MULTI-TOKEN PREDICTION BOOSTS CREATIVITY IN ALGORITHMIC TASKS

Vaishnavh Nagarajan[*1] **Chen Henry Wu**[* 2] **Charles Ding**[2] **Aditi Raghunathan**[2]

[1]Google Research [2]Carnegie Mellon University
vaishnavh@google.com {chenwu2,clding,aditirag}@cs.cmu.edu

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).

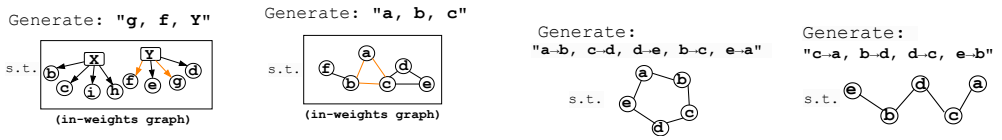
Such leaps, we argue, appear misaligned with the next-token objective. First, a leap must implicitly search, plan and orchestrate various choices — i.e., choices that are *random yet coherent* — to yield diverse yet interesting outputs. Next, even if the training data contained outputs of many

[*]Equal contribution

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).

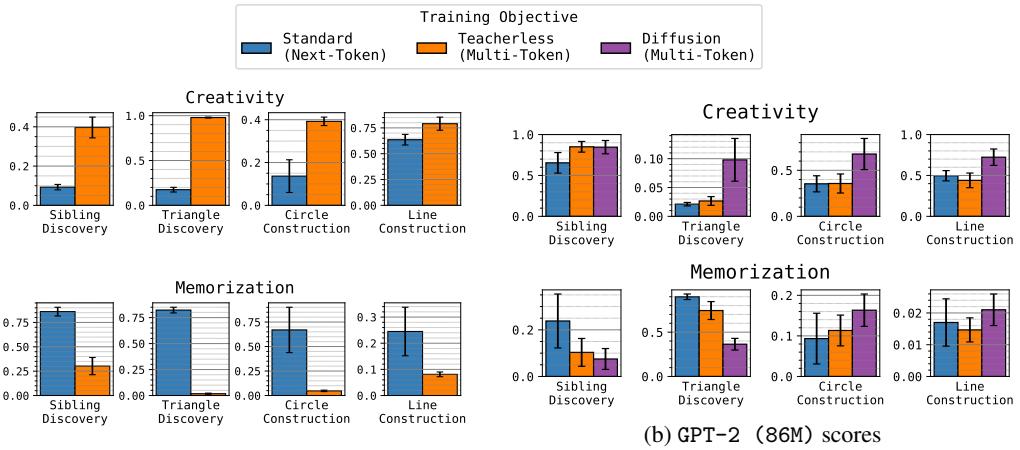


(a) Sibling Discovery (b) Triangle Discovery (c) Circle Construction (d) Line Construction

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 (Hoogeboom 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



(a) `Gemma v1 (2B)` scores

(b) `GPT-2 (86M)` scores

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)`.

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).

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 $\boldsymbol{s}_i \sim \mathcal{D}$. 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 $\texttt{coh} : \mathbb{V}^L \mapsto \{\texttt{true}, \texttt{false}\}$ which corresponds to the support i.e., $\texttt{supp}(\mathcal{D}) = \{\boldsymbol{s} \in \mathbb{V}^L | \ \texttt{coh}(\boldsymbol{s})\}$. 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 $\texttt{mem}_S(\boldsymbol{s})$ denote whether an example $\boldsymbol{s}$ is from the training set S and let the integer function $\texttt{uniq}(X)$ denote the number of unique examples in a set X. We define our (empirical, computational) creativity metric:

$$\hat{\texttt{cr}}_N(T) = \frac{\texttt{uniq}(\{\boldsymbol{s} \in T | \neg \texttt{mem}_S(\boldsymbol{s}) \wedge \texttt{coh}(\boldsymbol{s})\})}{|T|}. \tag{1}$$

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 $\mathcal{V} = \{A, B, C, \ldots\}$ each neighboring a corresponding set of children $\texttt{nbr}(A) = \{a_1, a_2, \ldots, \}, \texttt{nbr}(B) = \{b_1, b_2, \ldots\}$ and so on. We then define $\texttt{coh}(\boldsymbol{s})$ to hold true on "sibling-parent" triplets of the form $\boldsymbol{s} = (\gamma, \gamma', \Gamma)$ such that $\gamma, \gamma' \in \texttt{nbr}(\Gamma)$. 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 $\boldsymbol{s} = (\Gamma, \gamma, \gamma')$, 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 $\texttt{coh}((v_1, v_2, v_3)) = \texttt{true}$ iff all three edges between $\{v_1, v_2, v_3\}$ belong in $\mathcal{G}$. Furthermore, we define $\texttt{uniq}(\cdot)$ and $\texttt{mem}(\cdot)$ 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 $\boldsymbol{s} = (v_{i_1}, v_{i_2}), (v_{i_3}, v_{i_4}), \ldots$. We define $\texttt{coh}(\boldsymbol{s}) = \texttt{true}$ iff there exists a *resolving* permutation π such that $\pi(\boldsymbol{s}) = (v_{j_1}, v_{j_2}), (v_{j_2}, v_{j_3}), \ldots (v_{j_n}, v_{j_1})$ for distinct $j_1, j_2, \ldots j_n$. 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 $\pi(\boldsymbol{s}) = (v_{j_1}, v_{j_2}), (v_{j_2}, v_{j_3}) \ldots, (v_{j_{n-1}}, v_{j_n})$ for distinct $j_1, j_2, \ldots j_n$. 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 $p(\texttt{parent})$, and then figure out $p(\texttt{siblings}|\texttt{parent})$ 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 NPT model learns the parent using a next-token conditional of the form $p(\texttt{parent}|\texttt{siblings})$. 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) 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, $p(\texttt{sibling\_2}|\texttt{sibling\_1})$, 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 $\boldsymbol{z}$, here $\boldsymbol{z} := \texttt{parent}$) and then learning a distribution $p(\boldsymbol{s}|\boldsymbol{z})$ over coherent strings $\boldsymbol{s}$. However, NTP myopically factorizes this into pieces of the form $p(s_i|\boldsymbol{s}_{<i}, \boldsymbol{z})$. 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 $\boldsymbol{p}$ and ground truth sequence $\boldsymbol{s}$, the

model is trained to predict the i'th token s_i, given as input the prompt and all ground truth tokens uptil that point, $(\boldsymbol{p}, \boldsymbol{s}_{<i})$. 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 $\boldsymbol{p}$ (and some dummy tokens in place of the $\boldsymbol{s}$ 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.

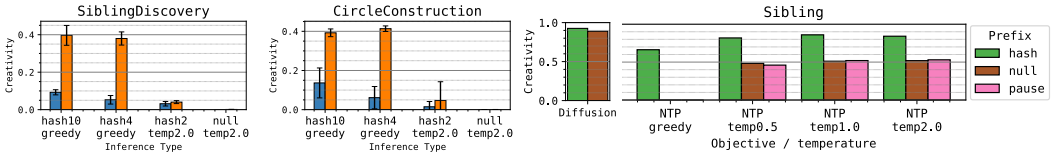
Inference. In all the above techniques, we extract each sample *independently* from the model (as against say, extracting them in continuous succession in the same context). For Transformers, during inference, we perform standard autoregression in both the next- and multi-token trained settings.

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.



(a) `Gemma v1 (2B)` (b) `GPT-2 (86M)`

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.

Csikszentmihalyi, M. *Creativity: Flow and the Psychology of Discovery and Invention*. HarperCollins Publishers, New York, NY, first edition, 1996.

Dawid, A. and LeCun, Y. Introduction to latent variable energy-based models: A path towards autonomous machine intelligence. *arXiv preprint arXiv:2306.02572*, 2023.

DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B.-L., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D.-L., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Zhang, H., Ding, H., Xin, H., Gao, H., Li, H., Qu, H., Cai, J. L., Liang, J., Guo, J., Ni, J., Li, J., Wang, J., Chen, J., Chen, J., Yuan, J., Qiu, J., Li, J., Song, J.-M., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Xu, L., Xia, L., Zhao, L., Wang, L., Zhang, L., Li, M., Wang, M., Zhang, M., Zhang, M., Tang, M., Li, M., Tian, N., Huang, P., Wang, P., Zhang, P., Wang, Q., Zhu, Q., Chen, Q., Du, Q., Chen, R. J., Jin, R. L., Ge, R., Zhang, R., Pan, R., Wang, R., Xu, R., Zhang, R., Chen, R., Li, S. S., Lu, S., Zhou, S., Chen, S., Wu, S.-P., Ye, S., Ma, S., Wang, S., Zhou, S., Yu, S., Zhou, S., Pan, S., Wang, T., Yun, T., Pei, T., Sun, T., Xiao, W. L., Zeng, W., Zhao, W., An, W., Liu, W., Liang, W., Gao, W., Yu, W.-X., Zhang, W., Li, X. Q., Jin, X., Wang, X., Bi, X., Liu, X., Wang, X., Shen, X.-C., Chen, X., Zhang, X., Chen, X., Nie, X., Sun, X., Wang, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yu, X., Song, X., Shan, X., Zhou, X., Yang, X., Li, X., Su, X., Lin, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhu, Y. X., Zhang, Y., Xu, Y., Huang, Y., Li, Y., Zhao, Y., Sun, Y., Li, Y., Wang, Y., Yu, Y., Zheng, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Tang, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y.-B., Liu, Y., Guo, Y., Wu, Y., Ou, Y., Zhu, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Zha, Y., Xiong, Y., Ma, Y., Yan, Y., Luo, Y.-W., mei You, Y., Liu, Y., Zhou, Y., Wu, Z. F., Ren, Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Huang, Z., Zhang, Z., Xie, Z., guo Zhang, Z., Hao, Z., Gou, Z., Ma, Z., Yan, Z., Shao, Z., Xu, Z., Wu, Z., Zhang, Z., Li, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z.-A., Xie, Z., Song, Z., Gao, Z., and Pan, Z. Deepseek-v3 technical report. In *ArXiv*, 2024.

DeSalvo, G., Kagy, J.-F., Karydas, L., Rostamizadeh, A., and Kumar, S. No more hard prompts: Softsrv prompting for synthetic data generation, 2024. URL https://arxiv.org/abs/2410.16534.

Du, L., Mei, H., and Eisner, J. Autoregressive modeling with lookahead attention. *arXiv preprint arXiv:2305.12272*, 2023.

Dziri, N., Lu, X., Sclar, M., Li, X. L., Jiang, L., Lin, B. Y., Welleck, S., West, P., Bhagavatula, C., Le Bras, R., et al. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36, 2024.

Feng, G., Zhang, B., Gu, Y., Ye, H., He, D., and Wang, L. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36, 2023.

Franceschelli, G. and Musolesi, M. On the creativity of large language models. *CoRR*, abs/2304.00008, 2023.

Gloeckle, F., Idrissi, B. Y., Rozière, B., Lopez-Paz, D., and Synnaeve, G. Better & faster large language models via multi-token prediction. 2024.

Gong, S., Li, M., Feng, J., Wu, Z., and Kong, L. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, 2020. doi: 10.1145/3422622. URL https://doi.org/10.1145/3422622.

Goyal, A., Sordoni, A., Côté, M.-A., Ke, N. R., and Bengio, Y. Z-forcing: Training stochastic recurrent networks. *NeurIPS*, 2017.

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.

Gu, J., Bradbury, J., Xiong, C., Li, V. O. K., and Socher, R. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Hofstadter, D. A review of mental leaps: Analogy in creative thought. *AI Mag.*, 16(3):75–80, 1995. doi: 10.1609/AIMAG.V16I3.1154. URL https://doi.org/10.1609/aimag.v16i3.1154.

Holyoak, K. J. and Thagard, P. *Mental leaps: analogy in creative thought*. MIT Press, Cambridge, MA, USA, 1995. ISBN 0262082330.

Hoogeboom, E., Nielsen, D., Jaini, P., Forr'e, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Neural Information Processing Systems*, 2021.

Hopkins, A. K., Renda, A., and Carbin, M. Can LLMs generate random numbers? evaluating LLM sampling in controlled domains. In *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*, 2023. URL https://openreview.net/forum?id=Vhh1K9LjVI.

Hu, E. S., Ahn, K., Liu, Q., Xu, H., Tomar, M., Langford, A., Jayaraman, D., Lamb, A., and Langford, J. Learning to achieve goals with belief state transformers. abs/2410.23506, 2024. doi: 10.48550/ARXIV.2410.23506. URL https://doi.org/10.48550/arXiv.2410.23506.

Hua, H., Li, X., Dou, D., Xu, C., and Luo, J. Fine-tuning pre-trained language models with noise stability regularization. *CoRR*, 2022.

Jain, N., Chiang, P., Wen, Y., Kirchenbauer, J., Chu, H., Somepalli, G., Bartoldson, B. R., Kailkhura, B., Schwarzschild, A., Saha, A., Goldblum, M., Geiping, J., and Goldstein, T. Neftune: Noisy embeddings improve instruction finetuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Kääriäinen, M. Lower bounds for reductions. In *Atomic Learning Workshop*, 2006.

Kalai, A. T. and Vempala, S. S. Calibrated language models must hallucinate. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pp. 160–171. ACM, 2024.

Kalavasis, A., Mehrotra, A., and Velegkas, G. On the limits of language generation: Trade-offs between hallucination and mode collapse. abs/2411.09642, 2024.

Kamb, M. and Ganguli, S. An analytic theory of creativity in convolutional diffusion models, 2024. URL https://arxiv.org/abs/2412.20292.

Khona, M., Okawa, M., Hula, J., Ramesh, R., Nishi, K., Dick, R. P., Lubana, E. S., and Tanaka, H. Towards an understanding of stepwise inference in transformers: A synthetic graph navigation model. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Kitouni, O., Nolte, N., Bouchacourt, D., Williams, A., Rabbat, M., and Ibrahim, M. The factorization curse: Which tokens you predict underlie the reversal curse and more. *CoRR*, abs/2406.05183, 2024. doi: 10.48550/ARXIV.2406.05183. URL https://doi.org/10.48550/arXiv.2406.05183.

Kleinberg, J. M. and Mullainathan, S. Language generation in the limit. *CoRR*, abs/2404.06757, 2024. doi: 10.48550/ARXIV.2404.06757. URL https://doi.org/10.48550/arXiv.2404.06757.

Lai, Y., Zhang, C., Feng, Y., Huang, Q., and Zhao, D. Why machine reading comprehension models learn shortcuts? In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pp. 989–1002. Association for Computational Linguistics, 2021.

Lau, G. K. R., Hu, W., Liu, D., Chen, J., Ng, S.-K., and Low, B. K. H. Dipper: Diversity in prompts for producing large language model ensembles in reasoning tasks, 2024. URL https://arxiv.org/abs/2412.15238.

LeCun, Y. Do large language models need sensory grounding for meaning and understanding? University Lecture, 2024.

Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.-G., and Chen, W. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada, July 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.acl-long.291/.

Liu, B., Ash, J. T., Goel, S., Krishnamurthy, A., and Zhang, C. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

Lou, A., Meng, C., and Ermon, S. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2023.

Lu, X., Sclar, M., Hallinan, S., Mireshghallah, N., Liu, J., Han, S., Ettinger, A., Jiang, L., Chandu, K. R., Dziri, N., and Choi, Y. AI as humanity's salieri: Quantifying linguistic creativity of language models via systematic attribution of machine text against web text. abs/2410.04265, 2024.

Malach, E. Auto-regressive next-token predictors are universal learners. *arXiv preprint arXiv:2309.06979*, 2023.

McCoy, R. T., Yao, S., Friedman, D., Hardy, M., and Griffiths, T. L. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*, 2023.

Merrill, W. and Sabharwal, A. The expressive power of transformers with chain of thought. In *The Twelfth International Conference on Learning Representations*, 2024.

Mirowski, P. W., Love, J., Mathewson, K. W., and Mohamed, S. A robot walks into a bar: Can language models serve as creativity support tools for comedy? an evaluation of llms' humour alignment with comedians. *CoRR*, abs/2405.20956, 2024.

Momennejad, I., Hasanbeig, H., Frujeri, F. V., Sharma, H., Ness, R. O., Jojic, N., Palangi, H., and Larson, J. Evaluating cognitive maps and planning in large language models with cogeval. *Advances in Neural Information Processing Systems*, 36, 2023.

Monea, G., Joulin, A., and Grave, E. Pass: Parallel speculative sampling. *3rd Workshop on Efficient Natural Language and Speech Processing (NeurIPS 2023)*, 2023.

Nagarajan, V., Raffel, C., and Goodfellow, I. J. Theoretical insights into memorization in gans. In *Neural Information Processing Systems Workshop*, volume 1, pp. 3, 2018.

Naik, R., Chandrasekaran, V., Yuksekgonul, M., Palangi, H., and Nushi, B. Diversity of thought improves reasoning abilities of llms, 2024. URL https://arxiv.org/abs/2310.07088.

Nakkiran, P., Bradley, A., Zhou, H., and Advani, M. Step-by-step diffusion: An elementary tutorial, 2024.

Nallapati, R., Zhou, B., dos santos, C. N., Gulcehre, C., and Xiang, B. Abstractive text summarization using sequence-to-sequence rnns and beyond, 2016. URL https://arxiv.org/abs/1602.06023.

Narayan, S., Cohen, S. B., and Lapata, M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization, 2018. URL https://arxiv.org/abs/1808.08745.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., and Lee, K. Scalable extraction of training data from (production) language models. *ArXiv*, 2023.

Nolte, N., Kitouni, O., Williams, A., Rabbat, M., and Ibrahim, M. Transformers can navigate mazes with multi-step prediction. *CoRR*, abs/2412.05117, 2024. doi: 10.48550/ARXIV.2412.05117. URL https://doi.org/10.48550/arXiv.2412.05117.

Padmakumar, V. and He, H. Does writing with language models reduce content diversity? 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=Feiz5HtCD0.

Pannatier, A., Courdier, E., and Fleuret, F. σ-gpts: A new approach to autoregressive models. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2024, Vilnius, Lithuania, September 9-13, 2024, Proceedings, Part VII*, volume 14947 of *Lecture Notes in Computer Science*, pp. 143–159. Springer, 2024.

Peeperkorn, M., Kouwenhoven, T., Brown, D., and Jordanous, A. Is temperature the creativity parameter of large language models? abs/2405.00492, 2024.

Ranaldi, L. and Zanzotto, F. M. Hans, are you clever? clever hans effect analysis of neural systems, 2023.

Ross, S. and Bagnell, D. Efficient reductions for imitation learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, 2010.

Sanford, C., Fatemi, B., Hall, E., Tsitsulin, A., Kazemi, S. M., Halcrow, J., Perozzi, B., and Mirrokni, V. Understanding transformer reasoning capabilities via graph algorithms. abs/2405.18512, 2024.

Saparov, A., Pawar, S., Pimpalgaonkar, S., Joshi, N., Pang, R. Y., Padmakumar, V., Kazemi, S. M., Kim, N., and He, H. Transformers struggle to learn to search, 2024. URL https://arxiv.org/abs/2412.04703.

Schnitzler, J., Ho, X., Huang, J., Boudin, F., Sugawara, S., and Aizawa, A. Morehopqa: More than multi-hop reasoning. abs/2406.13397. doi: 10.48550/ARXIV.2406.13397.

Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3): 379–423, 1948.

Shannon, C. E. Prediction and entropy of printed english. *The Bell System Technical Journal*, 30(1): 50–64, 1951.

Shlegeris, B., Roger, F., Chan, L., and McLean, E. Language models are better than humans at next-token prediction. *arXiv preprint arXiv:2212.11281*, 2022.

Si, C., Yang, D., and Hashimoto, T. Can llms generate novel research ideas? A large-scale human study with 100+ NLP researchers. 2024.

Talmor, A., Tafjord, O., Clark, P., Goldberg, Y., and Berant, J. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Tschannen, M., Kumar, M., Steiner, A., Zhai, X., Houlsby, N., and Beyer, L. Image captioners are scalable vision learners too. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Valmeekam, K., Marquez, M., and Kambhampati, S. Can large language models really improve by self-critiquing their own plans? *arXiv preprint arXiv:2310.08118*, 2023a.

Valmeekam, K., Marquez, M., Olmo, A., Sreedharan, S., and Kambhampati, S. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change, 2023b.

Valmeekam, K., Marquez, M., Sreedharan, S., and Kambhampati, S. On the planning abilities of large language models - A critical investigation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023c.

Varshney, L. R., Pinel, F., Varshney, K. R., Bhattacharjya, D., Schörgendorfer, A., and Chee, Y. A big data approach to computational creativity: The curious case of chef watson. *IBM J. Res. Dev.*, 63 (1):7:1–7:18, 2019.

Walsh, M., Preus, A., and Gronski, E. Does chatgpt have a poetic style? In *Proceedings of the Computational Humanities Research Conference 2024, Aarhus, Denmark, December 4-6, 2024*, volume 3834 of *CEUR Workshop Proceedings*, pp. 1201–1219. CEUR-WS.org.

Wang, H., Zhao, Y., Li, D., Wang, X., Liu, G., Lan, X., and Wang, H. Innovative thinking, infinite humor: Humor research of large language models through structured thought leaps. abs/2410.10370, 2024a.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 13484–13508. Association for Computational Linguistics, 2023.

Wang, Y., Luo, X., Wei, F., Liu, Y., Zhu, Q., Zhang, X., Yang, Q., Xu, D., and Che, W. Make some noise: Unlocking language model parallel inference capability through noisy training. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 12914–12926. Association for Computational Linguistics, 2024b.

Wies, N., Levine, Y., and Shashua, A. Sub-task decomposition enables learning in sequence to sequence tasks. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

Yang, S., Gribovskaya, E., Kassner, N., Geva, M., and Riedel, S. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 10210–10229. Association for Computational Linguistics, 2024a.

Yang, S., Kassner, N., Gribovskaya, E., Riedel, S., and Geva, M. Do large language models perform latent multi-hop reasoning without exploiting shortcuts? abs/2411.16679, 2024b. doi: 10.48550/ARXIV.2411.16679. URL https://doi.org/10.48550/arXiv.2411.16679.

Yang, Z., Hu, Z., Salakhutdinov, R., and Berg-Kirkpatrick, T. Improved variational autoencoders for text modeling using dilated convolutions. *ICML*, 2017.

Yang, Z., Band, N., Li, S., Candès, E. J., and Hashimoto, T. Synthetic continued pretraining. *CoRR*, abs/2409.07431, 2024c. doi: 10.48550/ARXIV.2409.07431. URL https://doi.org/10.48550/arXiv.2409.07431.

Ye, J., Gao, J., Gong, S., Zheng, L., Jiang, X., Li, Z., and Kong, L. Beyond autoregression: Discrete diffusion for complex reasoning and planning. 2024. doi: 10.48550/ARXIV.2410.14157.

Young, T. and You, Y. On the inconsistencies of conditionals learned by masked language models. *arXiv preprint arXiv:2301.00068*, 2022.

Yu, L., Jiang, W., Shi, H., YU, J., Liu, Z., Zhang, Y., Kwok, J., Li, Z., Weller, A., and Liu, W. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=N8N0hgNDRt.

Zhang, H., Li, L. H., Meng, T., Chang, K., and den Broeck, G. V. On the paradox of learning to reason from data. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pp. 3365–3373. ijcai.org, 2023.

Zhang, J., Jain, L., Guo, Y., Chen, J., Zhou, K. L., Suresh, S., Wagenmaker, A., Sievert, S., Rogers, 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.

Zhang, Y., Schwarzschild, A., Carlini, N., Kolter, Z., and Ippolito, D. Forcing diffuse distributions out of language models. abs/2404.10859, 2024b.

Zhong, S., Huang, Z., Gao, S., Wen, W., Lin, L., Zitnik, M., and Zhou, P. Let's think outside the box: Exploring leap-of-thought in large language models with creative humor generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024*, 2024.

A LIMITATIONS

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

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

Some limitations on the discussion on creativity:

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

B TRANSFORMER TRAINING OBJECTIVES

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

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

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

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

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.

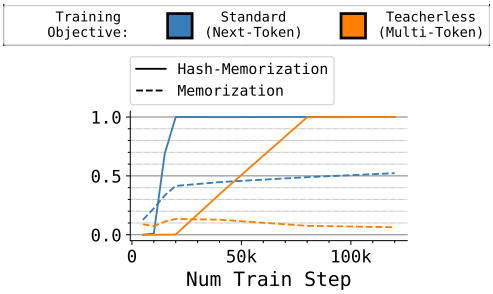


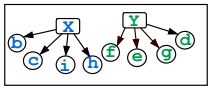
Figure 4: **Even if multi-token prediction reduces memorization (on unseen hash strings), it has enough capacity to memorize training data on the seen hash-strings (denoted by hash-memorization).** Note that the best creativity score for NTP and MTP are achieved at step 10k and 40k, respectively, which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

Effect of token reordering. The implictation 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 8. 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 $\mathcal{V} = \{A, B, C, \ldots\}$ each neighboring a corresponding set of children $\texttt{nbr}(A) = \{a_1, a_2, \ldots, \}$. We set the number of parent vertices $|\mathcal{V}|$ to be small and the number of children for each parent vertex $|\texttt{nbr}(A)|$ to be large. For example, $|\mathcal{V}| = 5$ and $|\texttt{nbr}(A)| = 500$. We define $\texttt{coh}(\boldsymbol{s})$ to hold on "sibling-parent" triplets of the form $\boldsymbol{s} = (\gamma, \gamma', \Gamma)$ such that $\gamma, \gamma' \in \texttt{nbr}(\Gamma)$.



(in-weights graph)

Training data

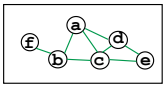
(1) "g, f, Y"
(2) "e, d, Y"
(3) "h, c, X"
(4) "b, i, Y"

Generated samples

(A) "b, f, Y" (incoherent)
(B) "b, i, Y" (memorized ~ (4))
(C) "e, d, Y" (memorized ~ (2))
(D) "e, d, X" (incoherent)
(E) "X, i, b" (incoherent)
(F) "b, c, Y" ✓
(G) "e, g, Y" ✓
(H) "e, g, Y" (duplicated ~ (G))

Algorithmic creativity = 2 / 8

(a) Sibling Discovery



(in-weights graph)

Training data

(1) "tri: ab, bc, ca"
(2) "edge: cd, dc"
(3) "edge: ab, ba"
(4) "edge: bc, cb"
(5) "edge: fb, bf"
(6) "edge: ac, ca"
(7) "tri: da, ac, cd"
(8) "edge: ce, ec"
(9) "edge: ed, de"
(10) "edge: ad, da"

Generated samples

(A) "tri: ab, bc, ca" (memorized ~ (1))
(B) "tri: ab, bf, fa" (incoherent)
(C) "tri: bc, ca, ab" (memorized ~ (1))
(D) "tri: ca, ab, bc" (memorized ~ (1))
(E) "tri: cd, de, ea" ✓
(F) "tri: de, ea, ad" (duplicated ~ (E))
(G) "tri: dc, ce, ed" (duplicated ~ (E))
(H) "tri: bc, ce, eb" (incoherent)
(I) "tri: dc, ca, ad" (memorized ~ (7))
(J) "tri: ac, ce, ea" (incoherent)

Algorithmic creativity = 1 / 10

(b) Triangle Discovery

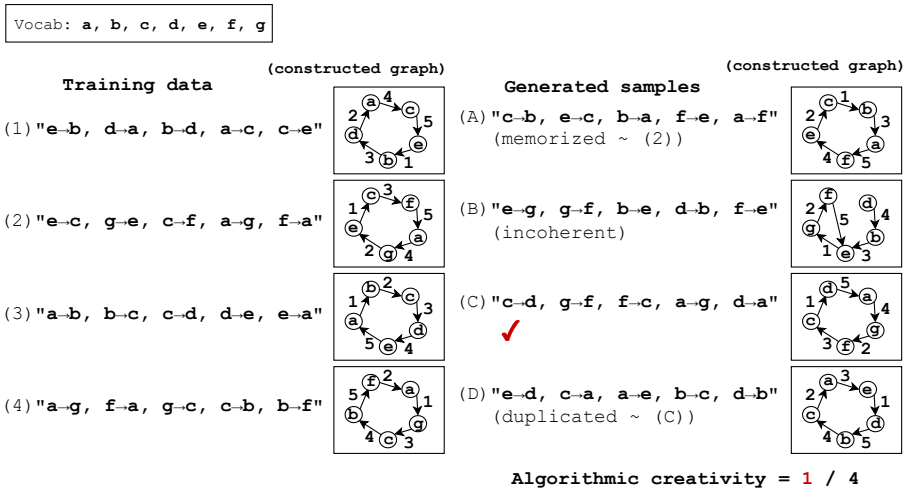
Figure 5: **Minimal tasks inspired by combinational creativity:** The in-weights graph represents the underlying knowledge graph used to generate the training data (not provided in-context). Based on our definition of algorithmic creativity in Eq. (1), generated samples that are incoherent, canonical memorized, or canonical duplicated are not counted as valid samples.

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 $n = |\texttt{nbr}(\Gamma)|$ (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 $S = \mathcal{O}(m \cdot n^2)$.

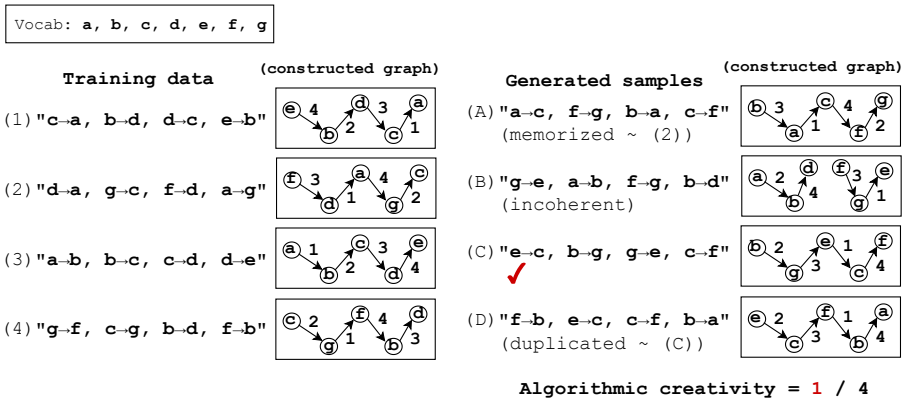
For the default version of this dataset, we set $|\mathcal{V}| = 5$ and $|\texttt{nbr}(\Gamma)| = 500$ for all $\Gamma \in \mathcal{V}$.

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 $\texttt{coh}((v_1, v_2, v_3)) = \texttt{true}$ 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 $\leq 1.2 \cdot \texttt{deg}$. 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.



(a) Circle Construction



(b) Line Construction

Figure 6: **Tasks inspired by exploratory creativity:** The constructed graph visualizes the graph induced by the training or generated sample. Edge indices represent the order of edge appearing in the string. Based on our definition of algorithmic creativity in Eq. (1), generated samples that are incoherent, canonical memorized, or canonical duplicated are not counted as valid samples.

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 $|V| = 999$, $\texttt{deg} = 3$, $\texttt{tri} = 6$.

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 $\boldsymbol{s} = (v_{i_1}, v_{i_2}), (v_{i_3}, v_{i_4}), \ldots$. We define $\texttt{coh}(\boldsymbol{s}) = \texttt{true}$ iff there exists a *resolving* permutation π such

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

that $\pi(\boldsymbol{s}) = (v_{j_1}, v_{j_2}), (v_{j_2}, v_{j_3}), \ldots (v_{j_n}, v_{j_1})$ for distinct $j_1, j_2, \ldots j_n$. 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 $\pi(\boldsymbol{s}) = (v_{j_1}, v_{j_2}), (v_{j_2}, v_{j_3}) \ldots, (v_{j_{n-1}}, v_{j_n})$ for distinct $j_1, j_2, \ldots j_n$. i.e., each edge leads to the next, stopping at a dead-end. We use the same set of hyperparamters as `Circle Construction`.

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

E FURTHER EXPERIMENTAL DETAILS

Details for `Gemma v1 (2B)` model.

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]

[1] We 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 7 for train set size, Fig 8 for task complexity, Fig 9 for the weight given to the multi-token objective (and Fig 10 correspondingly for memorization), Fig 11 for learning rates, Fig 12 for number of training steps and Fig 13 for batch size. In § F.1, we report analyses for varying sampling conditions.

Note on task-complexity. In Fig 8, 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).

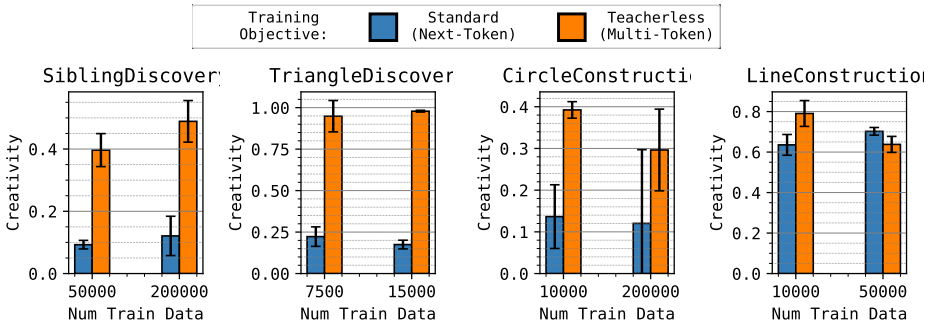


Figure 7: **Training size and creativity score for `Gemma v1 (2B)`:** Creativity score increases under multi-token prediction across various training set sizes. Note though that, in our examples, we except the gap to diminish eventually with sufficiently many training datapoints (this is unlike the failure of next-token prediction in (Bachmann & Nagarajan, 2024)).

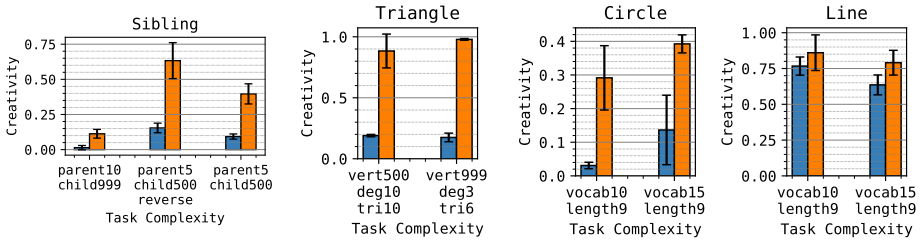


Figure 8: **Task complexity and creativity score for `Gemma v1 (2B)`:** Creativity score increases under multi-token prediction across (reasonable) variations in the dataset parameters (as described in §D).

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).

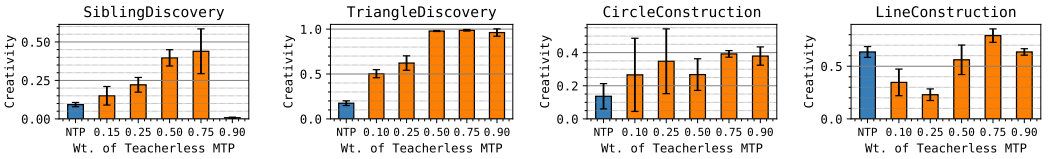


Figure 9: **Weight given to multi-token objective and creativity score for `Gemma v1 (2B)`:** Creativity score increases under multi-token prediction across various weights given to the multi-token component of the objective.

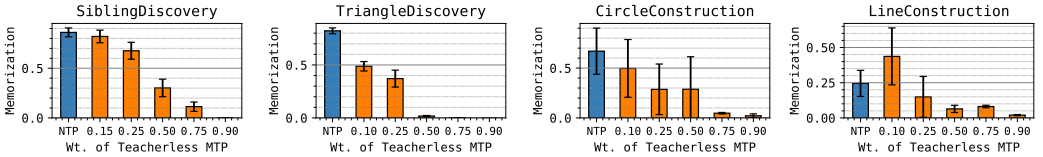


Figure 10: **Weight given to multi-token objective and memorization score for `Gemma v1 (2B)`:** Memorization reduces under multi-token prediction across various weights given to the multi-token component of the objective.

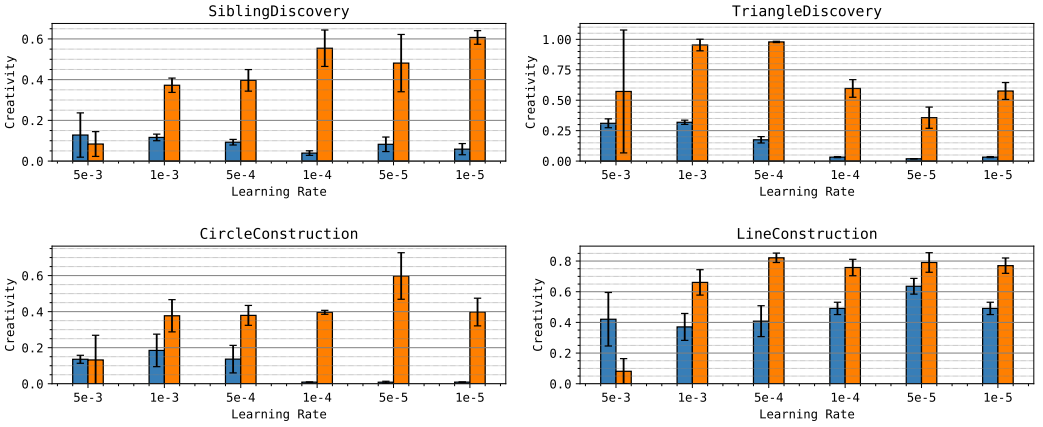


Figure 11: **Learning Rate and creativity score for `Gemma v1 (2B)`:** Creativity score increases under multi-token prediction across various learning rates.

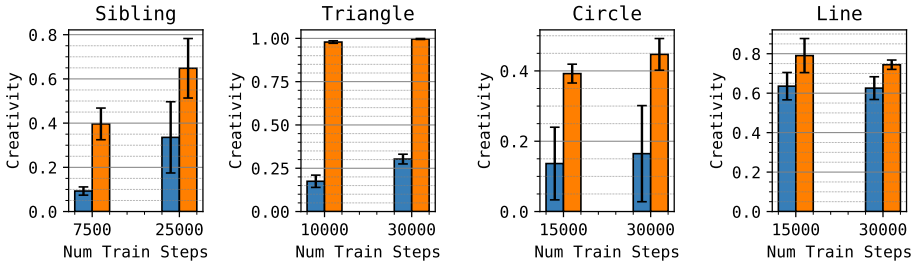


Figure 12: **Training steps and creativity score for `Gemma v1 (2B)`:** Creativity score under multi-token prediction across lengths of training.

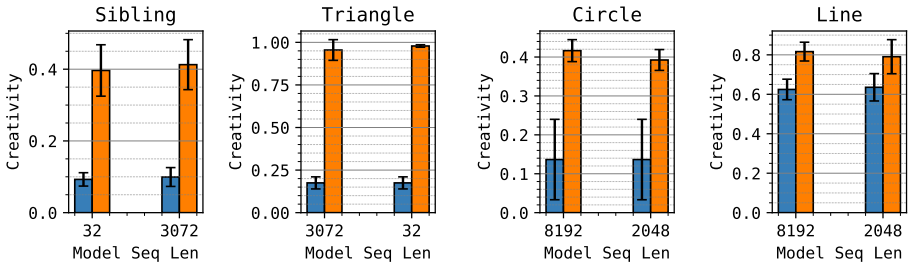


Figure 13: **Batch size and creativity score for `Gemma v1 (2B)`:** Creativity increases under multi-token prediction across various batch sizes. Note that here batch size is effectively proportional to the model sequence length, since we pack multiple finetuning examples into the sequence.

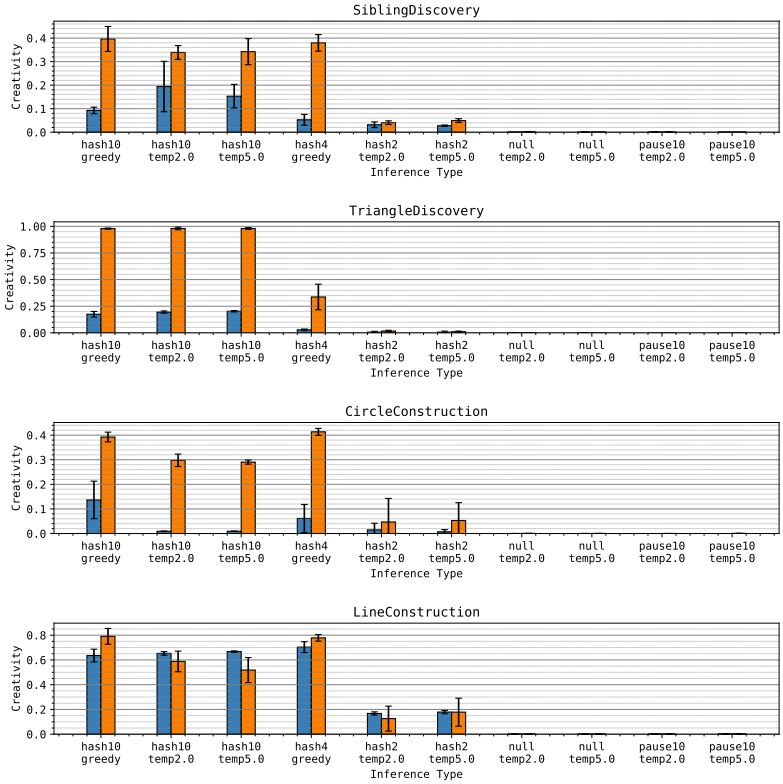


Figure 14: **Creativity under various sampling conditions for `Gemma v1 (2B)`:** Across all conditions, and in almost all datasets (except `Line Construction`), multi-token prediction improves creativity. Furthermore, hash-conditioning achieves best creativity scores, with a longer hash helping more.

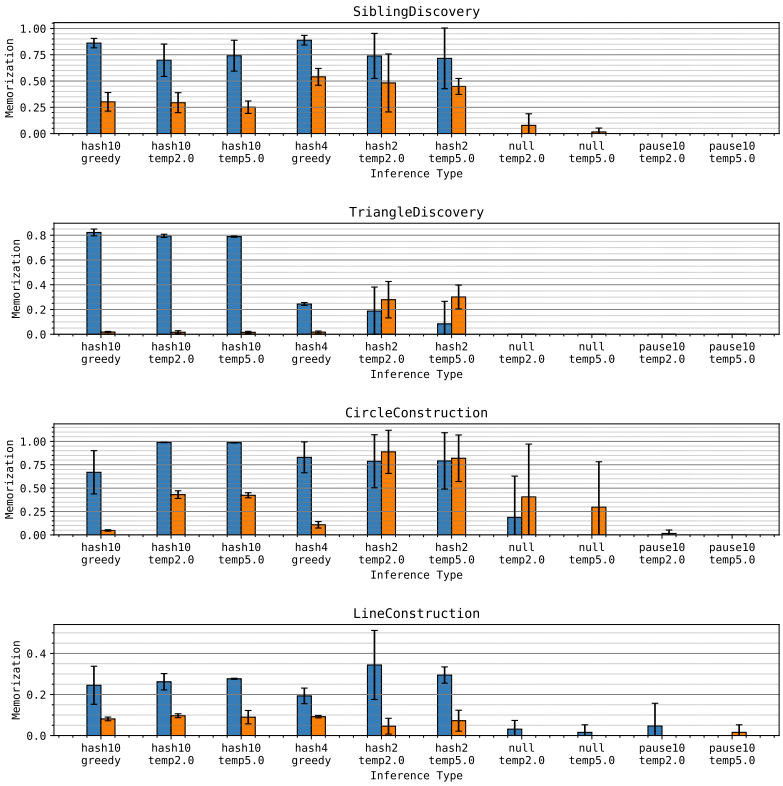


Figure 15: **Memorization under various sampling conditions for `Gemma v1 (2B)`:** Barring a few conditions, the most prominent trend is that memorization reduces under multi-token prediction for various sampling conditions. Observe that the null and pause-conditioned models *do* produce some memorized output while their creativity was non-existent.

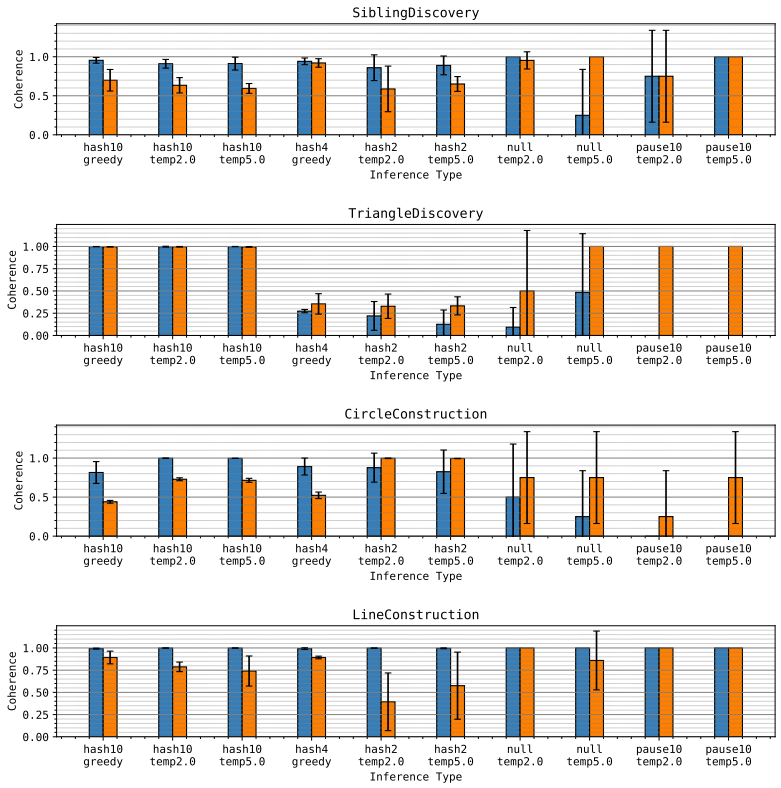


Figure 16: **Coherence under various sampling conditions for `Gemma v1 (2B)`:** Coherence of all models is high or at least noticeable, across various sampling conditions.

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 training and dataset settings (Figure 17 and Figure 18). We then provide an ablation study on the hash string length for NTP vs MTP (Figure 19).

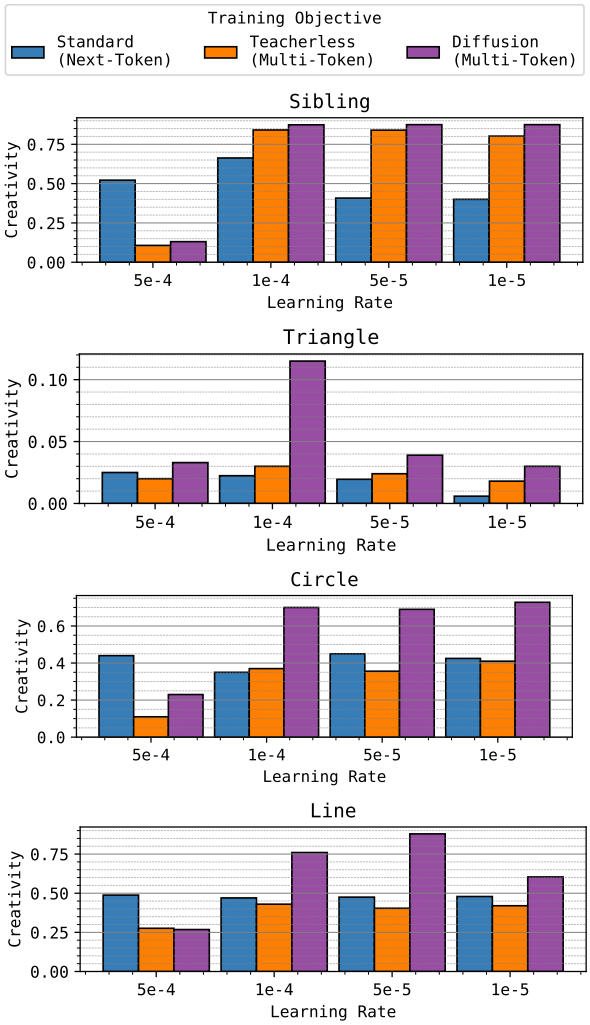


Figure 17: **Learning rate analysis for the SEDD (90M) model vs. GPT-2 (86M):** MTP achieves higher creativity than NTP when both are trained at their optimal learning rates.

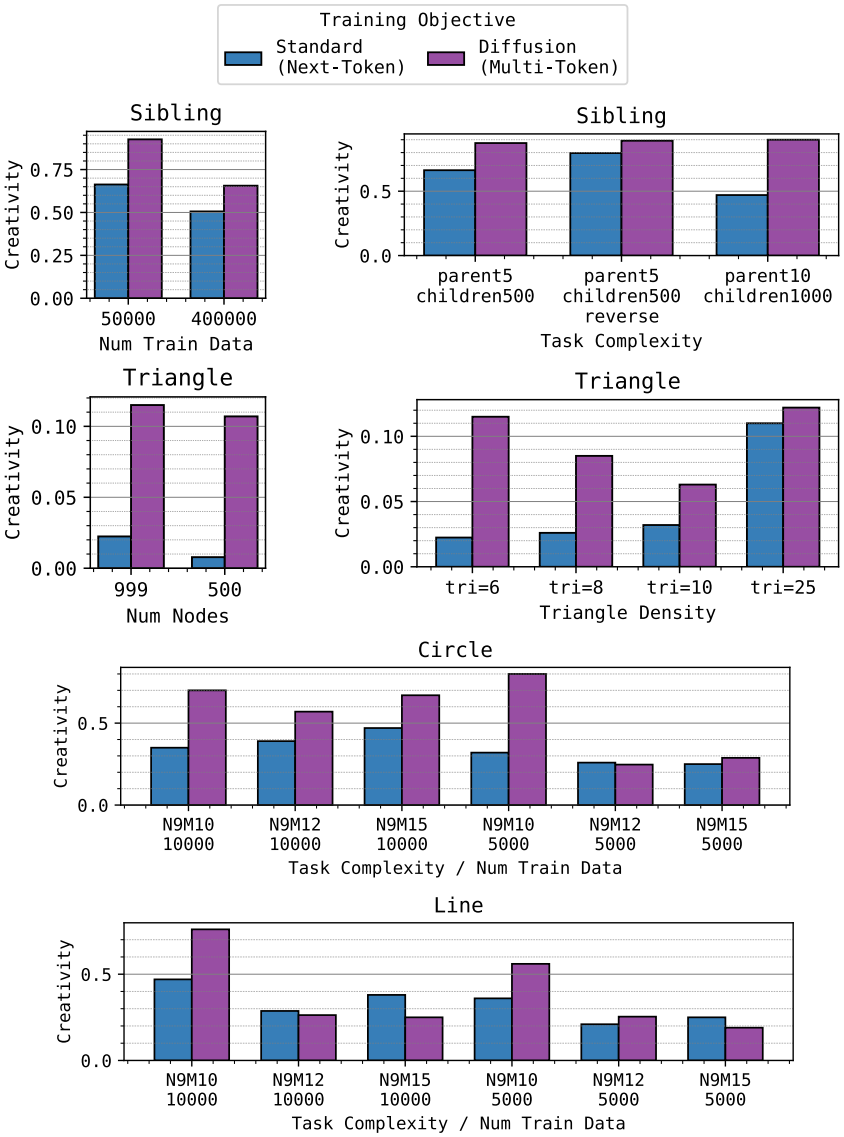


Figure 18: **Sensitivity analysis for the `SEDD (90M)` model vs. `GPT-2 (86M)` model on Sibling and Triangle Discovery tasks:** MTP consistently outperforms NTP under varying task configurations.

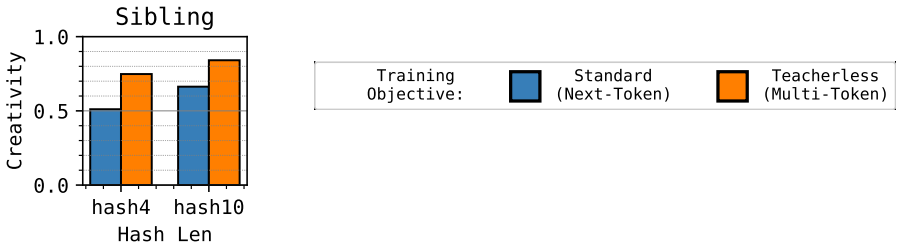


Figure 19: `GPT-2 (86M)` **Transformer achieves higher creativity with longer hash strings.** We report the creativity scores with hash strings of length 4 and 10, with both NTP and teacherless MTP.

H Additional experiments with medium-sized Transformer and SEDD

We replicate our `SEDD (90M)` and `GPT-2 (86M)` experiments on a larger model size (∼400M parameters). In Figure 20, we see similar trends to the smaller model sizes (Figure **??**).

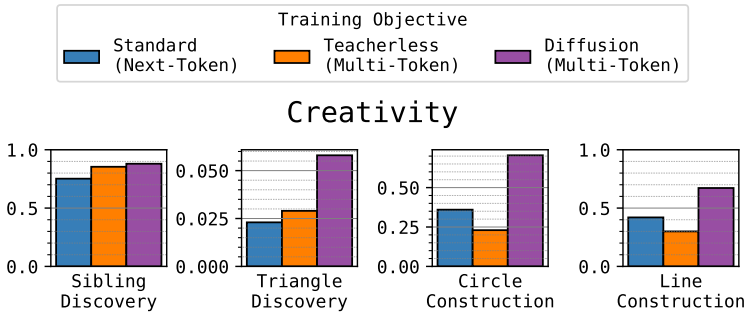


Figure 20: **On a medium-sized (∼400M) model, multi-token diffusion training improves creativity scores from Eq 1 (top) on our four open-ended algorithmic tasks.**

I Creativity vs. Diversity

I.1 Diversity score

Equation (1) defines our creativity score by rewarding samples that are both unique and novel. A higher score can be achieved either by enhancing diversity or by reducing memorization. In the following section, we examine this decomposition using the `Sibling Discovery` task. Formally, we define the diversity score as:

$$\hat{\mathrm{dv}}_N(T) = \frac{\mathtt{uniq}(\{\boldsymbol{s} \in T | \mathtt{coh}(\boldsymbol{s})\})}{|T|}. \tag{4}$$

We first demonstrate that creativity and diversity are not necessarily correlated, and that MTP particularly improves the creativity. To show this, we report the creativity scores and diversity scores along training in Figure 21. We see that for NTP, the diversity score keeps increasing and stays high, while the creativity score increases in the first 10k steps and starts to decrease. For teacherless training, both scores increases throughout training.

I.2 Decompose creativity as diversity and memorization

In this subsection, we aim to understand why a certain method improves creativity. We focus on the two methods that we show improve the creativity scores: (1) teacherless training and (2) hash-conditioning. Better creativity can be achieved either by enhancing diversity or by reducing memorization. In Figure 22, we plot the creativity, diversity, and memorization scores at the

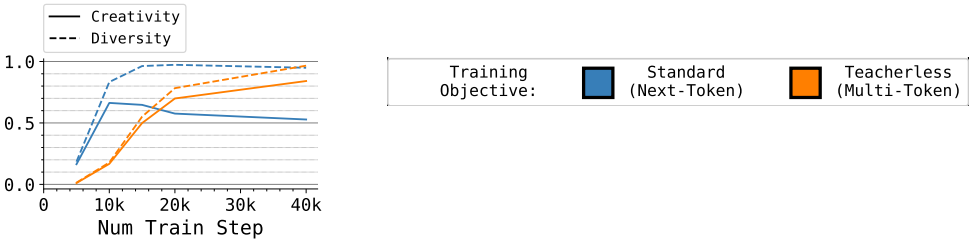


Figure 21: **Creativity and diversity are not necessarily correlated, and MTP particularly improves the creativity:** For NTP, the diversity score keeps increasing and stays high, while the creativity score increases in the first 10k steps and starts to decrease. For MTP, both scores increases throughout training.

checkpoint of best creativity score. We see that both hash string and teacherless training contributes to higher diversity; teacherless training also contributes to reducing memorization. In Figure 21, we see that the best creativity and best diversity are not achieved at the same checkpoint.

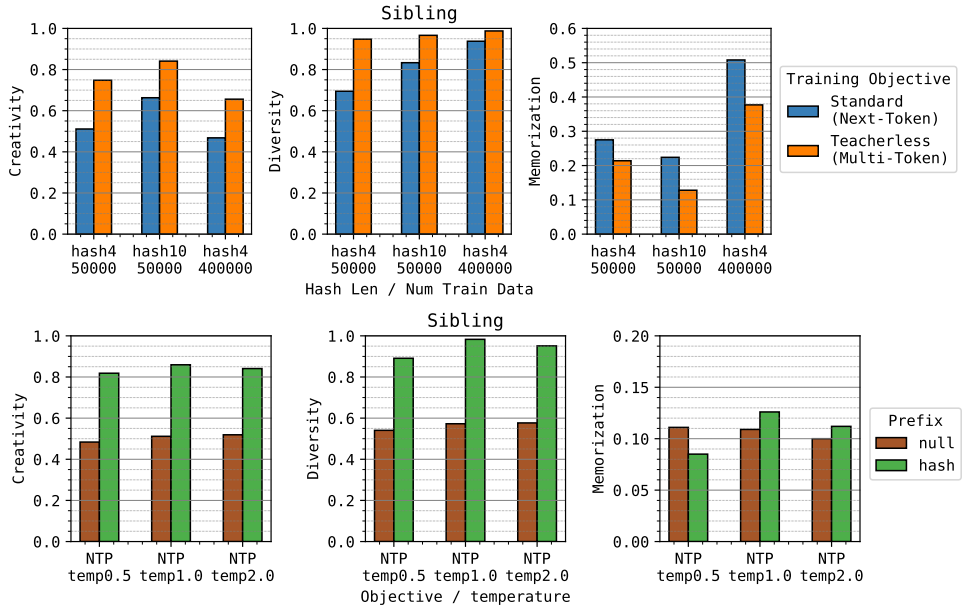


Figure 22: **Decomposition of creativity:** Higher creativity can be decomposed into higher diversity (less duplication) and less memorization. The numbers are reported at the checkpoint of best creativity score. We see that both hash string and teacherless training contributes to higher diversity; teacherless training also contributes to less memorization.

I.3 Data scaling for creativity

How does creativity change as we increase the amount of training data? Intuitively, more training data help the model learn the true distribution, but also make it harder to generate unseen samples. In this subsection, we aim to understand how models perform relative to the *theoretically expected creativity* with different amount of training data. To compute the *theoretically expected creativity*, we uniformly sample coherent samples from all samples *with replacement* to compute the creativity score in Eq. (1). In Figure 23, we see that as we increase the training data (for a fixed underlying graph), the theoretically expected creativity decreases as expected, while the theoretically expected diversity stays the same. MTP narrows the gap between NTP and the theoretically expected creativity and almost achieves the theoretically expected performance in the high data regime.

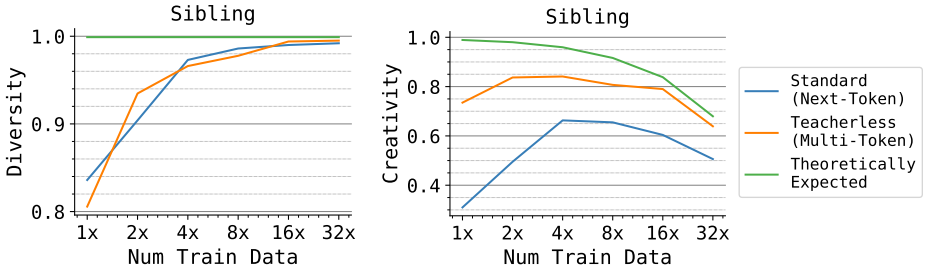


Figure 23: **Data scaling curve for creativity and diversity:** As we increase the training data (for a fixed underlying graph), the theoretically expected creativity decreases as expected, while the theoretically expected diversity stays the same. MTP narrows the gap between NTP and the theoretically expected creativity and almost achieves the theoretically expected performance in the high data regime.

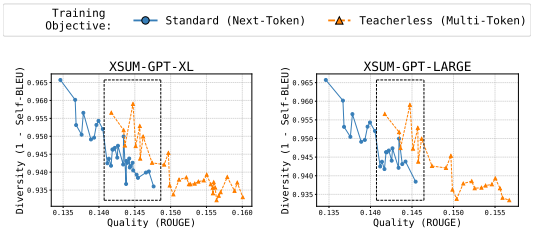


Figure 24: **Multi-token training improves diversity scores for `XSUM` summarization for large `GPT-2` models**: Here, we plot diversity and quality as measured over multiple checkpoints during finetuning, and observe differences in diversity for a fixed quality.

J 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 24 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.

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.

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

J.1 Additional graphs for effect of multi-token training

Fig 25 shows the diversity and quality graphs on the smaller-sized GPT-2 models on XSUM, and Fig 26 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.

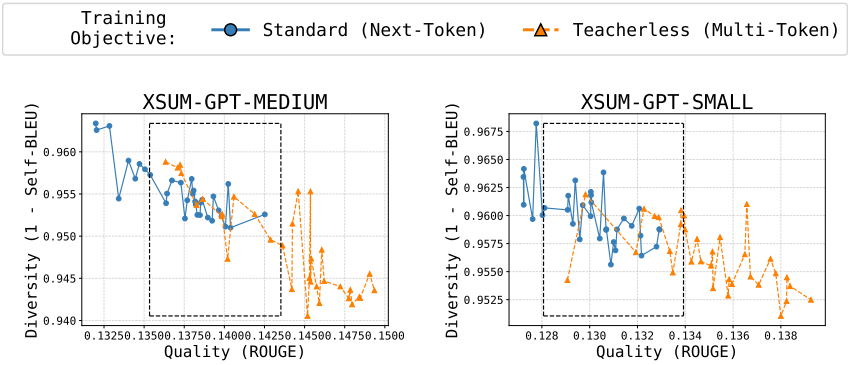


Figure 25: **Multi-Token Objective has no effect on diversity for smaller GPT models on XSUM.**

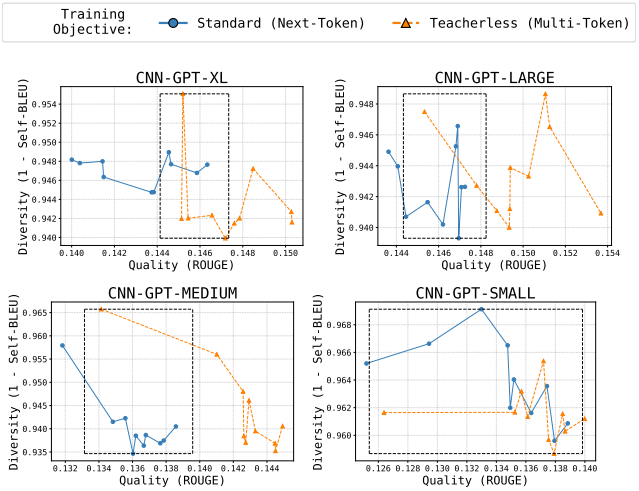


Figure 26: **Multi-Token Objective increases diversity for GPT-L and GPT-M but not for GPT-XL or GPT-S on CNN/DailyMail**

J.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 27 (for next-token prediction on XSUM) and Fig 28 (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.

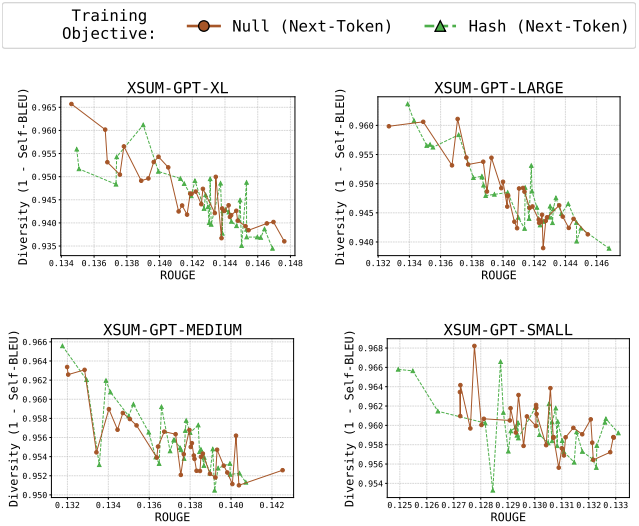


Figure 27: **Hash-conditioning has no effect on diversity for GPT models on XSUM summarization with next-token prediction.**

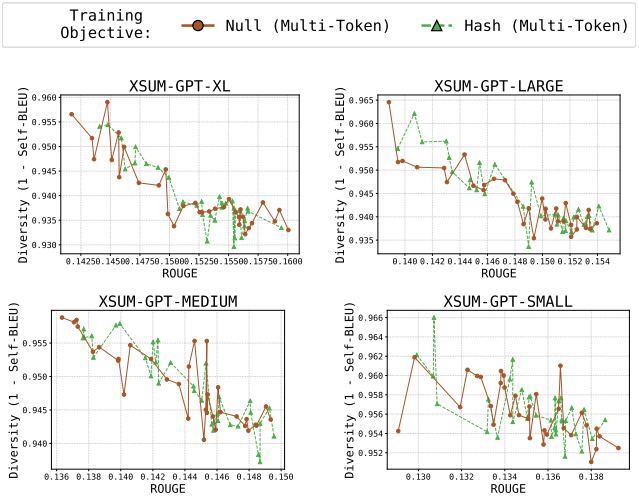


Figure 28: **Hash-conditioning has no effect on diversity for GPT models on XSUM summarization with multi-token prediction.**

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

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

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

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

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

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

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

One line of work relevant to us in the history of generative models is RNN-based VAE for text data (Bowman et al., 2016). The motivation, like in our work, was to learn high-level semantic features rather than next-token features with the hope of producing more novel sentence. However, this

suffered from posterior collapse, where the model ignores the latent variable altogether inspiring various solutions (Yang et al., 2017; Goyal et al., 2017).

EMPIRICAL STUDIES OF CREATIVITY IN LLMS. There is a long line of recent works that measure novelty and creativity of LLMs and LLM-assisted users. (Chakrabarty et al., 2024; Lu et al., 2024) 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, are limited when applied to creative humor-generation tasks. Likewise models like GPT4 and Claude currently underperform top human contestants in generating humorous captions. In poetry, Walsh et al. argue that there are certain characterstic styles that ChatGPT restricts itself to. Even assisted-writing can reduce diversity (Padmakumar & He, 2024) or produce bland writing (Mirowski et al., 2024). On the positive side, Si et al. (2024) report that LLMs surprisingly generate novel research ideas, although these are less feasible. (Anderson et al., 2024) find that users tend to produce more divergent ideas when assisted by ChatGPT (although at a group level, ideas tend to homogenize). Finally, we refer the reader to Franceschelli & Musolesi (2023) for a rigorous treatment of philosophical questions surrounding creativity in LLMs. Another line of work (Wang et al., 2024a; Talmor et al., 2020; Zhong et al., 2024) have proposed algorithmic improvements involve creative leaps-of-thought for real-world tasks.

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

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

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

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