

MULTI-TOKEN PREDICTION BOOSTS CREATIVITY IN ALGORITHMIC TASKS

Anonymous authors

Paper under double-blind review

ABSTRACT

In *open-ended* tasks — such as designing word problems or discovering novel proofs — the goal is not only correctness but also diversity and originality. Often, this requires a far-sighted, creative leap of thought. We argue that this requirement is misaligned with the objective of next-token prediction (NTP). To formulate our intuition, we design a suite of minimal algorithmic tasks loosely based on real-world creative endeavors. Concretely, our tasks require an open-ended *stochastic* planning step that (a) discovers new connections in a knowledge graph (loosely inspired by word-play, humor or drawing analogies) or (b) constructs new patterns (loosely inspired by constructing word problems, puzzles or mysteries). We then conceptually and empirically argue how NTP leads to myopic shortcut-learning and excessive memorization, limiting its ability to generate novel solutions. In contrast, we find that multi-token approaches, namely teacherless training and diffusion models, can overcome these limitations and comparatively excel on our algorithmic test-bed. Orthogonally, we find that creativity in our tasks is greatly improved by training with a random hash prefix (which we dub as “*hash-conditioning*”). Thus our work offers a principled, minimal test-bed for studying open-ended forms of intelligence and also a new angle to take a more serious interest in the paradigm of multi-token prediction.

1 INTRODUCTION

Not all forms of intelligence are solely about being correct or wrong. In *open-ended* tasks, what also matters is the ability to find creative ways to satisfy a request, making surprising and fresh connections never seen before. For instance, consider highly under-specified prompts like: “*Generate a challenging high-school word problem involving the Pythagoras Theorem.*” or “*Provide a vivid, compelling analogy to explain the difference between quantum and classical mechanics.*” or “*What happens when a physicist and a computer scientist walk into a bar?*”. Even the very task of generating the above illustrative prompts is an open-ended one. Being able to generate diverse and original responses becomes crucial as we explore LLMs as tools for scientific discovery and idea generation (Si et al., 2024) and as we enter into an era of generating training data with LLMs (Yu et al., 2024; Yang et al., 2024c; Wang et al., 2023).

Some open-ended tasks are conceptually straightforward, like generating names (Zhang et al., 2024b) or simple subject-verb-object sentences (Hopkins et al., 2023). But many others — like the open-ended prompts above — require a more sophisticated form of ideation. This sophistication is said to come from a purely random flash of creative insight, dubbed variously in literature as a leap of thought (Wang et al., 2024a; Talmor et al., 2020; Zhong et al., 2024) or a “eureka” moment (Bubeck et al., 2023) or a mental leap (Holyoak & Thagard, 1995; Callaway, 2013; Hofstadter, 1995) or an incubation step (Varshney et al., 2019).

Such leaps, we argue, appear misaligned with the next-token objective. First, a leap must implicitly search, plan and orchestrate various choices — i.e., choices that are *random yet coherent* — to yield diverse yet interesting outputs. Next, even if the training data contained outputs of many human-generated leaps, the leaps themselves are *latent*. In fact, it seems impossible to spell out the computation within leaps as a short, natural chain of thought. (What is the full chain of thought that goes behind each possible completion of “*A horse walks into a bar...*” joke?) These realizations about the nature of a leap lead us to the core question of our paper: given the outputs of some leaps in an

open-ended task, can next-token learning infer the underlying leap-producing process and generate novel and diverse outputs?

To attempt to answer this question, we crystallize open-ended settings that on the one hand are minimal, easy-to-quantify and controllable and on the other capture the crucial computational aspects of real-world open-ended tasks. This would complement a growing line of work addressing the lofty goal of directly studying creative natural language tasks (Zhang et al., 2024a; Si et al., 2024; Franceschelli & Musolesi, 2023; Lu et al., 2024; Chakrabarty et al., 2024). Most related to our goal are works that have studied diversity of next-token models in tasks such as graph path-finding (Khona et al., 2024) and challenging CFGs (Allen-Zhu & Li, 2023b). Broadly, we term these tasks as open-ended algorithmic tasks. In this context, our aim can be viewed as a more principled investigation of minimal instances of such tasks. This allows us to pinpoint fundamental issues with next-token prediction and systematically propose alternative learning approaches.

As a first step, we use algorithmic tasks to isolate two fundamental types of leaps, loosely inspired by two modes of creativity in cognitive science literature (Boden, 2003) (also see Franceschelli & Musolesi (2023)) termed *combinational* and *exploratory* creativity. For the vastly simpler scope of this paper, this distinction corresponds to computational tasks that require discovering novel connections in knowledge (e.g., wordplay and analogies, see Fig 1a, 1b), and tasks that require constructing patterns that are resolvable in novel ways (e.g., stories and word problems, see Fig 1c, 1d).

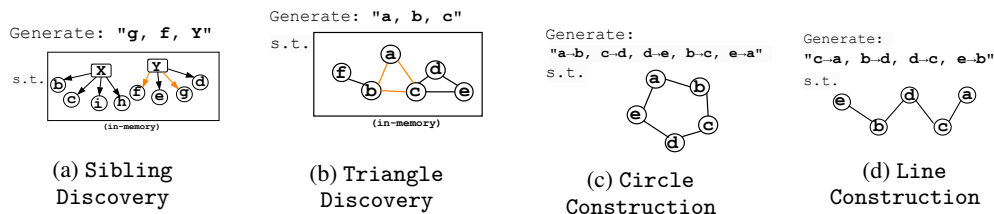


Figure 1: **Tasks inspired by combinational creativity (Fig 1a, 1b):** Skills like research, humor and analogies often require identifying novel multi-hop connections from known relationships: creating the wordplay “What kind of shoes do spies wear? Sneakers.” requires selecting a pair of words shoes and spies that lead to a pre-planned punchline, sneakers — which is a mutual semantic neighbor. Loosely inspired by this, we consider the minimal tasks of discovering siblings and triangles from an in-memory knowledge graph. **Tasks inspired by exploratory creativity (Fig 1c, 1d):** Skills like designing problem sets or writing plots, require devising patterns that can be *resolved* in novel ways under some logical constraints. Loosely inspired by this, we consider the minimal tasks of constructing randomized adjacency lists that resolve into a circle or a line graph.

In these tasks, we then show a separation in the creativity of next-token and multi-token approaches (namely, teacherless training (Bachmann & Nagarajan, 2024; Monea et al., 2023; Tschannen et al., 2023) and discrete diffusion models (Hoogetboom et al., 2021; Austin et al., 2021; Lou et al., 2023)). Intuitively, in all our tasks, optimally inferring the latent creative leap requires observing global patterns; but we argue how the next-token models learns local shortcut patterns (called *Clever Hans* cheats (B&N’24)). As a result, next-token learning under-utilizes the data, and results in significantly sub-optimal creativity and high memorization (See Fig 2); multi-token approaches show higher creativity and far lower memorization. As an orthogonal finding, we also discover that training either of these objectives via *hash-conditioning* — prefixing the input with random strings — significantly boosts to creativity in these open-ended tasks.

Overall, we hope our study advances the field in two directions. First, we provide a new angle to advocate for multi-token approaches, orthogonal to the “path-star” example in B&N’24. Whereas, the path-star example portrays a gap in *correctness* of reasoning, ours shows a gap in *diversity* of open-ended thinking. Next, the gap we show appears even in 2-token-lookahead tasks as against the multi-hop path-star task. Perhaps most conceptually important is the fact that, while the path-star task is amendable to next-token prediction upon reversing the tokens, we identify tasks where no re-ordering is friendly towards next-token prediction — *the optimal thing to do is to globally learn the whole string*. This presents a challenge to recent proposals that permute the next-token objective as a way of fixing it (Pannatier et al., 2024; Kitouni et al., 2024; Nolte et al., 2024).

108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

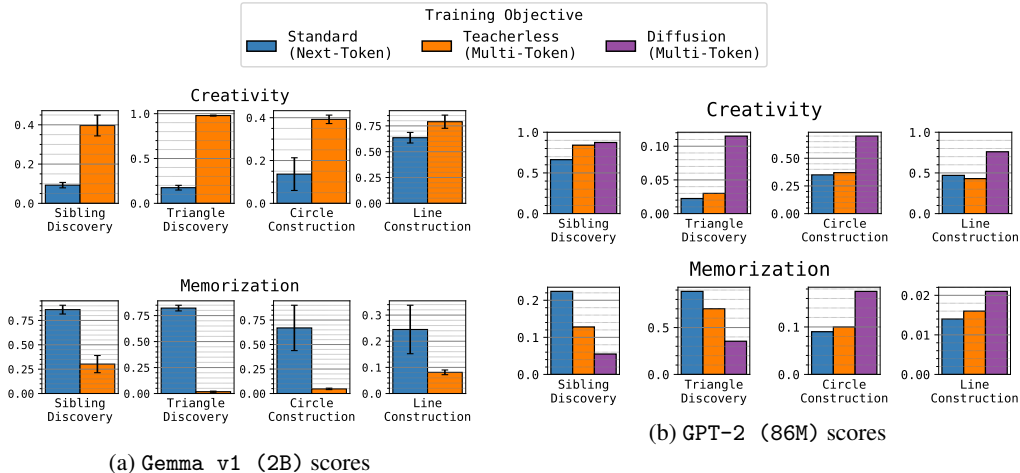


Figure 2: **Multi-token teacherless finetuning improves creativity (top) while reducing memorization (bottom) on our four open-ended algorithmic tasks.** Fig 2a is for finetuning a pretrained Gemma v1 (2B). Fig 2b is a diffusion model compared against a similarly-size GPT-2 (86M).

We also hope that our work provides a foundation to think about open-ended tasks which are extremely hard to quantify in the wild. This may spur algorithmic explorations on improving diversity (such as our approach of hash-conditioning) and on curbing verbatim memorization in language models.

Our contributions:

1. We create minimal, controlled and easy-to-quantify open-ended algorithmic tasks. These tasks isolate, and loosely capture two fundamental modes of creativity.
2. We find that multi-token prediction through teacherless training or diffusion results in significantly increased creativity and reduced memorization in all our tasks compared to next-token prediction.
3. We show a new gap between multi- and next-token prediction owing to shortcuts learned in NTP. The gap is in tasks that require creativity (not correctness), require much simpler 2-token lookahead, and are permutation-invariant.
4. We find that *hash-conditioning* i.e., training with random hash prefixes, greatly improves diversity of its outputs in our tasks, compared to temperature scaling.

2 OPEN-ENDED ALGORITHMIC TASKS & TWO TYPES OF CREATIVITY

We are interested in designing simple algorithmic tasks that are loosely inspired by real-world endeavors such as generating scientific ideas, humor, narration, or problem-set design, where one needs to generate strings that are both “interesting” and never seen before. A key characteristic of such tasks is that they require ideation or a flash of insight before beginning to generate the output. This ideation step is a leap of thought that (a) is implicit (is not spelled out in token space), (b) involves discrete *random* choices (c) and together, those choices must be *coherent* in that they are carefully planned to satisfy various non-trivial, discrete constraints. These constraints fundamentally define the task and make it interesting e.g., a word problem should be solvable by arithmetic rules, or a twist in a story must lead to a resolution by logical rules. The goal in such open-ended tasks is not just coherence though, but also diversity and novelty — generations must be as varied as possible and must not be regurgitated training data.

Open-ended tasks that do not require planning. To design tasks that capture the aforementioned “leap” or a creative planning step, we first clarify what tasks do not require such a step. One simple open-ended task that may come to mind is something akin to generating uniformly-random celebrity names (Zhang et al., 2024b). However, there is no opportunity to create a novel string here. A more interesting example may be generating grammatically coherent PCFG strings following a subject verb object format (e.g., the cat chased a rat) like in Hopkins et al. (2023). While novel strings become possible here, no sophisticated mental leaps are involved; each token can be generated

on the fly, satisfying a local next-token constraint to be coherent. Thus, we aim to create tasks that involve a more global constraint — we design these tasks inspired by two fundamentally different types of creative endeavors discussed in cognitive science literature (Boden, 2003).

Combinational creativity. Consider rudimentary word-play of the form “What genre do balloons enjoy? Pop music.” or “What kind of shoes do spies wear? Sneakers.” The novelty here lies in planning a sentence that begins with two unrelated entities (genre & balloons) but eventually reveals a punchline (pop) that is a mutual neighbor on a semantic graph. More broadly, Boden (2003) argues that many tasks, like the above, involve “making unfamiliar combinations of familiar ideas” or the “unexpected juxtaposition of [known] ideas”. Other tasks include drawing analogies, or finding connections between disparate ideas in science.

Exploratory creativity. Consider on the other hand, the act of developing a mystery or designing logical puzzles. These endeavors require constructing patterns that are altogether new but satisfy a non-trivial global constraint: they should be *resolvable* as per some rules (e.g., logic). Such endeavors appear to fall into a second class of *exploratory* creativity in Boden (2003). This includes much grander forms of exploration e.g., exploring various forms of artistic output within a stylistic constraint, or exploring various corollaries within a theoretical paradigm in physics or chemistry.

In the upcoming sections, we will attempt to capture some computational aspects of basic instances within the two classes of creative skills above. We emphasize that by no means does our minimal algorithmic setup intend to capture the human values or context that go into these endeavors; nor do they capture the rich array of creative acts that Boden (2003) discusses within these categories.

The basic setting and notations. In all our tasks, we assume the standard generative model setting where the model must learn an underlying distribution \mathcal{D} through a training set S of m independent samples $s_i \sim \mathcal{D}$. In our case, the distribution is over a space \mathbb{V}^L of L -length strings. Our tasks are open-ended in that there is no one correct answer to produce at test-time: the goal *is* to produce a random string from \mathcal{D} , much like responding to the query Design a high-school word problem. Each task is defined by a boolean coherence function $\text{coh} : \mathbb{V}^L \mapsto \{\text{true}, \text{false}\}$ which corresponds to the support i.e., $\text{supp}(\mathcal{D}) = \{s \in \mathbb{V}^L \mid \text{coh}(s)\}$. Upon witnessing a finite set of coherent examples S , the model must learn to generate only strings that are (a) coherent, (b) original (not memorized) and (c) diverse (covers the whole support). An exact quantification of this is computationally expensive. Instead, we approximate it by sampling a set T of many independent generations from the model and computing the fraction of T that is original, coherent and unique. Let the boolean $\text{mem}_S(s)$ denote whether an example s is from the training set S and let the integer function $\text{uniq}(X)$ denote the number of unique examples in a set X . Then, we define our (empirical, computational) creativity metric:

$$\hat{\text{cr}}_N(T) = \frac{\text{uniq}(\{s \in T \mid \neg \text{mem}_S(s) \wedge \text{coh}(s)\})}{|T|}. \quad (1)$$

2.1 ALGORITHMIC TASKS INSPIRED BY COMBINATIONAL CREATIVITY

Sibling discovery. We simplify the word-play example through a task with an implicit “knowledge graph” \mathcal{G} made of parent vertices $\mathcal{V} = \{A, B, C, \dots\}$ each neighboring a corresponding set of children $\text{nbr}(A) = \{a_1, a_2, \dots\}$, $\text{nbr}(B) = \{b_1, b_2, \dots\}$ and so on. We then define $\text{coh}(s)$ to hold true on “sibling-parent” triplets of the form $s = (\gamma, \gamma', \Gamma)$ such that $\gamma, \gamma' \in \text{nbr}(\Gamma)$. Here, one can think of the parent Γ as the “punchline” that delivers a connection to the first two vertices, in the same way *sneaker* connects *spies* and *shoes* in the wordplay example. In the learning task, the model witnesses a training set from a uniform distribution \mathcal{D} over all coherent strings for a fixed graph \mathcal{G} . The hope is that the model (a) stores the pairwise adjacencies of \mathcal{G} in its weights and (b) learns to generate novel sibling-parent triplets based on its in-weights knowledge of \mathcal{G} .

Reversed Sibling Discovery. Observe that the most natural order of generation is to plan the parent vertex (i.e., punchline) first, and pick the siblings after. Thus, the above task construction is adversarial towards NTP (more on this in §2.3). In contrast, if one were to reverse this task (i.e., generate parents followed by the siblings as $s = (\Gamma, \gamma, \gamma')$), we hypothesize that the task becomes NTP-friendly. We test the reversed dataset in our experiments too.

Triangle discovery. Next, we design a minimal dataset which we hope is not only adversarial towards left-to-right NTP, but should also be resistant to applying NTP on any permutation. In other words,

no refactorizing of the string must reveal a natural order of the string; one *must* learn to generate all tokens simultaneously, to infer the underlying process. Our idea is to make a simple change to the previous task: instead of demanding that the model discover siblings, we demand that it discover *triangles* from an appropriately-constructed knowledge graph $\mathcal{G} = (V, E)$ (which contains many triangles; see §D). Thus, in this task $\text{coh}((v_1, v_2, v_3)) = \text{true}$ iff all three edges between $\{v_1, v_2, v_3\}$ belong in \mathcal{G} . Furthermore, we define $\text{uniq}(\cdot)$ and $\text{mem}(\cdot)$ such that various permutations of the same triangle are counted as one.

2.2 ALGORITHMIC TASKS INSPIRED BY EXPLORATORY CREATIVITY

Circle construction. In this task, the generated strings must be randomized adjacency lists that can be rearranged to recover circle graphs of N vertices — *no knowledge graph is involved*. Specifically, let the generated list be $\mathbf{s} = (v_{i_1}, v_{i_2}), (v_{i_3}, v_{i_4}), \dots$. We define $\text{coh}(\mathbf{s}) = \text{true}$ iff there exists a *resolving* permutation π such that $\pi(\mathbf{s}) = (v_{j_1}, v_{j_2}), (v_{j_2}, v_{j_3}), \dots, (v_{j_n}, v_{j_1})$ for distinct j_1, j_2, \dots, j_n . i.e., each edge leads to the next, and eventually circles back to the first vertex. Loosely, we can think of the resolving permutation π as *how* a conflict in a story or a word problem or a puzzle is solved; the vertices as characters or mathematical objects; the rules of rearranging an adjacency list as rules of logic, math or story-building. The goal of creativity in this task is to create novel dynamics in the conflict, or equivalently, how it is resolved i.e., create strings with novel resolving permutations π . Thus we define uniq and mem such that different examples with the same resolving π are counted as the same, even if they have differing vertices (i.e., if only the entities differ, but the plot dynamics remain unaltered, we count them as duplicates).

Line construction. A simple variant of the above task is one where the edge set corresponds to a line graph. The resolving permutation π is such that $\pi(\mathbf{s}) = (v_{j_1}, v_{j_2}), (v_{j_2}, v_{j_3}), \dots, (v_{j_{n-1}}, v_{j_n})$ for distinct j_1, j_2, \dots, j_n . i.e., each edge leads to the next until a dead-end.

2.3 HOW NEXT-TOKEN PREDICTION MAY SUFFER FROM SHORT-CUT LEARNING IN OUR TASKS

Consider learning the Sibling Discovery dataset where we must generate sibling-parent triplets. Even if the parent must be emitted last, the most natural generative rule is to first learn to plan the parent $p(\text{parent})$, and then figure out $p(\text{siblings}|\text{parent})$ next, and then emit them in the reverse order. This is optimal as it requires learning only as many edges as there are in the graph i.e., $O(m \cdot n)$ many points, if there are m parents with n children.

With NTP however, we argue the model never plans the parent ahead of time. Observe that an NTP model learns the parent using a next-token conditional of the form $p(\text{parent}|\text{siblings})$. The parent here can be simply fit as the mutual neighbor of the two siblings revealed in the prefix. This is a shortcut which B&N’24 term such as Clever Hans *cheats*: the model witnesses and exploits part of the ground-truth it must generate (the siblings). Such cheats are simpler than even the true generative rule (where the parent has to be planned ahead of time) and are thus quickly picked up during learning.

Once the Clever Hans cheat is picked up, the model loses any supervision from the parent. In this backdrop, the model must learn the sibling only through the next-token-conditional, $p(\text{sibling}_2|\text{sibling}_1)$, without any supervision from the parent. This however would require witnessing every sibling-sibling pair totalling $O(m \cdot n^2)$ many training data — larger by a factor of n than the data requirements of the more natural rule.

More abstractly, much like in sophisticated creative tasks, in our tasks, the most natural way to generate the string is by planning various random latent choices (say z , here $z := \text{parent}$) and then learning a distribution $p(\mathbf{s}|z)$ over coherent strings \mathbf{s} . However, NTP myopically factorizes this into pieces of the form $p(s_i|s_{<i}, z)$. Consequently, the model learns uninformative latents from the later tokens (as it is lured by shortcuts called Clever Hans cheats). Conversely, the model is forced to learn complex data-hungry rules for the earlier tokens (as it lacks a natural plan).

3 EXPERIMENTS

Training objectives. For our next-token-trained (NTP) Transformers, we use the standard teacher-forcing objective used in supervised finetuning. Given prompt p and ground truth sequence s , the

model is trained to predict the i 'th token s_i , given as input the prompt and all ground truth tokens up to that point, $(p, s_{<i})$. We write the objective more explicitly in §B Eq 2. For the multi-token Transformer models, we use teacherless training (Monea et al., 2023; Bachmann & Nagarajan, 2024; Tschannen et al., 2023), where the model is trained to predict s_i simultaneously for all i , only given the prompt p (and some dummy tokens in place of the s that was once given as input). Since the exact details of this is irrelevant to our discussion, we describe this in Eq 2. To train our teacherless models, we use a hybrid of this objective and the next-token objective. We also use score entropy discrete diffusion model (SEDD, Lou et al., 2023) as a second multi-token model.

Inference. In all the above techniques, we extract each sample *independently* from the model (as against say, extracting them in continuous succession in the same context). For Transformers, during inference, we perform standard autoregression in both the next- and multi-token trained settings.

Hash-conditioning for Transformers Since our tasks are prompt-free, to produce diverse outputs from a Transformer, we *must* use temperature sampling (not greedy decoding). As an alternative to this we also consider prepending either a prompt of pause tokens (Goyal et al., 2024) or a unique hash string to each datapoint — both during training and during inference — in order to allow extra computation to the model before it emits its outputs. Perhaps, one could view these hash strings as a simpler alternative to varying the wordings of a prompt Li et al. (2023); Lau et al. (2024); Naik et al. (2024) or tuning a soft-prompt Wang et al. (2024b), both of which are known to induce diversity.

Please see §E for more experimental details, and §D for precise dataset details, and §F for ablations.

3.1 OBSERVATIONS

Multi-token prediction improves creativity score while reducing memorization significantly. In all our datasets, we observe that creativity score increases significantly — as much as a 5x factor — under the multi-token teacherless training for the Gemma v1 (2B) model, and under diffusion for a 90M model (as against next-token prediction on a similar-sized GPT-2 (86M) Transformer model). The gains are much smaller or absent with teacherless training of the smaller GPT-2 (86M) Transformer which echoes prior findings that multi-token objectives are hard to optimize (B&N'24) or even hurt smaller models (Gloeckle et al., 2024). Finally, in nearly all the above settings we find a strong reduction in memorization.

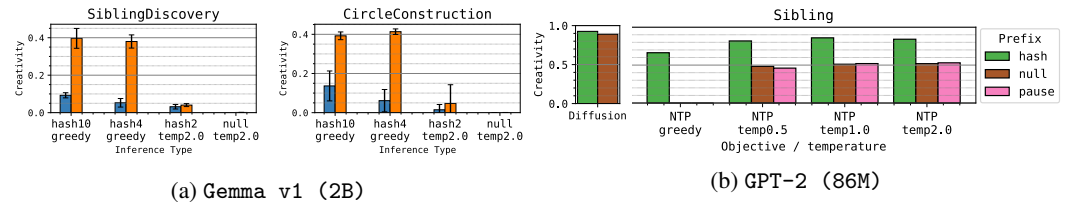


Figure 3: **Hash-conditioning significantly improves creativity for both next- and multi-token prediction** Fig 3a is for Gemma v1 (2B); the labels in the X-axis denote the prefix (used during training and inference) and the temperature (used during inference). Fig 3b is for the GPT-2 (86M) model; X-axis denotes the training and decoding procedure, while the legend indicates the prefix.

Hash-conditioning boosts Transformer creativity. For both the multi-token vs. next-token objectives, hash-conditioning crucially provides the best creativity in both our Transformer models (Fig 3a, 3b). It has no effect for diffusion models however. In fact, surprisingly, with hash-conditioning, there is no need for temperature; greedy coding suffices. Finally, increasing hash length correlates with increased creativity. Thus, for Transformers, we propose viewing hash-conditioning as a distinct knob for diversity more powerful than temperature-scaling.

4 CONCLUSIONS

This work provides a new argument in favor of multi-token learning, challenging the predominant next-token paradigm. To frame the argument, we design a suite of algorithmic tasks that are loosely inspired by two modes of creativity. Overall, we hope our work inspires discussion in the various directions of multi-token prediction, creativity and planning.

REFERENCES

- Alabdulmohsin, I., Tran, V. Q., and Dehghani, M. Fractal patterns may unravel the intelligence in next-token prediction, 2024.
- Allen-Zhu, Z. and Li, Y. Physics of language models: Part 3.2, knowledge manipulation. *CoRR*, abs/2309.14402, 2023a. doi: 10.48550/ARXIV.2309.14402. URL <https://doi.org/10.48550/arXiv.2309.14402>.
- Allen-Zhu, Z. and Li, Y. Physics of language models: Part 1, context-free grammar. *CoRR*, abs/2305.13673, 2023b. doi: 10.48550/ARXIV.2305.13673. URL <https://doi.org/10.48550/arXiv.2305.13673>.
- Anderson, B. R., Shah, J. H., and Kreminski, M. Homogenization effects of large language models on human creative ideation. In *Proceedings of the 16th Conference on Creativity & Cognition, Chicago, IL, USA, June 23-26, 2024*, pp. 413–425. ACM, 2024. URL <https://doi.org/10.1145/3635636.3656204>.
- Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces. *NeurIPS*, 2021.
- Bachmann, G. and Nagarajan, V. The pitfalls of next-token prediction. In *Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research*, pp. 2296–2318, 2024.
- Berglund, L., Tong, M., Kaufmann, M., Balesni, M., Stickland, A. C., Korbak, T., and Evans, O. The reversal curse: Llms trained on "a is b" fail to learn "b is a". In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL <https://openreview.net/forum?id=GPkTIktA0k>.
- Boden, M. A. *The Creative Mind - Myths and Mechanisms (2. ed.)*. Routledge, 2003.
- Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Józefowicz, R., and Bengio, S. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016*, pp. 10–21. ACL, 2016.
- Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.
- Callaway, E. Cognitive science: Leap of thought. *Nature*, 502, 2013.
- Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T. B., Song, D. X., Erlingsson, Ú., Oprea, A., and Raffel, C. Extracting training data from large language models. In *USENIX Security Symposium*, 2020.
- Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramèr, F., and Zhang, C. Quantifying memorization across neural language models. *ICLR*, 2023.
- Chakrabarty, T., Laban, P., Agarwal, D., Muresan, S., and Wu, C. Art or artifice? large language models and the false promise of creativity. In *Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI 2024, Honolulu, HI, USA, May 11-16, 2024*, pp. 30:1–30:34. ACM, 2024.
- Che, T., Li, Y., Jacob, A. P., Bengio, Y., and Li, W. Mode regularized generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.
- Chen, H. and Ding, N. Probing the "creativity" of large language models: Can models produce divergent semantic association? In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pp. 12881–12888. Association for Computational Linguistics, 2023.

- 378 Csikszentmihalyi, M. *Creativity: Flow and the Psychology of Discovery and Invention*. HarperCollins
379 Publishers, New York, NY, first edition, 1996.
380
- 381 Dawid, A. and LeCun, Y. Introduction to latent variable energy-based models: A path towards
382 autonomous machine intelligence. *arXiv preprint arXiv:2306.02572*, 2023.
- 383 DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B.-L., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang,
384 C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D.-L., Li, E., Lin, F., Dai, F., Luo, F., Hao,
385 G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Zhang, H., Ding, H., Xin, H., Gao, H.,
386 Li, H., Qu, H., Cai, J. L., Liang, J., Guo, J., Ni, J., Li, J., Wang, J., Chen, J., Chen, J., Yuan, J., Qiu,
387 J., Li, J., Song, J.-M., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang,
388 L., Xu, L., Xia, L., Zhao, L., Wang, L., Zhang, L., Li, M., Wang, M., Zhang, M., Zhang, M., Tang,
389 M., Li, M., Tian, N., Huang, P., Wang, P., Zhang, P., Wang, Q., Zhu, Q., Chen, Q., Du, Q., Chen,
390 R. J., Jin, R. L., Ge, R., Zhang, R., Pan, R., Wang, R., Xu, R., Zhang, R., Chen, R., Li, S. S., Lu, S.,
391 Zhou, S., Chen, S., Wu, S.-P., Ye, S., Ma, S., Wang, S., Zhou, S., Yu, S., Zhou, S., Pan, S., Wang,
392 T., Yun, T., Pei, T., Sun, T., Xiao, W. L., Zeng, W., Zhao, W., An, W., Liu, W., Liang, W., Gao, W.,
393 Yu, W.-X., Zhang, W., Li, X. Q., Jin, X., Wang, X., Bi, X., Liu, X., Wang, X., Shen, X.-C., Chen,
394 X., Zhang, X., Chen, X., Nie, X., Sun, X., Wang, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yu, X.,
395 Song, X., Shan, X., Zhou, X., Yang, X., Li, X., Su, X., Lin, X., Li, Y. K., Wang, Y. Q., Wei, Y. X.,
396 Zhu, Y. X., Zhang, Y., Xu, Y., Huang, Y., Li, Y., Zhao, Y., Sun, Y., Li, Y., Wang, Y., Yu, Y., Zheng,
397 Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Tang, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y.-B., Liu, Y.,
398 Guo, Y., Wu, Y., Ou, Y., Zhu, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Zha, Y., Xiong, Y., Ma, Y.,
399 Yan, Y., Luo, Y.-W., mei You, Y., Liu, Y., Zhou, Y., Wu, Z. F., Ren, Z., Ren, Z., Sha, Z., Fu, Z., Xu,
400 Z., Huang, Z., Zhang, Z., Xie, Z., guo Zhang, Z., Hao, Z., Gou, Z., Ma, Z., Yan, Z., Shao, Z., Xu,
401 Z., Wu, Z., Zhang, Z., Li, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z.-A., Xie, Z., Song, Z., Gao, Z., and
402 Pan, Z. Deepseek-v3 technical report. In *ArXiv*, 2024.
- 403 DeSalvo, G., Kagy, J.-F., Karydas, L., Rostamizadeh, A., and Kumar, S. No more hard prompts:
404 Softsrvc prompting for synthetic data generation, 2024. URL [https://arxiv.org/abs/2410.
405 16534](https://arxiv.org/abs/2410.16534).
- 406 Du, L., Mei, H., and Eisner, J. Autoregressive modeling with lookahead attention. *arXiv preprint
407 arXiv:2305.12272*, 2023.
- 408 Dziri, N., Lu, X., Sclar, M., Li, X. L., Jiang, L., Lin, B. Y., Welleck, S., West, P., Bhagavatula, C.,
409 Le Bras, R., et al. Faith and fate: Limits of transformers on compositionality. *Advances in Neural
410 Information Processing Systems*, 36, 2024.
- 411
- 412 Feng, G., Zhang, B., Gu, Y., Ye, H., He, D., and Wang, L. Towards revealing the mystery behind
413 chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*,
414 36, 2023.
- 415 Franceschelli, G. and Musolesi, M. On the creativity of large language models. *CoRR*,
416 abs/2304.00008, 2023.
- 417
- 418 Gloeckle, F., Idrissi, B. Y., Rozière, B., Lopez-Paz, D., and Synnaeve, G. Better & faster large
419 language models via multi-token prediction. 2024.
- 420
- 421 Gong, S., Li, M., Feng, J., Wu, Z., and Kong, L. Diffuseq: Sequence to sequence text generation with
422 diffusion models. In *The Eleventh International Conference on Learning Representations, ICLR
423 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.
- 424
- 425 Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C.,
426 and Bengio, Y. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, 2020. doi:
10.1145/3422622. URL <https://doi.org/10.1145/3422622>.
- 427
- 428 Goyal, A., Sordoni, A., Côté, M.-A., Ke, N. R., and Bengio, Y. Z-forcing: Training stochastic
429 recurrent networks. *NeurIPS*, 2017.
- 430
- 431 Goyal, S., Ji, Z., Rawat, A. S., Menon, A. K., Kumar, S., and Nagarajan, V. Think before you speak:
Training language models with pause tokens. *The Twelfth International Conference on Learning
Representations, ICLR 2024*, 2024.

- 432 Gu, J., Bradbury, J., Xiong, C., Li, V. O. K., and Socher, R. Non-autoregressive neural machine
433 translation. In *6th International Conference on Learning Representations, ICLR 2018, Conference*
434 *Track Proceedings*. OpenReview.net, 2018.
- 435 Hofstadter, D. A review of mental leaps: Analogy in creative thought. *AI Mag.*, 16(3):75–80, 1995.
436 doi: 10.1609/AIMAG.V16I3.1154. URL <https://doi.org/10.1609/aimag.v16i3.1154>.
- 437
438 Holyoak, K. J. and Thagard, P. *Mental leaps: analogy in creative thought*. MIT Press, Cambridge,
439 MA, USA, 1995. ISBN 0262082330.
- 440 Hooeboom, E., Nielsen, D., Jaini, P., Forr’e, P., and Welling, M. Argmax flows and multinomial
441 diffusion: Learning categorical distributions. In *Neural Information Processing Systems*, 2021.
- 442
443 Hopkins, A. K., Renda, A., and Carbin, M. Can LLMs generate random numbers? evaluating LLM
444 sampling in controlled domains. In *ICML 2023 Workshop: Sampling and Optimization in Discrete*
445 *Space*, 2023. URL <https://openreview.net/forum?id=Vhh1K9LjVI>.
- 446
447 Hu, E. S., Ahn, K., Liu, Q., Xu, H., Tomar, M., Langford, A., Jayaraman, D., Lamb, A., and
448 Langford, J. Learning to achieve goals with belief state transformers. abs/2410.23506, 2024. doi:
449 10.48550/ARXIV.2410.23506. URL <https://doi.org/10.48550/arXiv.2410.23506>.
- 450
451 Hua, H., Li, X., Dou, D., Xu, C., and Luo, J. Fine-tuning pre-trained language models with noise
452 stability regularization. *CoRR*, 2022.
- 453
454 Jain, N., Chiang, P., Wen, Y., Kirchenbauer, J., Chu, H., Somepalli, G., Bartoldson, B. R., Kailkhura,
455 B., Schwarzschild, A., Saha, A., Goldblum, M., Geiping, J., and Goldstein, T. Neptune: Noisy
456 embeddings improve instruction finetuning. In *The Twelfth International Conference on Learning*
457 *Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.
- 458
459 Kääriäinen, M. Lower bounds for reductions. In *Atomic Learning Workshop*, 2006.
- 460
461 Kalai, A. T. and Vempala, S. S. Calibrated language models must hallucinate. In *Proceedings of the*
462 *56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June*
463 *24-28, 2024*, pp. 160–171. ACM, 2024.
- 464
465 Kalavasis, A., Mehrotra, A., and Velegkas, G. On the limits of language generation: Trade-offs
466 between hallucination and mode collapse. abs/2411.09642, 2024.
- 467
468 Kamb, M. and Ganguli, S. An analytic theory of creativity in convolutional diffusion models, 2024.
469 URL <https://arxiv.org/abs/2412.20292>.
- 470
471 Khona, M., Okawa, M., Hula, J., Ramesh, R., Nishi, K., Dick, R. P., Lubana, E. S., and Tanaka, H.
472 Towards an understanding of stepwise inference in transformers: A synthetic graph navigation
473 model. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria,*
474 *July 21-27, 2024*. OpenReview.net, 2024.
- 475
476 Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.),
477 *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April*
478 *14-16, 2014, Conference Track Proceedings*, 2014. URL <http://arxiv.org/abs/1312.6114>.
- 479
480 Kitouni, O., Nolte, N., Bouchacourt, D., Williams, A., Rabbat, M., and Ibrahim, M. The factorization
481 curse: Which tokens you predict underlie the reversal curse and more. *CoRR*, abs/2406.05183, 2024.
482 doi: 10.48550/ARXIV.2406.05183. URL <https://doi.org/10.48550/arXiv.2406.05183>.
- 483
484 Kleinberg, J. M. and Mullainathan, S. Language generation in the limit. *CoRR*, abs/2404.06757, 2024.
485 doi: 10.48550/ARXIV.2404.06757. URL <https://doi.org/10.48550/arXiv.2404.06757>.
- 486
487 Lai, Y., Zhang, C., Feng, Y., Huang, Q., and Zhao, D. Why machine reading comprehension models
488 learn shortcuts? In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021,*
489 *Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pp. 989–1002.
490 Association for Computational Linguistics, 2021.
- 491
492 Lau, G. K. R., Hu, W., Liu, D., Chen, J., Ng, S.-K., and Low, B. K. H. Dipper: Diversity in
493 prompts for producing large language model ensembles in reasoning tasks, 2024. URL <https://arxiv.org/abs/2412.15238>.

- 486 LeCun, Y. Do large language models need sensory grounding for meaning and understanding?
487 University Lecture, 2024.
- 488
- 489 Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.-G., and Chen, W. Making language models better
490 reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association*
491 *for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada, July 2023. Association
492 for Computational Linguistics. URL <https://aclanthology.org/2023.acl-long.291/>.
- 493 Liu, B., Ash, J. T., Goel, S., Krishnamurthy, A., and Zhang, C. Transformers learn shortcuts to
494 automata. In *The Eleventh International Conference on Learning Representations, ICLR 2023*,
495 2023.
- 496 Lou, A., Meng, C., and Ermon, S. Discrete diffusion modeling by estimating the ratios of the data
497 distribution. In *International Conference on Machine Learning*, 2023.
- 498
- 499 Lu, X., Sclar, M., Hallinan, S., Mireshghallah, N., Liu, J., Han, S., Ettinger, A., Jiang, L., Chandu,
500 K. R., Dziri, N., and Choi, Y. AI as humanity’s salieri: Quantifying linguistic creativity of language
501 models via systematic attribution of machine text against web text. abs/2410.04265, 2024.
- 502 Malach, E. Auto-regressive next-token predictors are universal learners. *arXiv preprint*
503 *arXiv:2309.06979*, 2023.
- 504
- 505 McCoy, R. T., Yao, S., Friedman, D., Hardy, M., and Griffiths, T. L. Embers of autoregression:
506 Understanding large language models through the problem they are trained to solve. *arXiv preprint*
507 *arXiv:2309.13638*, 2023.
- 508 Merrill, W. and Sabharwal, A. The expressive power of transformers with chain of thought. In *The*
509 *Twelfth International Conference on Learning Representations*, 2024.
- 510
- 511 Mirowski, P. W., Love, J., Mathewson, K. W., and Mohamed, S. A robot walks into a bar: Can
512 language models serve as creativity support tools for comedy? an evaluation of llms’ humour
513 alignment with comedians. *CoRR*, abs/2405.20956, 2024.
- 514 Momennejad, I., Hasanbeig, H., Frujeri, F. V., Sharma, H., Ness, R. O., Jojic, N., Palangi, H.,
515 and Larson, J. Evaluating cognitive maps and planning in large language models with cogeval.
516 *Advances in Neural Information Processing Systems*, 36, 2023.
- 517
- 518 Monea, G., Joulin, A., and Grave, E. Pass: Parallel speculative sampling. *3rd Workshop on Efficient*
519 *Natural Language and Speech Processing (NeurIPS 2023)*, 2023.
- 520 Nagarajan, V., Raffel, C., and Goodfellow, I. J. Theoretical insights into memorization in gans. In
521 *Neural Information Processing Systems Workshop*, volume 1, pp. 3, 2018.
- 522
- 523 Naik, R., Chandrasekaran, V., Yuksekogonul, M., Palangi, H., and Nushi, B. Diversity of thought
524 improves reasoning abilities of llms, 2024. URL <https://arxiv.org/abs/2310.07088>.
- 525 Nakkiran, P., Bradley, A., Zhou, H., and Advani, M. Step-by-step diffusion: An elementary tutorial,
526 2024.
- 527
- 528 Nallapati, R., Zhou, B., dos santos, C. N., Gulcehre, C., and Xiang, B. Abstractive text summarization
529 using sequence-to-sequence rnns and beyond, 2016. URL <https://arxiv.org/abs/1602.06023>.
- 530
- 531 Narayan, S., Cohen, S. B., and Lapata, M. Don’t give me the details, just the summary! topic-aware
532 convolutional neural networks for extreme summarization, 2018. URL <https://arxiv.org/abs/1808.08745>.
- 533
- 534 Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A.,
535 Wallace, E., Tramèr, F., and Lee, K. Scalable extraction of training data from (production) language
536 models. *ArXiv*, 2023.
- 537
- 538 Nolte, N., Kitouni, O., Williams, A., Rabbat, M., and Ibrahim, M. Transformers can navigate mazes
539 with multi-step prediction. *CoRR*, abs/2412.05117, 2024. doi: 10.48550/ARXIV.2412.05117.
URL <https://doi.org/10.48550/arXiv.2412.05117>.

- 540 Padmakumar, V. and He, H. Does writing with language models reduce content diversity? In *The*
541 *Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May*
542 *7-11, 2024*. OpenReview.net, 2024. URL <https://openreview.net/forum?id=Feiz5HtCD0>.
543
- 544 Pannatier, A., Courdier, E., and Fleuret, F. σ -gpts: A new approach to autoregressive models. In
545 *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference,*
546 *ECML PKDD 2024, Vilnius, Lithuania, September 9-13, 2024, Proceedings, Part VII*, volume
547 14947 of *Lecture Notes in Computer Science*, pp. 143–159. Springer, 2024.
- 548 Peeperkorn, M., Kouwenhoven, T., Brown, D., and Jordanous, A. Is temperature the creativity
549 parameter of large language models? [abs/2405.00492](https://arxiv.org/abs/2405.00492), 2024.
550
- 551 Ranaldi, L. and Zanzotto, F. M. Hans, are you clever? clever hans effect analysis of neural systems,
552 2023.
553
- 554 Ross, S. and Bagnell, D. Efficient reductions for imitation learning. In *Proceedings of the Thirteenth*
555 *International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna*
556 *Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, 2010.
- 557 Sanford, C., Fatemi, B., Hall, E., Tsitsulin, A., Kazemi, S. M., Halcrow, J., Perozzi, B., and Mirrokni,
558 V. Understanding transformer reasoning capabilities via graph algorithms. [abs/2405.18512](https://arxiv.org/abs/2405.18512), 2024.
559
- 560 Saparov, A., Pawar, S., Pimpalgaonkar, S., Joshi, N., Pang, R. Y., Padmakumar, V., Kazemi, S. M.,
561 Kim, N., and He, H. Transformers struggle to learn to search, 2024. URL [https://arxiv.org/](https://arxiv.org/abs/2412.04703)
562 [abs/2412.04703](https://arxiv.org/abs/2412.04703).
- 563 Schnitzler, J., Ho, X., Huang, J., Boudin, F., Sugawara, S., and Aizawa, A. Morehopqa: More than
564 multi-hop reasoning. [abs/2406.13397](https://arxiv.org/abs/2406.13397). doi: 10.48550/ARXIV.2406.13397.
565
- 566 Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):
567 379–423, 1948.
- 568 Shannon, C. E. Prediction and entropy of printed english. *The Bell System Technical Journal*, 30(1):
569 50–64, 1951.
570
- 571 Shlegeris, B., Roger, F., Chan, L., and McLean, E. Language models are better than humans at
572 next-token prediction. *arXiv preprint arXiv:2212.11281*, 2022.
- 573 Si, C., Yang, D., and Hashimoto, T. Can llms generate novel research ideas? A large-scale human
574 study with 100+ NLP researchers. 2024.
575
- 576 Talmor, A., Tafjord, O., Clark, P., Goldberg, Y., and Berant, J. Leap-of-thought: Teaching pre-trained
577 models to systematically reason over implicit knowledge. In *Advances in Neural Information*
578 *Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020,*
579 *NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- 580 Tschannen, M., Kumar, M., Steiner, A., Zhai, X., Houlsby, N., and Beyer, L. Image captioners are
581 scalable vision learners too. In *Advances in Neural Information Processing Systems 36: Annual*
582 *Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA,*
583 *USA, December 10 - 16, 2023*, 2023.
584
- 585 Valmeekam, K., Marquez, M., and Kambhampati, S. Can large language models really improve by
586 self-critiquing their own plans? *arXiv preprint arXiv:2310.08118*, 2023a.
587
- 588 Valmeekam, K., Marquez, M., Olmo, A., Sreedharan, S., and Kambhampati, S. Planbench: An
589 extensible benchmark for evaluating large language models on planning and reasoning about
590 change, 2023b.
- 591 Valmeekam, K., Marquez, M., Sreedharan, S., and Kambhampati, S. On the planning abilities of
592 large language models - A critical investigation. In *Advances in Neural Information Processing*
593 *Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023,*
New Orleans, LA, USA, December 10 - 16, 2023, 2023c.

- 594 Varshney, L. R., Pinel, F., Varshney, K. R., Bhattacharjya, D., Schörgendorfer, A., and Chee, Y. A big
595 data approach to computational creativity: The curious case of chef watson. *IBM J. Res. Dev.*, 63
596 (1):7:1–7:18, 2019.
- 597
598 Walsh, M., Preus, A., and Gronski, E. Does chatgpt have a poetic style? In *Proceedings of the*
599 *Computational Humanities Research Conference 2024, Aarhus, Denmark, December 4-6, 2024*,
600 volume 3834 of *CEUR Workshop Proceedings*, pp. 1201–1219. CEUR-WS.org.
- 601 Wang, H., Zhao, Y., Li, D., Wang, X., Liu, G., Lan, X., and Wang, H. Innovative thinking,
602 infinite humor: Humor research of large language models through structured thought leaps.
603 abs/2410.10370, 2024a.
- 604 Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct:
605 Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual*
606 *Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023,*
607 *Toronto, Canada, July 9-14, 2023*, pp. 13484–13508. Association for Computational Linguistics,
608 2023.
- 609
610 Wang, Y., Luo, X., Wei, F., Liu, Y., Zhu, Q., Zhang, X., Yang, Q., Xu, D., and Che, W. Make
611 some noise: Unlocking language model parallel inference capability through noisy training. In
612 *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing,*
613 *EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 12914–12926. Association for
614 Computational Linguistics, 2024b.
- 615 Wies, N., Levine, Y., and Shashua, A. Sub-task decomposition enables learning in sequence to
616 sequence tasks. In *The Eleventh International Conference on Learning Representations, ICLR*
617 *2023, 2023*.
- 618
619 Yang, S., Gribovskaya, E., Kassner, N., Geva, M., and Riedel, S. Do large language models latently
620 perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for*
621 *Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16,*
622 *2024*, pp. 10210–10229. Association for Computational Linguistics, 2024a.
- 623 Yang, S., Kassner, N., Gribovskaya, E., Riedel, S., and Geva, M. Do large language models
624 perform latent multi-hop reasoning without exploiting shortcuts? abs/2411.16679, 2024b. doi:
625 10.48550/ARXIV.2411.16679. URL <https://doi.org/10.48550/arXiv.2411.16679>.
- 626 Yang, Z., Hu, Z., Salakhutdinov, R., and Berg-Kirkpatrick, T. Improved variational autoencoders for
627 text modeling using dilated convolutions. *ICML*, 2017.
- 628
629 Yang, Z., Band, N., Li, S., Candès, E. J., and Hashimoto, T. Synthetic continued pretraining.
630 *CoRR*, abs/2409.07431, 2024c. doi: 10.48550/ARXIV.2409.07431. URL [https://doi.org/10.](https://doi.org/10.48550/arXiv.2409.07431)
631 [48550/arXiv.2409.07431](https://doi.org/10.48550/arXiv.2409.07431).
- 632 Ye, J., Gao, J., Gong, S., Zheng, L., Jiang, X., Li, Z., and Kong, L. Beyond autoregression: Discrete
633 diffusion for complex reasoning and planning. 2024. doi: 10.48550/ARXIV.2410.14157.
- 634
635 Young, T. and You, Y. On the inconsistencies of conditionals learned by masked language models.
636 *arXiv preprint arXiv:2301.00068*, 2022.
- 637 Yu, L., Jiang, W., Shi, H., YU, J., Liu, Z., Zhang, Y., Kwok, J., Li, Z., Weller, A., and Liu, W.
638 Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth*
639 *International Conference on Learning Representations, 2024*. URL [https://openreview.net/](https://openreview.net/forum?id=N8N0hgNDRt)
640 [forum?id=N8N0hgNDRt](https://openreview.net/forum?id=N8N0hgNDRt).
- 641
642 Zhang, H., Li, L. H., Meng, T., Chang, K., and den Broeck, G. V. On the paradox of learning to
643 reason from data. In *Proceedings of the Thirty-Second International Joint Conference on Artificial*
644 *Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pp. 3365–3373. ijcai.org,
645 2023.
- 646 Zhang, J., Jain, L., Guo, Y., Chen, J., Zhou, K. L., Suresh, S., Wagenmaker, A., Sievert, S., Rogers,
647 T. T., Jamieson, K., Mankoff, R., and Nowak, R. Humor in AI: massive scale crowd-sourced
preferences and benchmarks for cartoon captioning. *CoRR*, abs/2406.10522, 2024a.

648 Zhang, Y., Schwarzschild, A., Carlini, N., Kolter, Z., and Ippolito, D. Forcing diffuse distributions
649 out of language models. [abs/2404.10859](https://arxiv.org/abs/2404.10859), 2024b.
650
651 Zhong, S., Huang, Z., Gao, S., Wen, W., Lin, L., Zitnik, M., and Zhou, P. Let’s think outside the box:
652 Exploring leap-of-thought in large language models with creative humor generation. In *IEEE/CVF*
653 *Conference on Computer Vision and Pattern Recognition, CVPR 2024*, 2024.
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701

702 A LIMITATIONS

703
704 First, we discuss limitations regarding our study of multi-token prediction

- 705
706 1. Our examples show that multi-token prediction outperforms next-token prediction in simple
707 tasks; but this does not preclude the existence of tasks where next-token prediction will
708 supercede in performance (i.e., the no free lunch theorem). Multi-token prediction is simply
709 a more general-purpose objective suitable to lookahead tasks.
- 710
711 2. In our proposed algorithmic tasks, there may be many ways to easily improve upon next-
712 token prediction — success here does not guarantee success on more complex tasks. (These
713 simple benchmarks are more interesting as a *failure* case of next-token prediction — failure
714 here guarantees failure in more complex tasks)
- 715
716 3. The teacherless multi-token prediction technique we advocate as an alternative is generally
717 harder to optimize than next-token prediction, especially for smaller models.
- 718
719 4. Even if teacherless training outperforms next-token prediction relatively, it is also far from
720 being a sufficiently diverse model even in some of our simple tasks.

721 Some limitations on the discussion on creativity:

- 722
723 1. The type of algorithmic tasks we study capture only a specific computational component of
724 a tiny subset of creative tasks that fall under the taxonomy in [Boden \(2003\)](#). There is yet
725 another class called *transformative* creativity that we do not look at, and also other important
726 taxonomies such as the Big-C/little-c creativity [Csikszentmihalyi \(1996\)](#). Big-C Creativity
727 corresponds breakthroughs and world-changing ideas; what we focus on is adjacent to
728 a class of little-c creativity tasks. Furthermore, there are many social and human values
729 encoded in creative human endeavors that we do not capture in our discussion.
- 730
731 2. Many real-world creative tasks are “out-of-distribution” in nature. Although our in-
732 distribution formulation provides a first step to thinking about the challenges of next-token
733 prediction and creative planning, we do not formulate what it means to “think out of the
734 box”.
- 735
736 3. Real-world creative tasks also apply over much larger context length and require drawing
737 connections from a significantly larger memory (literally, the set of all things a human may
738 know about). Our algorithmic tasks are tiny in comparison (although deliberately so).
- 739
740 4. Our measure of empirical creativity for algorithmic tasks is only a computationally-efficient
741 proxy. Achieving an absolute high creativity score does not imply a perfect coverage of the
742 space.

743 B TRANSFORMER TRAINING OBJECTIVES

744 Let LM_θ be our language model, parameterized by θ , for which $\text{LM}_\theta(\hat{s}_i = s_i; \mathbf{s}_{<i})$ is the probability it
745 assigns to the i th output \hat{s}_i being s_i , given as input a sequence $\mathbf{s}_{<i}$. Let (\mathbf{p}, \mathbf{r}) be a prefix-response
746 pair. In standard next-token finetuning, we maximize the objective:

$$747 \mathcal{J}_{\text{next-token}}(\theta) = \mathbb{E}_{\mathcal{D}} \left[\sum_{i=1}^{L_{\text{resp}}} \log \text{LM}_\theta(\hat{r}_i = r_i; \mathbf{p}, \mathbf{r}_{<i}) \right] \quad (2)$$

748 In teacherless (multi-token) training ([Monea et al., 2023](#); [Bachmann & Nagarajan, 2024](#); [Tschannen et al., 2023](#)), we make use of an uninformative input string \mathbb{S} that simply corresponds to a series of
749 dummy tokens $\$$.

$$750 \mathcal{J}_{\text{multi-token}}(\theta) = \mathbb{E}_{\mathcal{D}} \left[\sum_{i=1}^{L_{\text{resp}}} \log \text{LM}_\theta(\hat{r}_i = r_i; \mathbf{p}, \mathbb{S}_{<i}) \right] \quad (3)$$

C REMARKS AND DISCUSSION

Remark 1. We note that our approach of injecting noise into the model is somewhat different from how noise is processed in traditional VAEs (Kingma & Welling, 2014) or GANs (Goodfellow et al., 2020). In traditional approaches, although the model learns a noise-output mapping, this mapping is enforced only at a distribution level i.e., the distribution of noise vectors must map to a distribution of real vectors. However, in our approach we arbitrarily enforce what noise vector goes to what real datapoint, at a pointwise level. This raises the open questions of why hash-conditioning works in the first place — surprisingly, without breaking optimization or generalization — and whether there is a way to enforce it at distribution-level, and whether that can provide even greater improvements.

C.1 FURTHER EVIDENCE OF OUR ARGUMENT IN §2.3

Below we provide two more pieces of evidence affirming the failure mechanism of next-token prediction outlined in §2.3.

Improved creativity is not due to some form of capacity control. While §2.3 argues that multi-token prediction should help creativity by providing critical lookahead capabilities, it is also possible that it simply acts as a form of capacity control that prevents memorization. We rule this out in Fig 4: even as memorization computed on *unseen* hash strings is controlled, the multi-token model perfectly reproduces the training data on *seen* hash strings. We term this *hash-memorization*. An exact equivalence of this phenomenon was noticed in GANs in Nagarajan et al. (2018), where the generator can be trained on specific latent vectors and memorize the mapping on those, and yet produce fresh samples outside of those latent vectors.

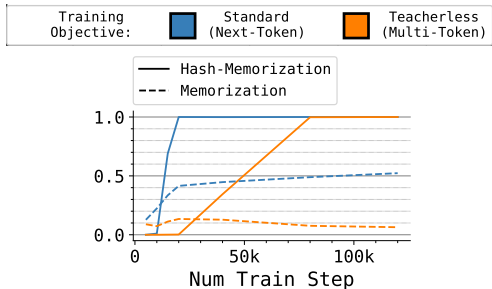


Figure 4: Even if multi-token prediction reduces memorization (on unseen hash strings), it has enough capacity to memorize training data on the seen hash-strings (denoted by hash-memorization). Note that the best creativity score for NTP and MTP are achieved at step 10k and 40k, respectively, which are the checkpoints we used to report metrics in Fig ??.

Effect of token reordering. The implication of our argument in §2.3 is that next-token learning would benefit from reversing the token ordering of the Sibling Discovery task (i.e., parent appears before siblings). Indeed, we find this to be the case in Appendix Fig 6. Interestingly, we find that the reverse-trained model is still far from the original multi-token teacherless model. More surprisingly, a teacherless model trained on the reversed data, achieves even higher creativity of all training methods here. Note that in all other datasets, no reordering of the tokens should make any change to the training.

D DESCRIPTION OF DATASETS

D.1 DATASETS INSPIRED BY COMBINATIONAL CREATIVITY

DATASET 1: SIBLING DISCOVERY. This task is based off a bipartite graph \mathcal{G} made of parent vertices $\mathcal{V} = \{A, B, C, \dots\}$ each neighboring a corresponding set of children $\text{nbr}(A) = \{a_1, a_2, \dots\}$. We set the number of parent vertices $|\mathcal{V}|$ to be small and the number of children for each parent vertex $|\text{nbr}(A)|$ to be large. For example, $|\mathcal{V}| = 5$ and $|\text{nbr}(A)| = 500$. We define $\text{coh}(s)$ to hold on “sibling-parent” triplets of the form $s = (\gamma, \gamma', \Gamma)$ such that $\gamma, \gamma' \in \text{nbr}(\Gamma)$.

Next, we ensure that the training set is large enough for the model to infer all the edges in the graph. Let $m = |\mathcal{V}|$ and $n = |\text{nbr}(\Gamma)|$ (for all $\Gamma \in \mathcal{V}$). This means $S = \Omega(m \cdot n)$. At the same time, to keep the task non-trivial, the training set must be small enough to not cover all the coherent sibling-parent triplets. Thus, we ensure $S = \mathcal{O}(m \cdot n^2)$.

For the default version of this dataset, we set $|\mathcal{V}| = 5$ and $|\text{nbr}(\Gamma)| = 500$ for all $\Gamma \in \mathcal{V}$.

DATASET 2: TRIANGLE DISCOVERY This task is based off an undirected graph $\mathcal{G} = (V, E)$ which contains many triangles. Since a triangle is a symmetric structure, the problem remains the same even upon reordering the vertices. Thus, in this task $\text{coh}((v_1, v_2, v_3)) = \text{true}$ iff all three edges between $\{v_1, v_2, v_3\}$ belong in \mathcal{G} . To make this task interesting (neither too trivial nor too non-trivial) for our models to learn, we enforce several constraints on the graph. First, we try to keep the degree deg of each vertex to be sufficiently small. On the one hand, this is so that no vertex requires too much computation to find a triangle it is part of; on the other, we also do not want a very dense graph where most random triplets are a triangle. The above requirement alone may create vertices that participate in no triangles; so we ensure that each vertex has a minimum number of triangles.

Thus to create a graph that is neither too trivial nor too non-trivial, we define a two-step graph generation procedure. In the first step, we iterate over the vertices, and add deg many edges from that vertex to other vertices in the set (where deg is small, such as 3 or 10). To avoid creating high-degree vertices inadvertently, we only select neighbors with degree $\leq 1.2 \cdot \text{deg}$. Since this may not ensure a sufficient number triangles in each vertex, we then iterate over the vertices, and create tri random triangles on each vertex (where tri is small, such as 6 or 10). We do this by selecting pairs of a vertex’s neighbors and drawing an edge between them.

Next, we want a training dataset such that (a) the model can infer all the edges from the graph and yet (b) not all triangles appear in the dataset. This necessitates training on a dataset that consists not only of a subset of the triangles, but also of edges from the graph. Our training data consists of two parts: (1) 1/3 are random triangles from the graph, (2) 2/3 are random edges from the graph. In the training set, the triangle and edge samples are distinguished by a prefix “triangle:” or “edge:”. During test-time, we ensure that the model is prompted with “triangle:”. A triangle (u, v, w) is tokenized as “tri: $(u, v), (v, w), (w, u)$ ” and an edge (u, v) as “edge: $(u, v), (v, u)$ ”. We provide both the directions of edge to potentially avoid any issues with the reversal curse (Berglund et al., 2024; Allen-Zhu & Li, 2023a).

For the default setting of the dataset, we set $|V| = 999$, $\text{deg} = 3$, $\text{tri} = 6$.

D.2 DATASETS INSPIRED BY EXPLORATORY CREATIVITY

DATASET 3: CIRCLE CONSTRUCTION. In this task, the generated strings must be randomized adjacency lists that can be rearranged to recover circle graphs of N vertices. The vertices come from a fixed vocabulary of M tokens. Specifically, let the generated list be $s = (v_{i_1}, v_{i_2}), (v_{i_3}, v_{i_4}), \dots$. We define $\text{coh}(s) = \text{true}$ iff there exists a *resolving* permutation π such that $\pi(s) = (v_{j_1}, v_{j_2}), (v_{j_2}, v_{j_3}), \dots, (v_{j_n}, v_{j_1})$ for distinct j_1, j_2, \dots, j_n . i.e., each edge leads to the next, and eventually circles back to the first vertex. In our experiments, we set M to be larger than N .

Our default experiments are reported for $N = 9$, $M = 15$.

DATASET 4: LINE CONSTRUCTION This task is a simple variant of the above. We also consider a task where the edge set E corresponds to a line graph. The details are same here except for coherence to hold, we need a resolving permutation π such that $\pi(s) = (v_{j_1}, v_{j_2}), (v_{j_2}, v_{j_3}), \dots, (v_{j_{n-1}}, v_{j_n})$ for distinct j_1, j_2, \dots, j_n . i.e., each edge leads to the next, stopping at a dead-end. We use the same set of hyperparameters as Circle Construction.

Our default experiments are reported for $N = 9$, $M = 15$.

E FURTHER EXPERIMENTAL DETAILS

Details for Gemma v1 (2B) model.

Table 1: **Hyperparameter details for Gemma v1 (2B) model.**

Hyperparameter	Sibling Discovery	Triangle Discovery	Circle Construction	Line Construction
Max. Learning Rate	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-5}
Model Seq. Len.	32	32	2048	2048
Training steps	7500	10k	15k	15k
Training size	50k	15k	10k	10k
Weight given to multi-token obj.	0.5	0.5	0.75	0.75

In Table 1, we provide the hyperparameter details for each of our datasets. We note some common details here. First, the batch size is 4, but each sequence is packed with multiple examples; thus the model sequence length (divided by the input length) can be treated as a multiplicative factor that determines the effective batch size. The learning rates are chosen favorable to next-token prediction (not multi-token prediction). The training steps were chosen roughly based on a point after which the model had saturated in creativity score (and exhibited decreasing creativity). We use a learning rate with linear warm up for 100 steps, followed by cosine annealing upto a factor $0.01 \times$ of the maximum learning rate. To measure creativity, we sample a test dataset T of 1024 datapoints.

We represent the main tokens in our tasks with integers (ranging upwards of 0 to as many distinct integers are required). In the hash-conditioning setting, we use hash strings of default length 10, using randomly sampled uppercase characters from the English alphabet. In all datasets, we space-separate the vertices in a string, and comma-separate the edges.

Details for GPT-2 (86M) model.

We use GPT-2 (small) with 86M non-embedding parameters when we are comparing Transformers with diffusion models. We train these models with a learning rate of 10^{-4} and a batch size of 64, to convergence in terms of the creativity score.

Details for SEDD (90M) model.

We use SEDD’s “absorb” variant, which begins denoising with a fully masked sequence and iteratively refines tokens over 128 denoising steps. This variant achieves the best language modeling performance in the original paper. Same as GPT-2 (86M), we train these models with a learning rate of 10^{-4} and a batch size of 64, to convergence in terms of the creativity score. ¹

¹We use the codebase of (Lou et al., 2023) at <https://github.com/louaaron/Score-Entropy-Discrete-Diffusion>.

F SENSITIVITY ANALYSES FOR GEMMA v1 (2B)

In this section, we report that our observations are robust to the choice of various hyper-parameters. First, we present a series of plots for the Gemma v1 (2B) model; each group of plots reports varying one hyperparameter for all the datasets. Fig 5 for train set size, Fig 6 for task complexity, Fig 7 for the weight given to the multi-token objective (and Fig 8 correspondingly for memorization), Fig 9 for learning rates, Fig 10 for number of training steps and Fig 11 for batch size. In § F.1, we report analyses for varying sampling conditions.

Note on task-complexity. In Fig 6, we report robustness of our results to variations in the task complexity (e.g., degree, path length etc.). Note that the variations we have explored are within reasonable factors. If we vastly increase certain factors (e.g., increase the degree of the vertices), we expect learning to become either highly trivial or non-trivial (see §D for some reasoning). Besides, as discussed in the main paper, teacherless training is a hard objective to optimize especially for smaller models; thus, we expect increasing the task complexity beyond a point to hurt for a fixed model size (for optimization reasons, not generalization reasons).

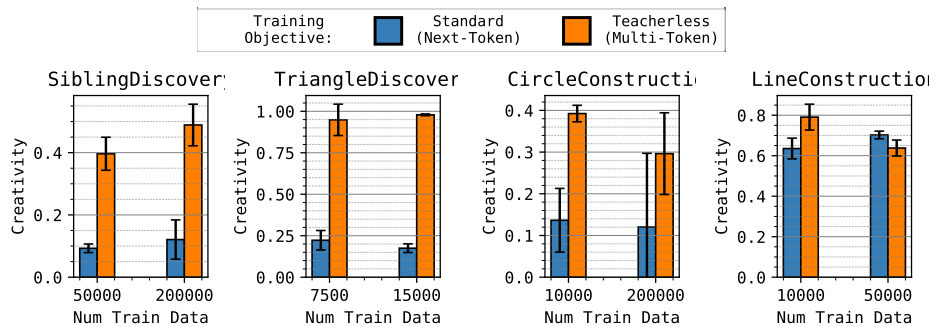


Figure 5: **Training size and creativity score for Gemma v1 (2B):** Creativity score increases under multi-token prediction across various training set sizes. Note though that, in our examples, we expect the gap to diminish eventually with sufficiently many training datapoints (this is unlike the failure of next-token prediction in (Bachmann & Nagarajan, 2024)).

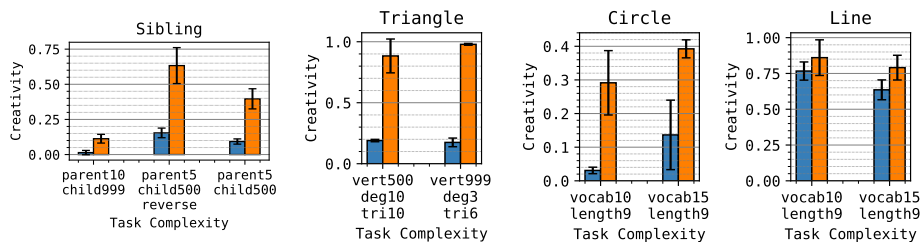


Figure 6: **Task complexity and creativity score for Gemma v1 (2B):** Creativity score increases under multi-token prediction across (reasonable) variations in the dataset parameters (as described in §D).

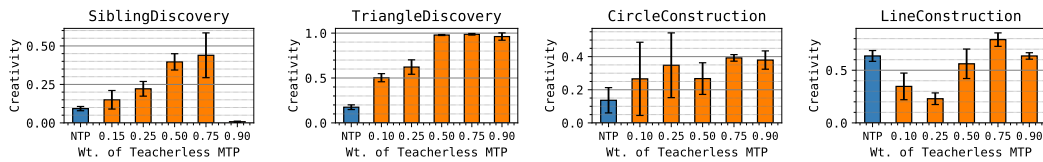


Figure 7: **Weight given to multi-token objective and creativity score for Gemma v1 (2B):** Creativity score increases under multi-token prediction across various weights given to the multi-token component of the objective.

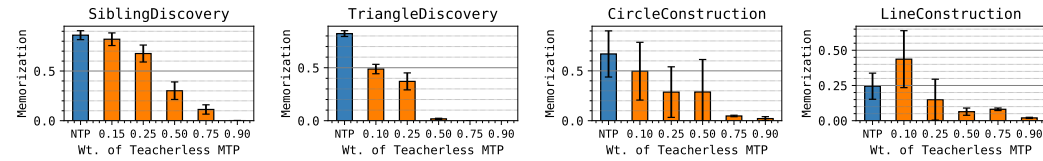


Figure 8: **Weight given to multi-token objective and memorization score for Gemma v1 (2B):** Memorization reduces under multi-token prediction across various weights given to the multi-token component of the objective.

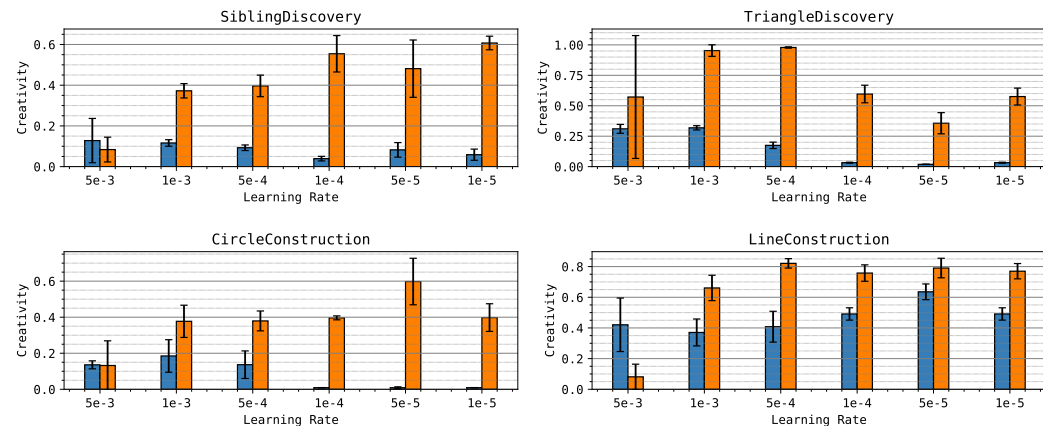


Figure 9: **Learning Rate and creativity score for Gemma v1 (2B):** Creativity score increases under multi-token prediction across various learning rates.

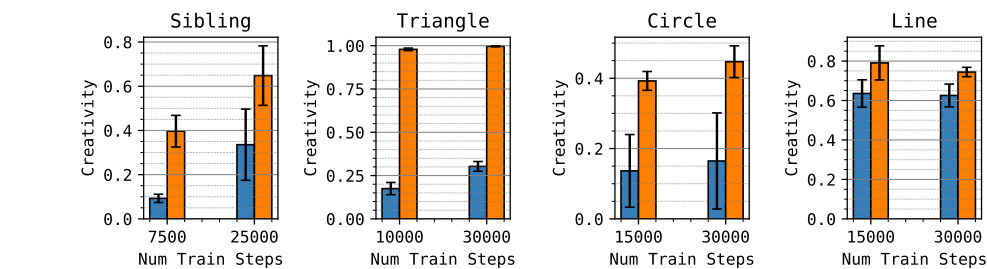
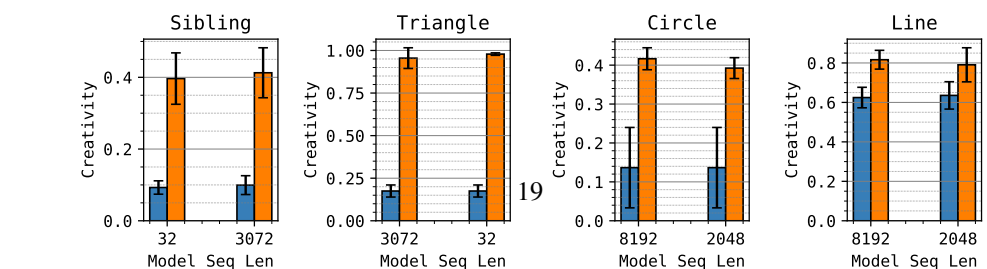


Figure 10: **Training steps and creativity score for Gemma v1 (2B):** Creativity score under multi-token prediction across lengths of training.



972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

F.1 VARYING SAMPLING METHODS

The next three sets of plots report creativity, memorization and coherence (i.e., fraction of generated strings that are coherent) for various sampling methods (greedy decoding and nucleus sampling) with various prefix conditionings (namely, null, pause and hash).

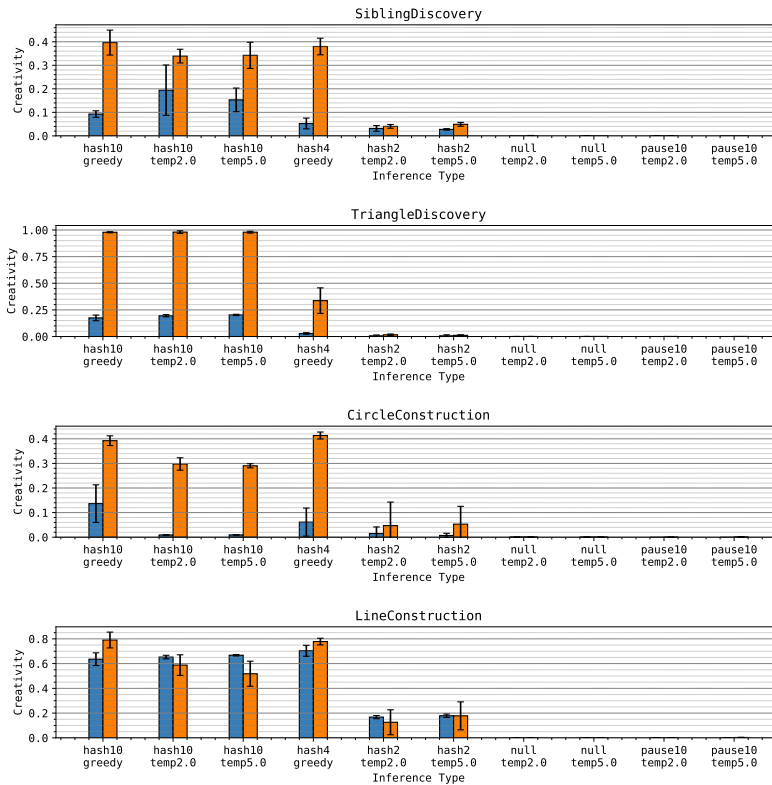


Figure 12: **Creativity under various sampling conditions for Gemma v1 (2B)**: Across all conditions, and in almost all datasets (except Line Construction), multi-token prediction improves creativity. Furthermore, hash-conditioning achieves best creativity scores, with a longer hash helping more.

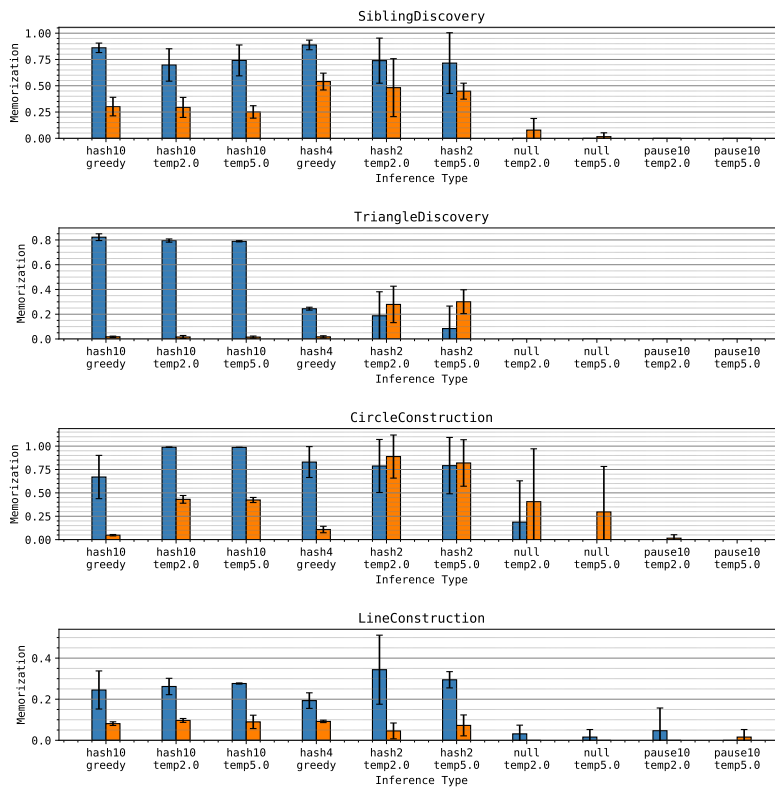


Figure 13: **Memorization under various sampling conditions for Gemma v1 (2B):** Barring a few conditions, the most prominent trend is that memorization reduces under multi-token prediction for various sampling conditions. Observe that the null and pause-conditioned models *do* produce some memorized output while their creativity was non-existent.

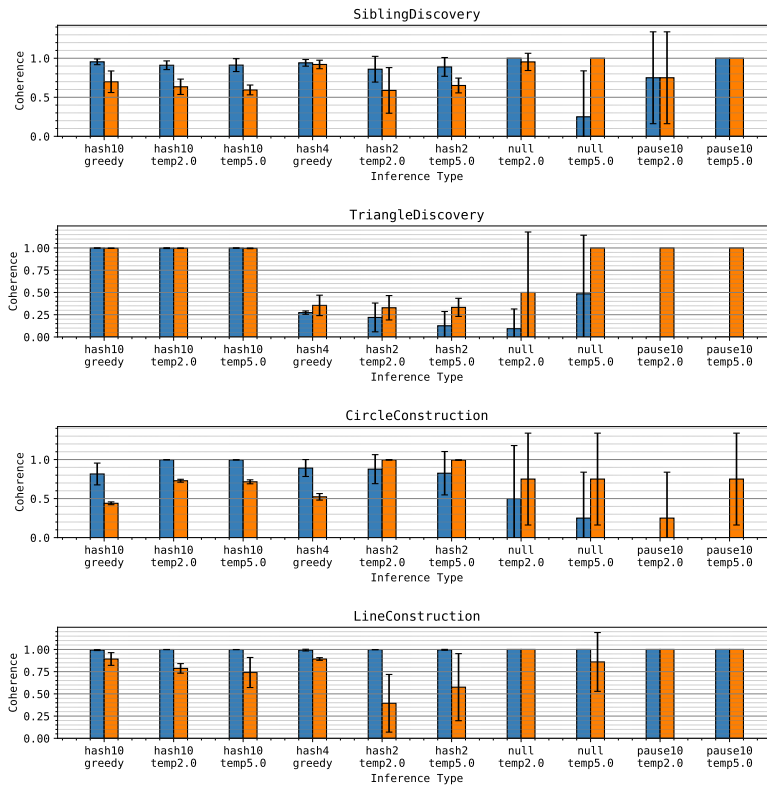


Figure 14: **Coherence under various sampling conditions for Gemma v1 (2B):** Coherence of all models is high or at least noticeable, across various sampling conditions.

G ADDITIONAL EXPERIMENTS IN SEDD (90M) vs. GPT-2 (86M)

In this section, we first provide additional experiments on the sensitivity analysis for SEDD (90M) vs GPT-2 (86M) with different dataset settings. We then provide an ablation study on the hash string length for NTP vs MTP.

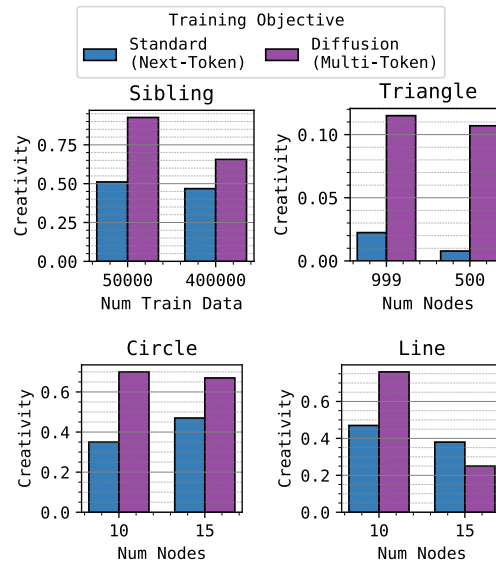


Figure 15: **Sensitivity analysis for the SEDD (90M) model vs. GPT-2 (86M) model:** We report the creativity score for various changes: training set size (top-left) and number of nodes (others).

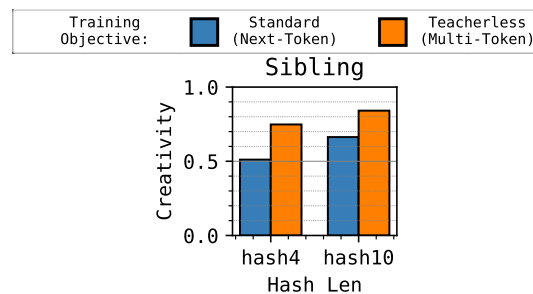


Figure 16: **GPT-2 (86M) Transformer achieves higher creativity with longer hash strings.** We report the creativity scores with hash strings of length 4 and 10, with both NTP and teacherless MTP.

1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295

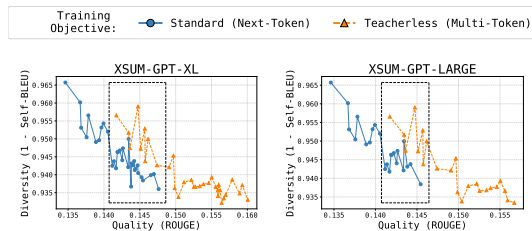


Figure 17: **Multi-token training improves diversity scores for XSUM summarization for large GPT-2 models:** Here, we plot diversity and quality as measured over multiple checkpoints during finetuning, and observe differences in diversity for a fixed quality.

H AN INITIAL EXPLORATION OF REAL-WORLD SUMMARIZATION

For a more realistic examination of our findings, we conduct preliminary investigation of GPT models finetuned with next/multi-token objectives on summarization tasks (XSUM, CNN/DailyMail). We then measure the diversity of a model for any given prompt by generating 5 different completions and computing a Self-Bleu metric.

Admittedly though, a summarization task is not as open-ended as we would like: a higher quality model (i.e., higher Rouge) necessarily means lower diversity. To account for this, we plot how diversity evolves over time as a function of the quality of the model; we then find in Fig 17 that for a given model quality, the larger multi-token models achieve higher diversity (albeit only by a slight amount). This increase does not hold for smaller models and is not always noticeable for CNN/DailyMail. Interestingly, teacherless training consistently shows an increase in summarization quality, measured by Rouge.

Experimental Details . In Table 2, we provide the hyperparameter details for the GPT models finetuned on both XSUM (Narayan et al., 2018) and CNN/DailyMail (Nallapati et al., 2016) for one epoch. We use a learning rate with linear warm up for 0.05 of the total steps, followed by linear decay to 0. To measure Rouge and Self-Bleu, we generate and average across 5 summarizations per document, on a test dataset T of 250 datapoints. We finetune our models with either the next-token prediction objective (Eq 2) or a hybrid of that with the multi-token teacherless objective (Eq 2), with equal weight to both.

Table 2: **Hyperparameter details for summarization experiments.**

Hyperparameter	XSUM	CNN/DailyMail
Batch Size	32	32
Max. Learning Rate	5×10^{-5}	3×10^{-6}
Warmup Steps	338	124
Training Steps	7778	2486
Training Size	248906	79552

To measure quality, we compute the average of Rouge-1, Rouge-2, Rouge-L as Rouge. For measuring diversity, we generate five different summaries per test example, and compute Self-Bleu. This computes average pairwise sentence Bleu-2 scores with weights (0.5, 0.5, 0, 0) on 1- and 2-tuples.

H.1 ADDITIONAL GRAPHS FOR EFFECT OF MULTI-TOKEN TRAINING

Fig 18 shows the diversity and quality graphs on the smaller-sized GPT-2 models on XSUM, and Fig 19 for CNN/DailyMail. While we consistently see improved quality from the multi-token model across the board, we don’t see an increased diversity for fixed Rouge scores anymore.

1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349

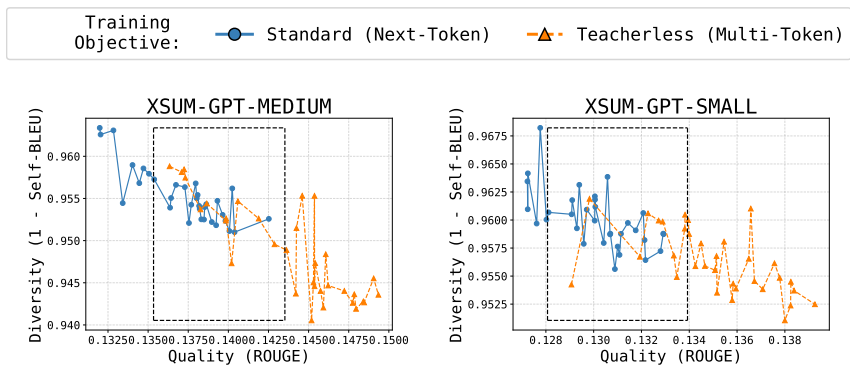


Figure 18: Multi-Token Objective has no effect on diversity for smaller GPT models on XSUM.

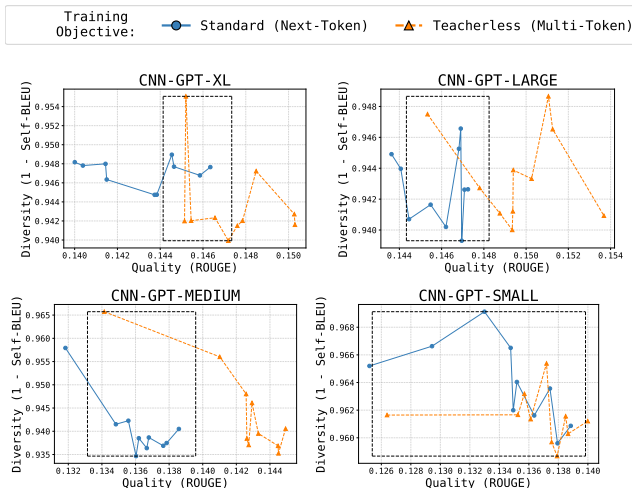


Figure 19: Multi-Token Objective increases diversity for GPT-L and GPT-M but not for GPT-SL or GPT-S on CNN/DailyMail

H.2 EFFECT OF HASH-CONDITIONING

We also conducted hash-conditioning experiments as described in §3. The hash strings we use are 10 randomly sampled uppercase characters from the English alphabet. We report the quality-diversity plots in Fig 20 (for next-token prediction on XSUM) and Fig 21 (for multi-token prediction on XSUM). As such, we do not find any changes in diversity, perhaps because this is not a sufficiently open-ended task.

1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403

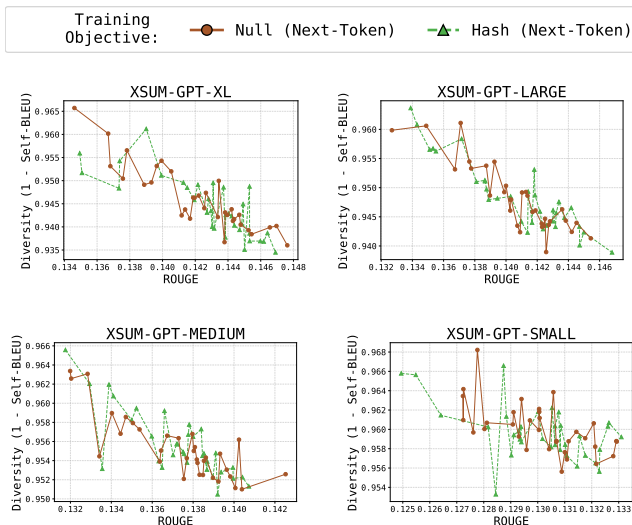


Figure 20: Hash-conditioning has no effect on diversity for GPT models on XSUM summarization with next-token prediction.

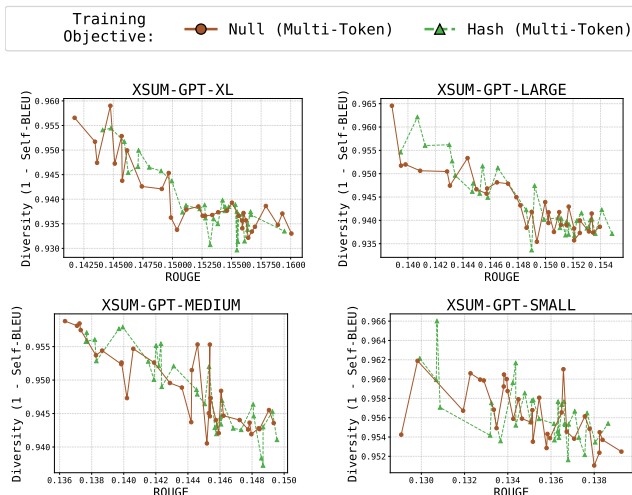


Figure 21: Hash-conditioning has no effect on diversity for GPT models on XSUM summarization with multi-token prediction.

I RELATED WORK

OPEN-ENDED ALGORITHMIC TASKS. Khona et al. (2024); Allen-Zhu & Li (2023b) are most directly related to us as they both study diversity of next-token-trained models on an open-ended algorithmic task. Khona et al. (2024) consider path-connectivity on a knowledge graph. They observe that under temperature-scaling, diversity is at odds with accuracy. Our work shows that this tradeoff can be greatly improved when we consider alternative training methods (multi-token, or hash-conditioning). Allen-Zhu & Li (2023b) empirically demonstrate that next-token predictors are able to learn a synthetic, challenging CFG, in the “infinite” data regime ($\approx 100m$ tokens). Our datasets are not CFGs, with the exception of Sibling Discovery, which can be thought of as a simple PCFG. Our negative result does not contradict theirs since what we show is a sub-optimality of

1404 NTP in a much smaller data regime. Our work also extends the above works by studying limitations
1405 in much more minimal tasks that require as little as 2-hop lookahead.
1406

1407 CRITICISMS OF NEXT-TOKEN PREDICTION (NTP). There has been a recent emerging discussion
1408 surrounding the role of NTP as foundational piece in developing intelligent models. On the critical
1409 side, arguments have been made about the inference-time issues with auto-regression (Dziri et al.,
1410 2024; LeCun, 2024; Kääriäinen, 2006; Ross & Bagnell, 2010). Others have reported the planning and
1411 arithmetic limitations of next-token trained models (McCoy et al., 2023; Momennejad et al., 2023;
1412 Valmeekam et al., 2023a;b;c; Bachmann & Nagarajan, 2024) where the goal is accuracy, not diversity.
1413 Other Transformer failures such as the reversal curse (Allen-Zhu & Li, 2023a) or shortcut-learning in
1414 arithmetic or algorithmic tasks (Dziri et al., 2024; Zhang et al., 2023; Liu et al., 2023; Young & You,
1415 2022; Lai et al., 2021; Ranaldi & Zanzotto, 2023), however these are *out-of-distribution* failures; the
1416 sub-optimality we show is in-distribution, like in (Bachmann & Nagarajan, 2024).

1417 MULTI-TOKEN PREDICTION Recently, there has been growing interest in training language models
1418 beyond NTP. Of relevance to us are elegant ideas such as (Pannatier et al., 2024; Kitouni et al., 2024;
1419 Nolte et al., 2024) which propose applying the next-token objective subject to various permutations of
1420 the token ordering. This should however not resolve the sub-optimality of NTP on our permutation-
1421 invariant tasks. As for diffusion, our findings parallel that of Ye et al. (2024) who show that their
1422 variant of diffusion is able to solve the challenging path-star task of (Bachmann & Nagarajan, 2024).
1423 We also note that Bachmann & Nagarajan (2024) conceptually motivate alternatives to next-token
1424 learning using story-writing, and (Hu et al., 2024) empirically test this; however, their end goal is of
1425 narrative quality rather than creativity across various independent story generations.

1426 There are also other multi-token training approaches like those using independent output heads or
1427 modules (Gloeckle et al., 2024; DeepSeek-AI et al., 2024) or inserting a lookahead attention (Du
1428 et al., 2023). Another line of research is discrete diffusion models (Hoogeboom et al., 2021; Austin
1429 et al., 2021; Gong et al., 2023; Lou et al., 2023), which avoid strict left-to-right factorization by
1430 iteratively refining an entire sequence at multiple positions. There are other models as well, such as
1431 energy-based models Dawid & LeCun (2023) and non-autoregressive models or (Gu et al., 2018).

1432 INJECTING NOISE INTO A TRANSFORMER. Most related to hash-conditioning is DeSalvo et al.
1433 (2024) who induce diversity by varying a *soft*-prompt learned using a reconstruction loss. Our
1434 approach requires no modification to the architecture or the loss; however, we train the whole model,
1435 and not just a soft-prompt generator. The benefits of hash-conditioning may be related to the fact that
1436 varying the wording in a prompt is known to induce diverse outputs (Li et al., 2023; Lau et al., 2024;
1437 Naik et al., 2024). Various works inject noise into a Transformer, in a different form from ours (e.g.,
1438 inducing Gaussian noise), and for a different function such as quality, robustness (Hua et al., 2022;
1439 Jain et al., 2024) or efficiency Wang et al. (2024b).

1440 Our finding that hash-conditioning is superior to temperature sampling echoes Peeperkorn et al.
1441 (2024); Chen & Ding (2023) who find that, in realistic tasks, temperature only has a weak correlation
1442 with creativity, often inadvertently introducing incoherence.
1443

1444 DIVERSITY IN GENERATIVE MODELS. Generative diversity has long been a major goal, at least
1445 until the revolution in reasoning of language models, when accuracy took prominence over diversity.
1446 Much work has gone into concerns such as mode collapse (Che et al., 2017) or posterior collapse
1447 (Bowman et al., 2016) and memorization. In LLMs, regurgitation of training data has been a serious
1448 concern (Carlini et al., 2020; 2023; Nasr et al., 2023).

1449 One line of work relevant to us in the history of generative models is RNN-based VAE for text data
1450 (Bowman et al., 2016). The motivation, like in our work, was to learn high-level semantic features
1451 rather than next-token features with the hope of producing more novel sentence. However, this
1452 suffered from posterior collapse, where the model ignores the latent variable altogether inspiring
1453 various solutions (Yang et al., 2017; Goyal et al., 2017).
1454

1455 EMPIRICAL STUDIES OF CREATIVITY IN LLMs. There is a long line of recent works that measure
1456 novelty and creativity of LLMs and LLM-assisted users. (Chakrabarty et al., 2024; Lu et al., 2024)
1457 quantitatively evaluate and report that models vastly underperform under expert human evaluation
against human writers. Zhang et al. (2024a) argue that finetuning methods such as RLHF and DPO,

1458 are limited when applied to creative humor-generation tasks. Likewise models like GPT4 and Claude
1459 currently underperform top human contestants in generating humorous captions. In poetry, [Walsh et al.](#)
1460 argue that there are certain characteristic styles that ChatGPT restricts itself to. Even assisted-writing
1461 can reduce diversity ([Padmakumar & He, 2024](#)) or produce bland writing ([Mirowski et al., 2024](#)).
1462 On the positive side, [Si et al. \(2024\)](#) report that LLMs surprisingly generate novel research ideas,
1463 although these are less feasible. ([Anderson et al., 2024](#)) find that users tend to produce more divergent
1464 ideas when assisted by ChatGPT (although at a group level, ideas tend to homogenize). Finally,
1465 we refer the reader to [Franceschelli & Musolesi \(2023\)](#) for a rigorous treatment of philosophical
1466 questions surrounding creativity in LLMs. Another line of work ([Wang et al., 2024a](#); [Talmor et al.,](#)
1467 [2020](#); [Zhong et al., 2024](#)) have proposed algorithmic improvements involve creative leaps-of-thought
1468 for real-world tasks.

1469 **LEARNING-THEORETIC STUDIES OF DIVERSITY IN LLMs.** Various theoretical works provide
1470 rigorous arguments for how preventing hallucination and maximizing the model’s coverage are at
1471 odds with each other in abstract settings ([Kalai & Vempala, 2024](#); [Kalavasis et al., 2024](#); [Kleinberg &](#)
1472 [Mullainathan, 2024](#)). We clarify that this tension does not apply in our concrete settings. In those
1473 abstract settings, the strings in the support can be arbitrary and adversarially chosen whereas, our
1474 strings are generated by a simple rule (which can be learned).

1475 Another theoretical question underlying generative models is that the optimum of their objectives are
1476 attained at perfect memorization; yet they tend to produce novel examples e.g., this question has been
1477 posed for GANs in ([Nagarajan et al., 2018](#)) and for diffusion in ([Nakkiran et al., 2024](#)) (see “remarks
1478 on generalization”) or [Kamb & Ganguli \(2024\)](#). Of relevance to us is, [Kamb & Ganguli \(2024\)](#) who
1479 provide a theoretical and empirical argument for how image diffusion models are able to generate
1480 combinatorially many creative outputs; theirs however do not require the type of planning our tasks
1481 do.

1482 **THE NEXT-TOKEN PREDICTION DEBATE.** In support of next-token prediction, there are arguments
1483 ([Shannon, 1948; 1951](#); [Alabdulmohsin et al., 2024](#)) that claim that language is captured by NTP
1484 with models even superceding humans ([Shlegeris et al., 2022](#)) at NTP. There are also and theoretical
1485 results emphasizing the immense expressivity ([Merrill & Sabharwal, 2024](#); [Feng et al., 2023](#)) and
1486 learnability ([Malach, 2023](#); [Wies et al., 2023](#)). of autoregressive Transformers as long as there is a
1487 sufficiently long chain of thought.
1488

1489 **TRANSFORMERS AND GRAPH ALGORITHMIC TASKS.** Graph tasks have been used to understand
1490 various limitations of Transformers in orthogonal settings. [Bachmann & Nagarajan \(2024\)](#); [Saparov](#)
1491 [et al. \(2024\)](#) report that Transformers are limited in terms of learning to search tasks on graphs,
1492 while [Sanford et al. \(2024\)](#) provide positive expressivity results for a range of algorithmic tasks that
1493 process an graph. These works differ from our study of combinational creativity since their graphs
1494 are provided in-context and the tasks have a unique answer. Other works ([Schnitzler et al.](#); [Yang](#)
1495 [et al., 2024a;b](#)) study multi-hop question answering on a knowledge graph; however, this does not
1496 require planning.
1497

1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511