Does inference-time reasoning lead to more creative problem-solving?

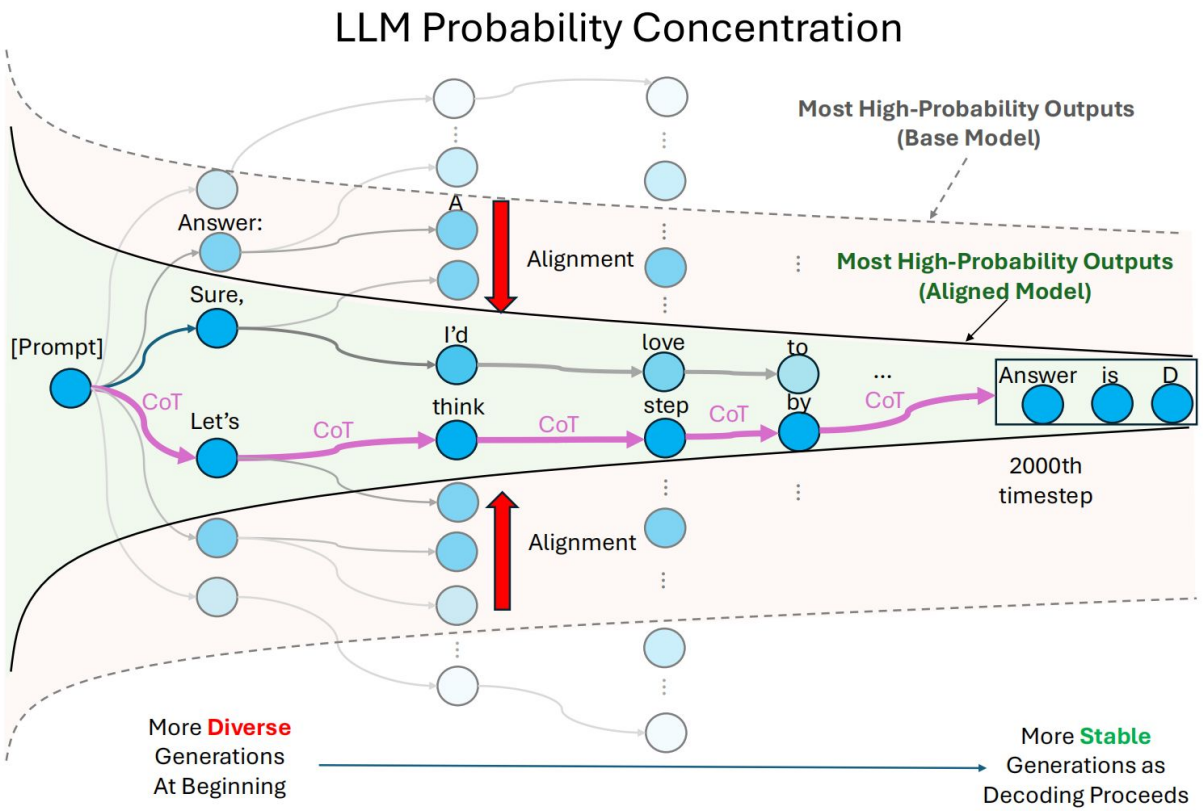
- How Alignment Shrinks the Generative Horizon
- Roll the dice & look before you leap: Going beyond the creative limits of next-token prediction

Tom and Matt

How Alignment Shrinks the Generative Horizon

https://arxiv.org/abs/2506.17871

Big Picture


LLM Probability Concentration
Most High-Probability Outputs
(Base Model)
Most High-Probability Outputs
(Aligned Model)
Answer:
A
Alignment
Sure,
[Prompt]
I'd
love
to
...
Answer
is
D
CoT
Let's
CoT
think
CoT
step
CoT
by
CoT
2000th
timestep
Alignment
More Diverse
Generations
At Beginning
More Stable
Generations as
Decoding Proceeds

Some quick Math to set the stage

Branching Factor

$$\mathrm{PPL}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\}$$

- Measuring the output diversity of the model.

$$B(x;\theta) = \exp\left(\bar{H}(\mathrm{Y}_{1:N}|x;\theta)\right)$$

Avg Conditional entropy:

$$\bar{H}(\mathrm{Y}_{1:N}|x;\theta) = \frac{1}{N}\tilde{H}\left(\mathrm{Y}_{1:N}|x;\theta\right)$$

Prefix-level Entropy: (expected entropy of the last token

$$\tilde{H}\left(\mathrm{Y}_t|[x,\mathrm{Y}_{1:t-1}];\theta\right) = \mathbb{E}_{y_{1:t-1}}\tilde{H}\left(\mathrm{Y}_t|[x,y_{1:t-1}];\theta\right)$$

Token-level Entropy:

$$\tilde{H}\left(\mathrm{Y}_t|[x,y_{1:t-1}];\theta\right) = -\sum_{y_t}\tilde{P}\left(y_t|[x,y_{1:t-1}];\theta\right)\log\tilde{P}\left(y_t|[x,y_{1:t-1}];\theta\right)$$

- Questions: What's the different with perplexity? When are they (approx.) equal?

“

How to estimate BF?

Short outputs? Long outputs?

Branching Factor - Estimation

- Short outputs: **MC sampling**
- For short outputs, where it's tractable to sample sufficiently many sequences to closely estimate the conditional entropy at each position, we can estimate the BF by computing the conditional entropy at each position and then aggregating as:

$$B(x;\theta) \approx \exp\left(\frac{1}{M}\sum_{i=1}^{M}\frac{\sum_{t=1}^{|y^{(i)}|}\tilde{H}(\mathrm{Y}_t|[x, y^{(i)}_{1:t-1}];\theta)}{|y^{(i)}|}\right)$$

Entropy

Per token

Sentence length

M samples

Branching Factor - Estimation

$$\mathrm{PPL}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\}$$

- What about long output - (Weak) **Law of large number** to the rescue.
- When LLMs generate sufficiently long outputs, the average log-probability of each output sequence will be roughly the same, and can approximate average output entropy well, following the Asymptotic Equipartition Property (AEP)

Avg Conditional entropy are sum of small values: $\bar{H}(\mathrm{Y}_{1:N}|x;\theta) = \frac{1}{N}\tilde{H}\left(\mathrm{Y}_{1:N}|x;\theta\right)$

Theorem 4.1 (AEP for LLMs) *Given* $0 < \epsilon < 1$, *we have:*

$$\lim_{N\to\infty} P\left(\left|-\frac{1}{N}\log\tilde{P}\left(y_{1:N}|x;\theta\right) - \bar{H}\left(\mathrm{Y}_{1:N}|x;\theta\right)\right| < \epsilon\right) = 1$$

Concentration Argument, convergent i.p

Want a.s. Conv aka strong LLN? => Need i.i.d. or stationary ergodic Condition, which LLM don't give

LLM Sequence Probability (per token)

Per token entropy

Experiments & Results

BF decrease over decoding time

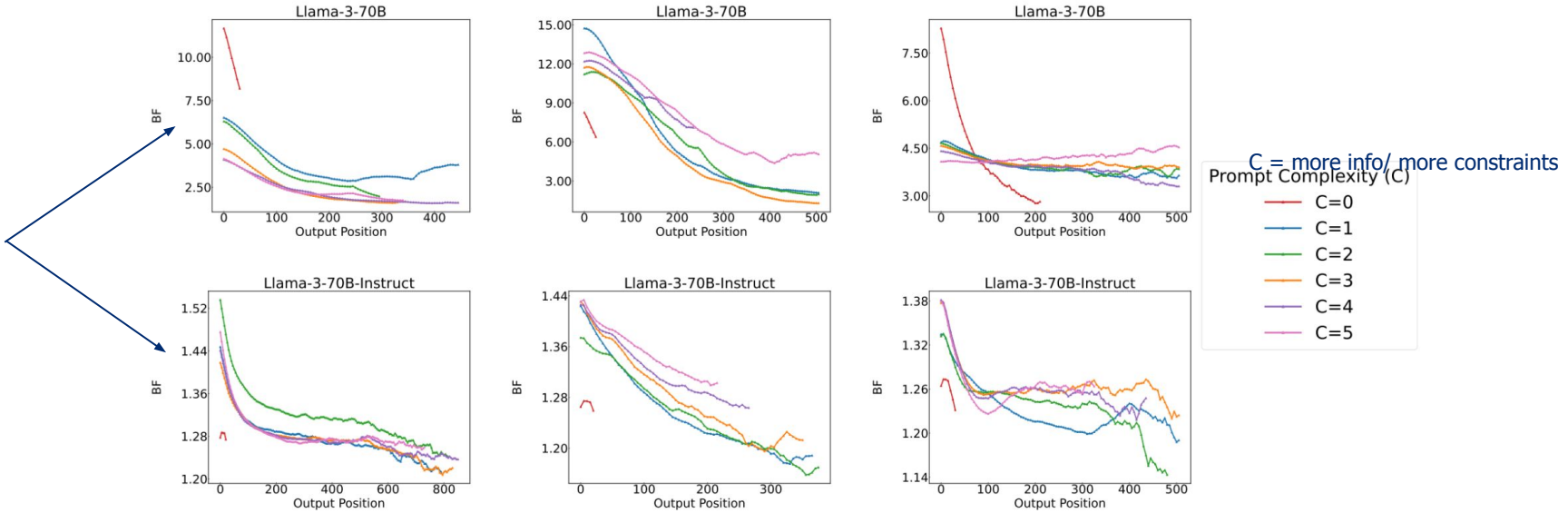


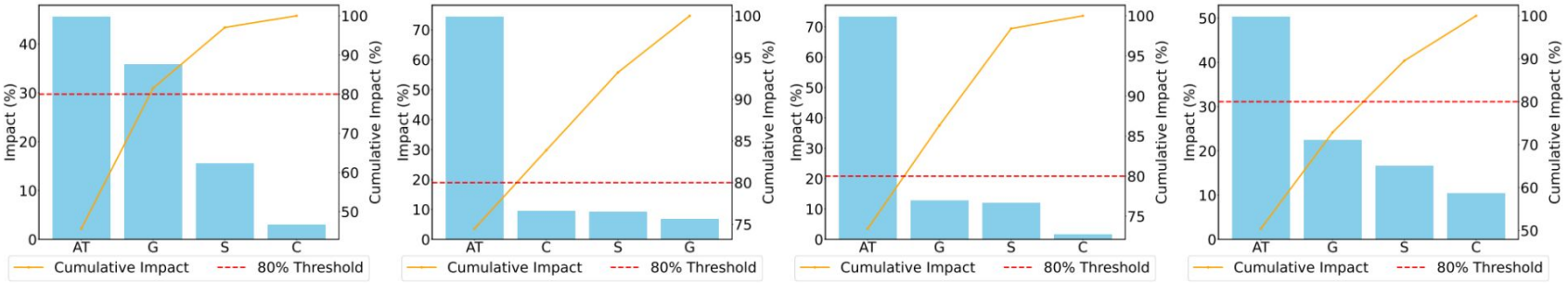
Figure 3: **Shrinking BF with output length over various tasks for Llama-3-70B and Llama-3-70B-Instruct.** For better visualization, we compute the exponential moving averaged values of BF with the smoothing factor set as 0.1.

CoT: Push decision to later stage, where BF is low. Complexity has different effect on BF => Cognitive Overload

Which Factor affect BF the most

Y-axis measures sensitivity of that factor on BF.

- alignment tuning is the most influential factor affecting BF
- For task with richer inputs, prompt complexity C and model size M also important.



(a) Cognac (b) MMLU (c) BBCNews (d) Creative StoryGen

Figure 4: **Pareto Analysis of BF across various IFs.** AT indicates whether the model is aligned. C denotes the prompt complexity. S refers to model size, and G refers to model generation (Llama-2 vs. Llama-3). Across all settings, alignment tuning has the most pronounced impact on BF.

Resampling

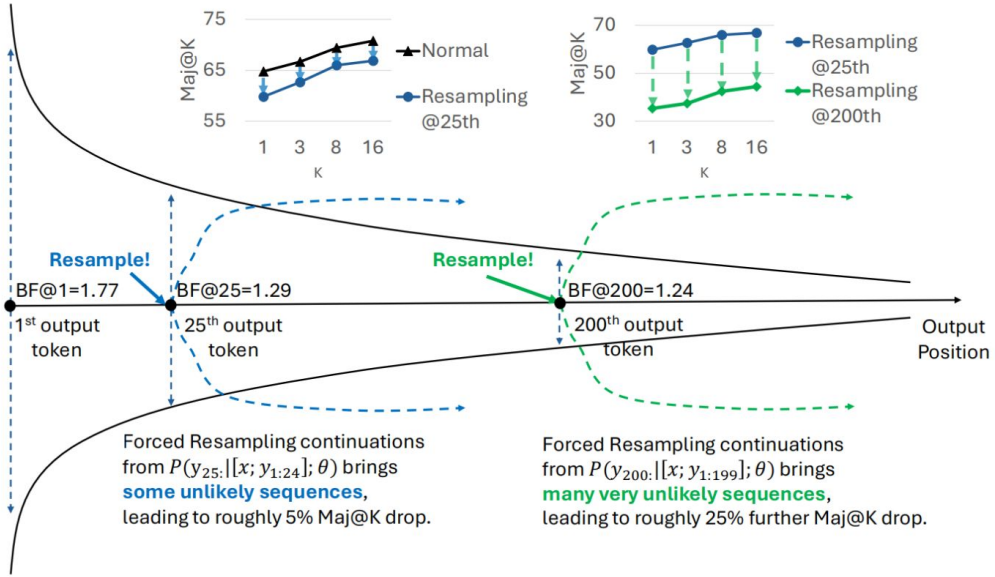


Figure 6: **Resampling from different output positions to assess the effect of interrupting BF reduction**. We resample new continuations at the 25th and 200th output token of DeepSeek-Distilled Llama-8B MMLU outputs. Results show substantial performance drops at both positions.

Insight: Parallel sampling should be applied early, while BF remains high, to ensure meaningful diversity and avoid quality degradation.

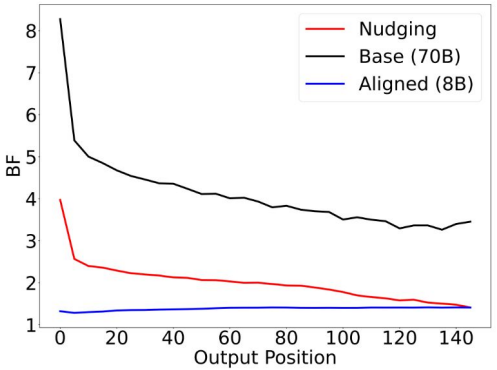
Nudging Experiment

Does base model already has the low BF, "aligned-like" paths?

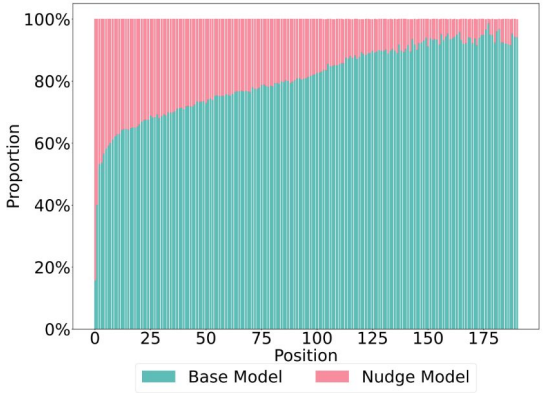
- Start with base model decoding, but whenever the base model's top-1 probability is "low," they temporarily switch to an aligned model (Llama-3-8B-Instruct) to emit a single token.

The left graph show nudging actually decrease BF => the prefix generated by the nudging model is of low probability.

Right graph show nudging happens more in earlier decoding. Base model locks in if the prefix is "nudged" (See how low BF is at position >100, while only 20% token is nudged)



(a) Output BF Dynamics



(b) Nudging Ratio Histogram

Figure 7: Nudging Experiments over Just-Eval-Instruct dataset.

Summary

Let's pop out and think about what we learned!

- Aligned models exhibit BF values nearly an order of magnitude lower than their base counterparts, with BF declining further during generation.
  - Alignment training as pruning, one major effect is stability (less creative - hold on to this…)
  - Helps explain Aligned model's reduced output diversity, low sampling variance, and insensitivity to decoding strategies
- Aligned CoT models due to their especially low BF, produce more stable outputs. Mid-generation resampling yields degraded performance by forcing unlikely continuations, especially for resampling later in the output.
- Nudging experiments further support that alignment narrows generation by steering it toward stylistic tokens that activate low-entropy subspaces already present in the base model.
- Questions: Which components of alignment tuning/ pretraining drive these effects?

Roll the dice & look before you leap: Going beyond the creative limits of next-token prediction

Authors: Vaishnavh Nagarajan, Chen Henry Wu, Charles Ding, Aditi Raghunathan

Problem Statement

Current LLM Limitations in Creative Tasks

- Large Language Models (LLMs) struggle with creative generation that requires **novelty, diversity, and originality** beyond mere correctness.
- Examples include generating challenging math problems, designing new proteins, or drawing surprising analogies.
- **Root cause:** Standard training misaligns with the global planning required for creative outputs.

Key Challenges

Why Creativity is Hard for Current LLMs

- Creativity is **difficult to quantify** objectively while maintaining coherence and avoiding training-data memorization.
- Next-Token Prediction (NTP) **optimizes for local likelihood but not global structure.**
- Output-level randomness (temperature) can hurt coherence and cause "**cognitive overload"**

Innovation

- Controlled Evaluation: Design minimal, algorithmic tasks to quantify creativity objectively
- Training Alternatives: **Teacherless Multi-token** and **diffusion** methods to capture global structure beyond NTP
- Input-level Randomness: Seed-conditioning with **random prefixes** to elicit diverse, coherent plans even with greedy decoding

Creativity Framework

- **Combinational Creativity:** Creating novel connections from existing stored knowledge
- Example: wordplay, analogies, and humor

- **Exploratory Creativity:** Constructing new structures within given constraints
- Example: design puzzle, word problems, and new proteins

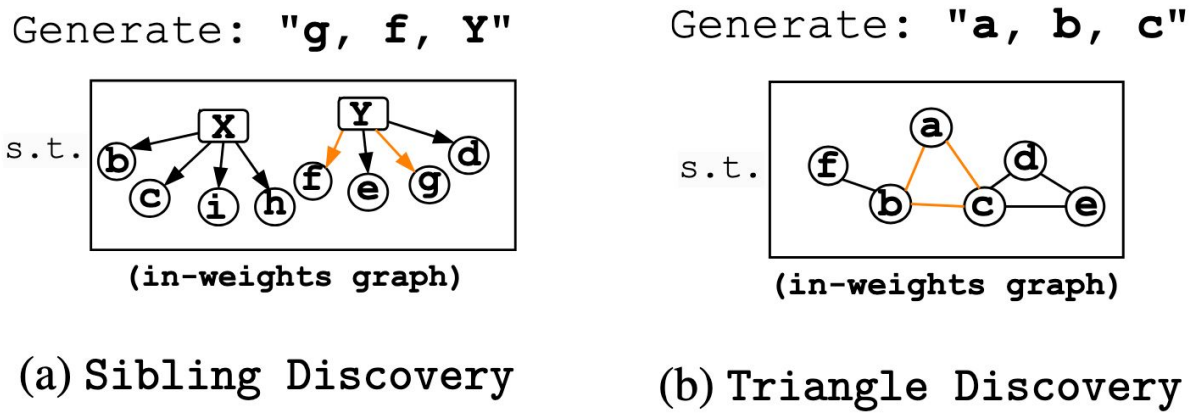
Combinational Creativity

Sibling Discovery:

- Goal: Find coherent triplet (child1, child2, parent) where both children connect to the same parent in a bipartite graph
- Challenge: Model must implicitly select parent first, then find siblings, but present parent last (adversarial to NTP)

Triangle Discovery:

- Goal: Generate 3 nodes forming a triangle in an undirected graph
- Challenge: Coordinate three edges simultaneously from memory



Exploratory Creativity

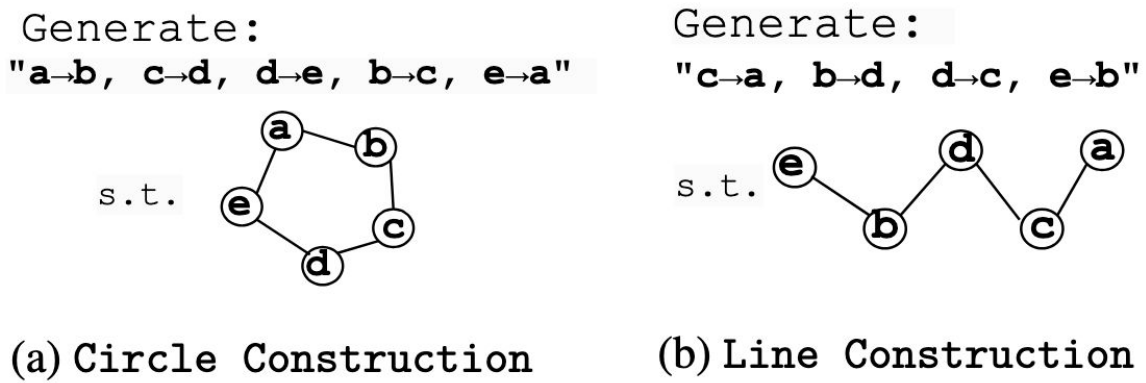
Circle Construction:

- Goal: Generate a list of connections (an "adjacency list") between distinct nodes that, when followed, form a complete, closed loop or cycle.

Line Construction:

- Goal: Similar to Circle Construction, but the generated connections must instead form a single, open path from a start node to an end node.

Creativity & Challenge: The model must produce novel and diverse arrangements of connections while ensuring global coherence (the entire structure is a valid circle or line).



(a) Circle Construction (b) Line Construction

Evaluation Metric

- **Creativity Score** = proportion of generated samples that are:
  - **Original**: not in training set
  - **Coherent**: valid under task rules
  - **Unique**: not duplicates in generation set

$$\hat{\mathbf{cr}}_N(T) = \frac{\mathbf{uniq}(\{\boldsymbol{s} \in T | \neg \mathbf{mem}_S(\boldsymbol{s}) \wedge \mathbf{coh}(\boldsymbol{s})\})}{|T|}$$

Next-Token Prediction Failures

- **Short-Sighted Focus:** NTP's sequential "next-token" approach is too local and struggles with the overall planning and "leap of thought" required for creative tasks.
- **Hidden Structure Problem:** NTP fails to infer deeper, non-obvious patterns where underlying thought process isn't immediately visible in the token sequence.
- **“Clever Hans” Cheat:** NTP can exploit local shortcuts (e.g., immediately inferring a parent from siblings) rather than learning a true, global latent plan.
- **Data-Hungry Learning:** Without a global plan, NTP resorts to learning complex, less efficient conditional distributions, requiring significantly more training data for creative tasks.

Alternative Training Paradigms

Teacherless Multi-token Training

- Method: Trains the model to predict **all output tokens simultaneously**, given only the initial prompt and dummy placeholders instead of sequential "teacher-forced" tokens. (Inference can still be autoregressive.)
- Advantage: Learns global to local mapping, closer to human planning process

Results (Gemma 2B):

- Leads to significantly higher algorithmic creativity and notably lower memorization of training data across diverse tasks on large transformer.

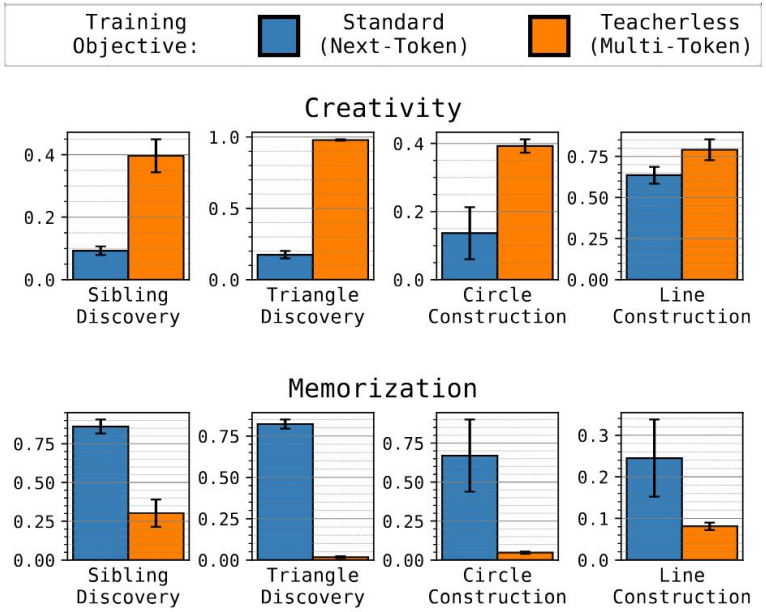


Figure 3. **Multi-token teacherless finetuning improves algorithmic creativity (top; Eq 1) and reduces memorization (bottom; fraction of generations seen during training) on our four open-ended algorithmic tasks for a `Gemma v1 (2B)` model.**

Alternative Training Paradigms

Diffusion Models

- Method: **Iteratively refines** a corrupted input (denoising from a random state) to generate a coherent output.
- Advantage: Naturally suitable for satisfying global constraints by refining the entire sequence, a good fit for creative tasks.

Results (GPT-2 v.s. SEDD):

- Diffusion models show better algorithmic creativity than (NTP) Transformers on most tasks, especially exploratory tasks.
- Diffusion provides the more reliable boost than teacherless training for smaller model.

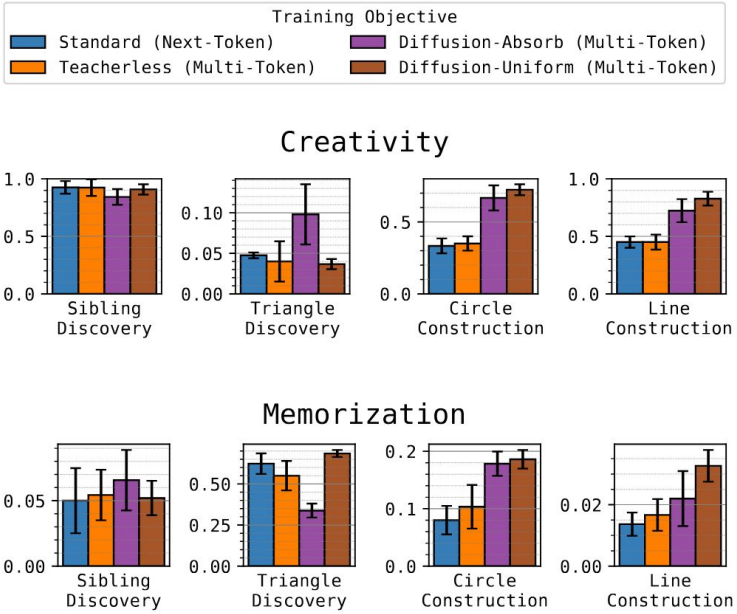


Figure 4. **Multi-token diffusion training improves algorithmic creativity (top; Eq 1) on three of our four open-ended tasks on `GPT-2 (86M)` and similarly-sized diffusion model, `SEDD (90M)`.** We report the best performance after tuning the sampling hyperparameters – temperature from $\{0, 0.5, 1, 2\}$, with or without top-K where $K = 50$.

Alternative Randomness

Seed-Conditioning

- Traditional temperature sampling (output randomness) might cause "cognitive overload" by forcing the model to process multiple "thoughts" to create a diverse output distribution.
- Introduce "seed-conditioning" by adding an arbitrary, random prefix to the input during both training and inference.
- This input-level noise helps the model focus on developing one particular thought per seed, resulting in diverse but coherent outputs. It also works as a structured way to introduce prompt variations.

Alternative Randomness

Seed-Conditioning

- Boosts algorithmic creativity in Transformers, even with greedy decoding.
- Often matches or surpasses the creativity achieved by conventional temperature sampling.
- Longer seed lengths generally lead to even higher algorithmic creativity.

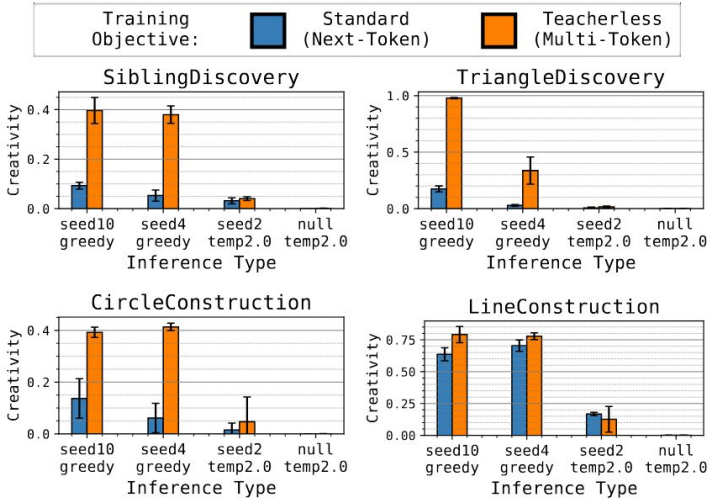


Figure 5. **Seed-conditioning significantly improves algorithmic creativity of both next- and multi-token prediction on `Gemma v1 (2B)` model.** The X-axis labels denote the prefix (at training and inference) and the temperature (at inference).

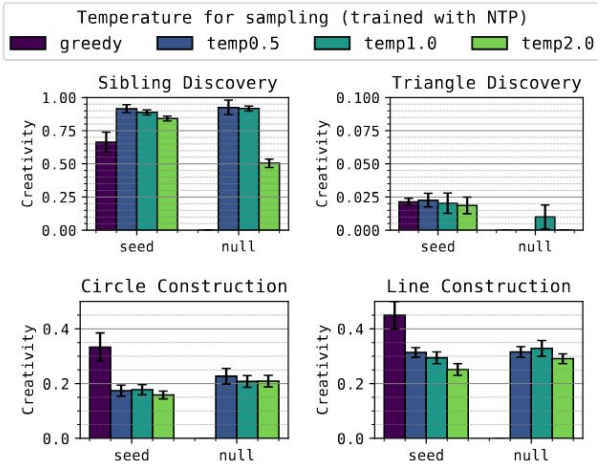


Figure 6. **Seed-conditioning achieves comparable algorithmic creativity in the `GPT-2 (86M)` model**: All models are trained with `NTP`. The color denotes temperature, while the X-axis, the prefix. Remarkably, seed-conditioning achieves comparable creativity even with greedy decoding, and improves creativity particularly in our exploratory creativity tasks (**bottom**).

Summary

- NTP's Creative Bottleneck: Standard next-token prediction limits creativity in open-ended tasks. It fails to capture global latent plans and leads to memorization.
- Multi-Token Paradigm Boosts Creativity: Teacherless multi-token objectives and diffusion models achieve higher algorithmic creativity, with substantially less memorization, especially in larger models.
- Seed-Conditioning Over Output Sampling: Injecting noisiness at the input via random prefixes is more effective for diversity. It competes with or outperforms output-temperature sampling, even with greedy decoding.
- **Limitations:** 1. Tasks are simplified abstractions 2. Multi-token training is harder to optimize 3. Full understanding and generalization of seed-conditioning to real data need further research.

Questions

- Can CoT help global planning?
- Which components of alignment tuning/finetuning drive BF down? Why BF decrease over output generation?
- What is the role of temperature at inference?
- What is rationale behind the “random prefix”?

Thank You!