From Prediction to Generation

David I. Inouye

Dynamic Content · Last Updated: October 30, 2026

A next-token model becomes a generator through repeated feedback

Factorize → choose a token → append → repeat → evaluate.

We have an email-prefix model. Now we must choose a continuation and decide when to stop.

A sequence probability is a product of conditional probabilities

For a fixed prompt c and continuation tokens $w_1, \dots, w_L$:

$$p_\theta(w_{1:L} \mid c) = \prod_{\ell=1}^{L} p_\theta(w_\ell \mid c, w_{<\ell}).$$

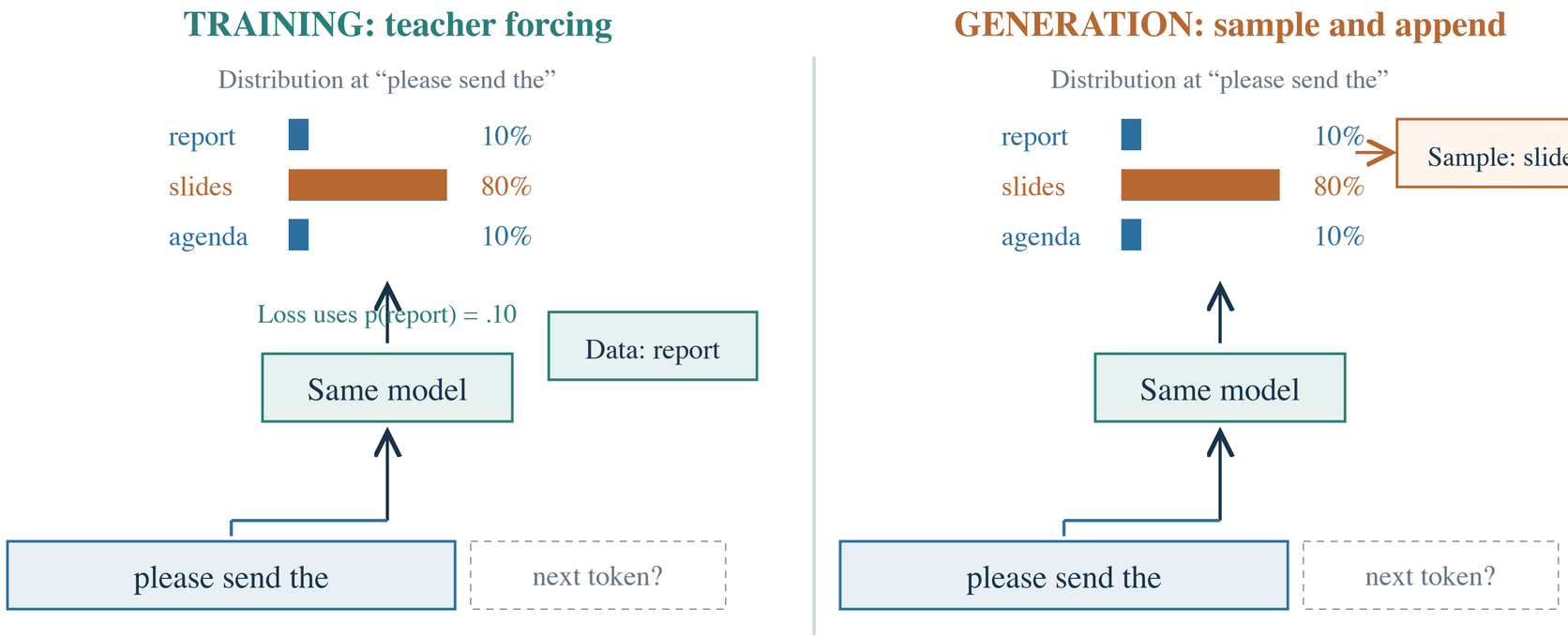
Synthetic token example:

$$p_\theta(\text{report, today, EOS} \mid \text{please send the})$$

$$= p_\theta(\text{report} \mid c)\; p_\theta(\text{today} \mid c, \text{report})\; p_\theta(\text{EOS} \mid c, \text{report,today}).$$

Taking **negative logs** turns the product into the sum used in next-token training.

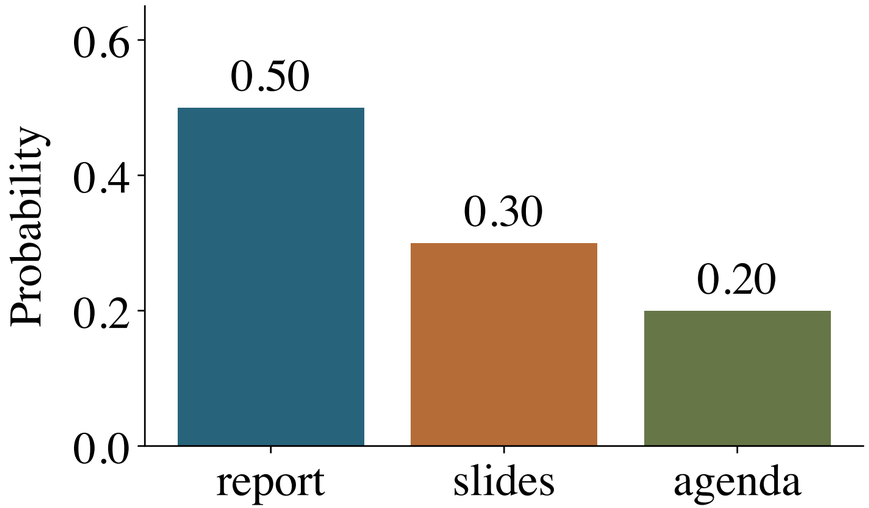
Teacher forcing uses the true prefix—even when the model is wrong

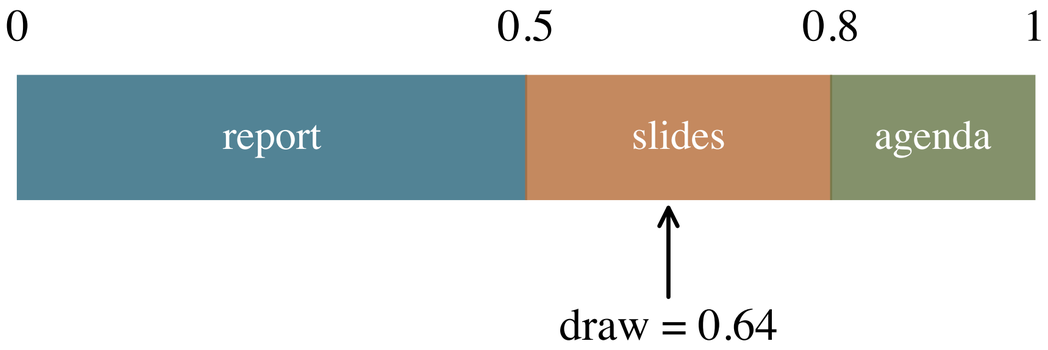


Advance: append the observed token in training, or the sampled token in generation.

Training prefixes are all known in advance → a causal transformer can process them together.

Greedy choice takes the largest bar; sampling uses all the bars



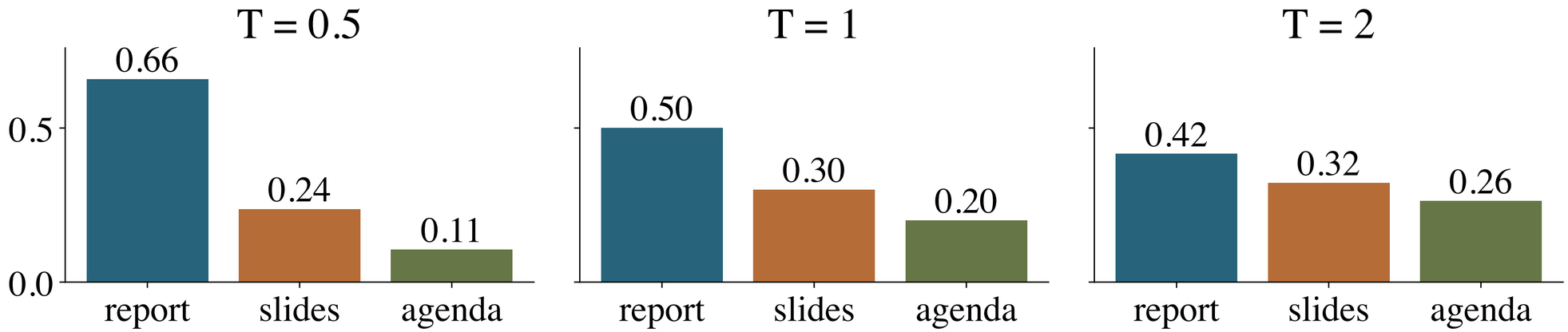


Greedy: choose $\arg\max_j p_j$. **Categorical sampling:** choose token j with probability p_j.

Temperature changes how concentrated the choices are

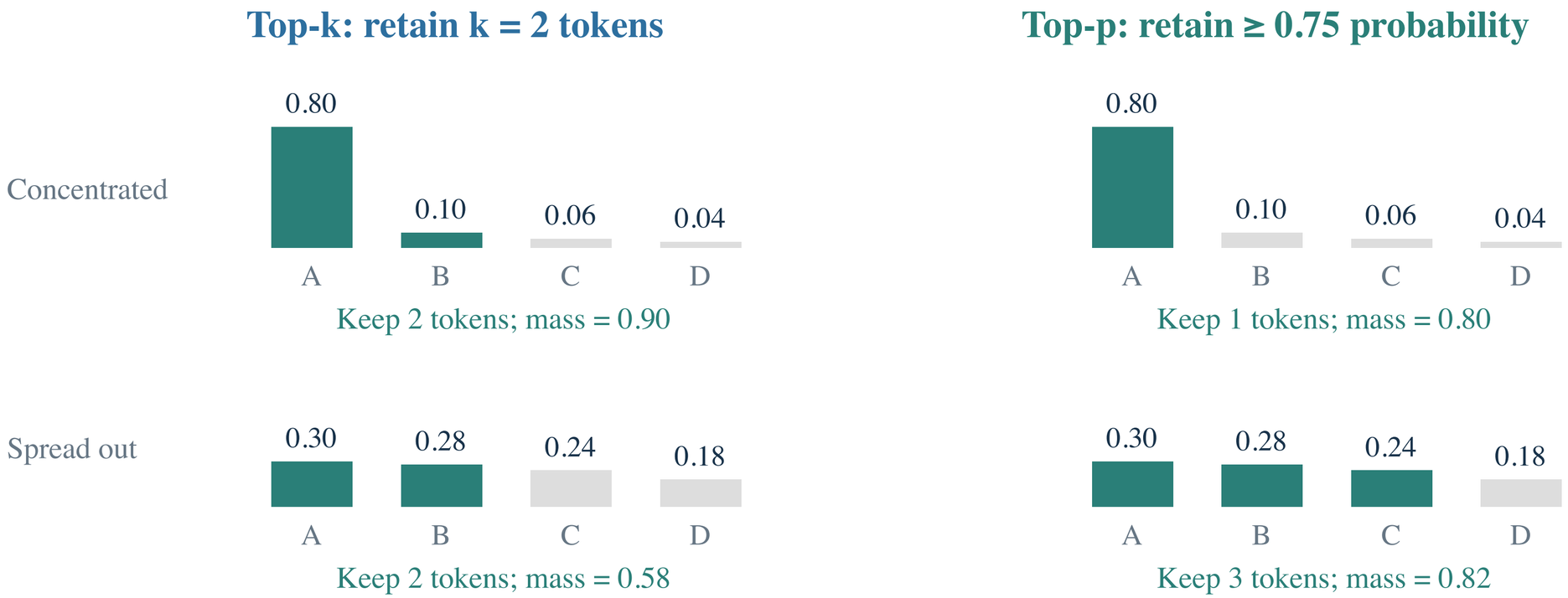
For the vocabulary-logit column z and temperature $T > 0$:

$$p_{T,j} = \frac{\exp(z_j/T)}{\sum_{k=1}^{v} \exp(z_k/T)}.$$



Same model and prefix. Different sampling distribution.

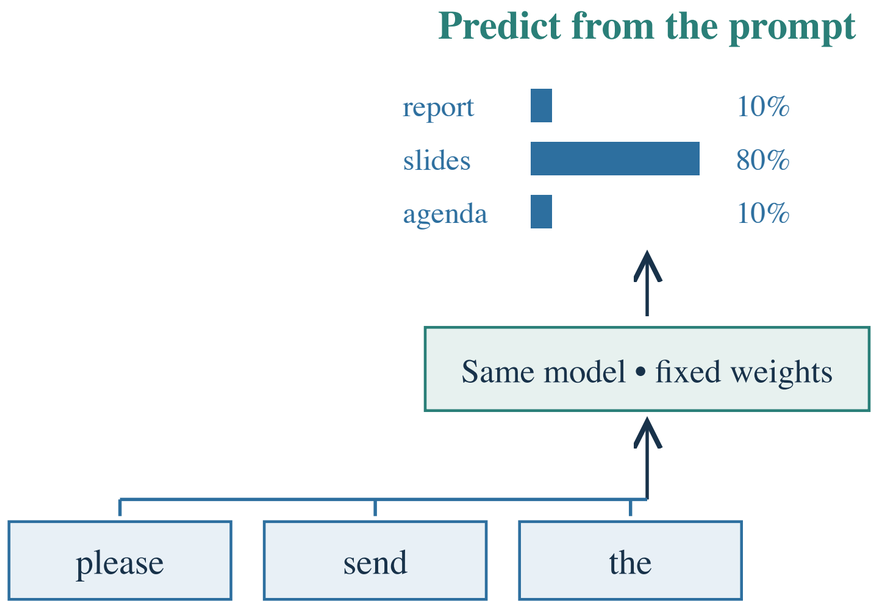
Top-k fixes the count; top-p sets a mass threshold



Keep the selected tokens → renormalize their probabilities → sample.

Top-p (*nucleus sampling*) keeps the smallest highest-probability set reaching the threshold.

Generation samples, appends, and runs the model again



One prefix in one place: it grows only by adding the sampled token at the right.

Stop on **EOS**, a **new-token cap**, or the model's **context limit**.

Can we change how a trained model generates?

> **We switch a trained model from greedy decoding to top-p sampling.**
> **What must change?**
> **A. The model's learned weights**
> **B. The rule for choosing the next token**
> **C. The causal attention mask**

B. No retraining needed. We sample from the retained tokens instead of always taking the largest probability.

The same trained model supports different generation choices

Choice	How it chooses the next token	Main control
Greedy	Take the largest probability	Deterministic local choice
Categorical sampling	Draw from the full distribution	Random variation
Temperature sampling	Rescale logits by $1/T$, then sample	Concentration
Top-k sampling	Keep k highest-probability tokens, renormalize, sample	Fixed candidate count
Top-p sampling	Keep the smallest set reaching mass p, renormalize, sample	Adaptive candidate count

Inference choices: same weights and training; different generated outputs.

Evaluate the **model + decoding rule** for the task. Temperature and filters can be combined.