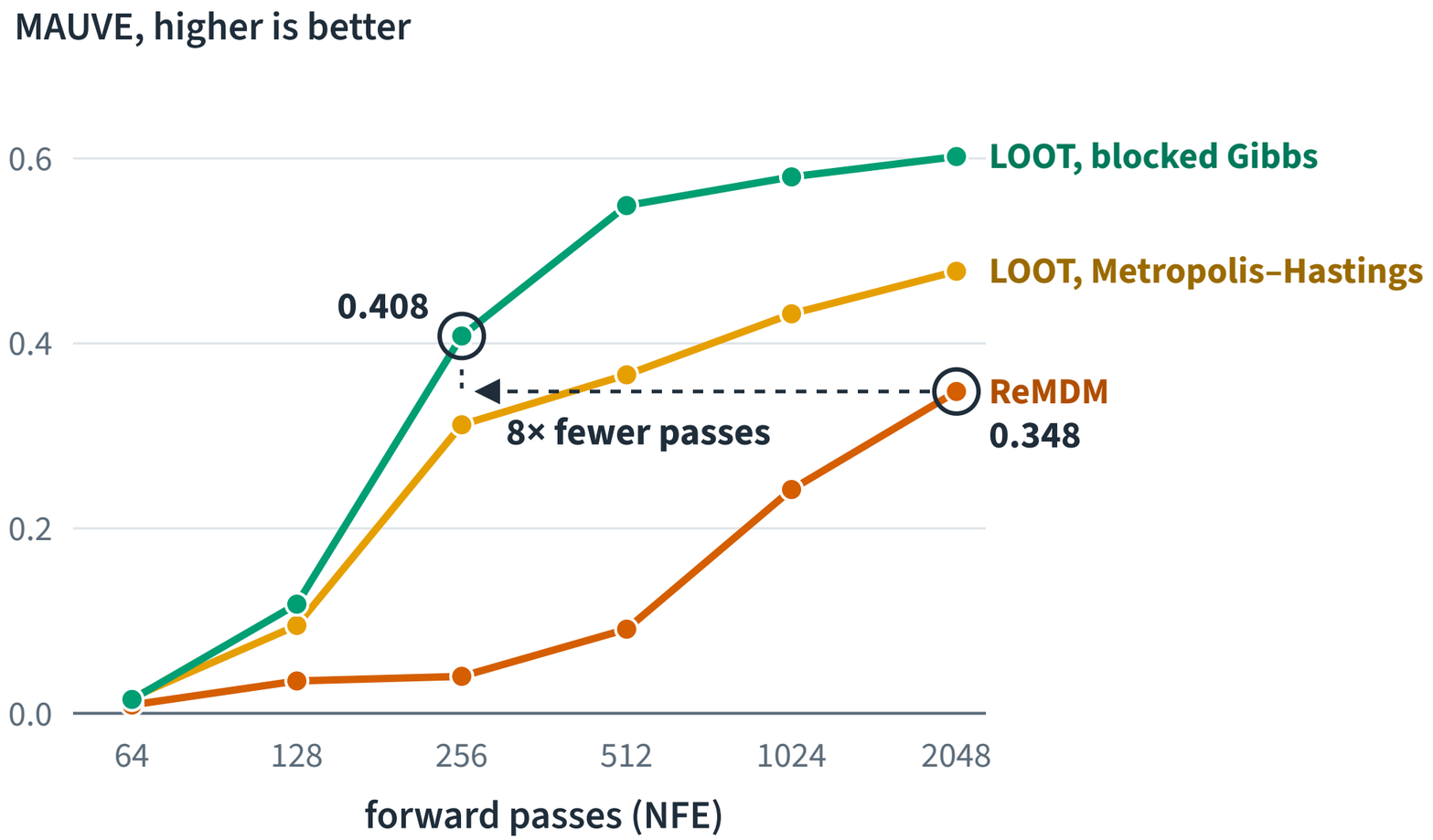
Validation perplexity, lower is better

- **Merging at every layer:** 41.1 → 36.2 PPL
- **LOOT objective:** 39.4 → 36.2 PPL

LM1B ablation, 200K steps

Both contributions close the gap: merging at every layer, and the new objective.

Results: Sampling and Self-Correction



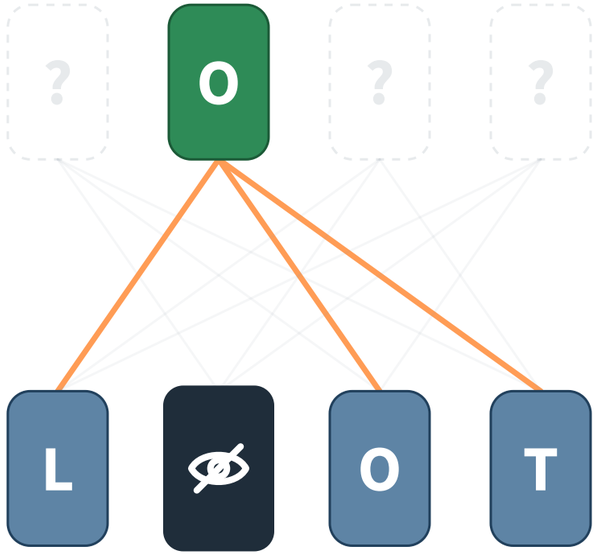
LOOT at 256 NFE
vs. ReMDM at 2,048 NFE

- **8×** fewer forward passes
- **3.5×** fewer FLOPs
- MAUVE **0.408** vs. 0.348
- Gen. PPL **20.1** vs. 31.2

Blocked Gibbs at 256 passes beats ReMDM at 2,048: higher MAUVE, and lower generative perplexity at the same entropy.

Summary

- **Architecture.** Three streams, merged at every layer, give all N leave-one-out predictions in one forward pass, at about 2.2× the cost.
- **Objective.** Every position is supervised with constant weights: the same NELBO, estimated with lower variance.
- **Sampling.** Gibbs-style self-correction beats remasking with 8× fewer forward passes.



Paper and code: github.com/szalouk/loot-diffusion