RECROLL: ADAPTIVE DEPTH FIRST SEARCH IN AUTOREGRESSIVE PREDICTIVE SPACE

Mykyta Ielanskyi[1], **Sepp Hochreiter**[1,2]
[1] ELLIS Unit Linz and LIT AI Lab, Institute for Machine Learning,
Johannes Kepler University Linz, Austria
[2] NXAI GmbH, Linz, Austria
{ielanskyi, hochreit}@ml.jku.at

ABSTRACT

One of the compelling difficulties of autoregressive generation is its one way nature. This often leads to drift and overconfidence effects, particularly in language modeling. We introduce RecRoll (Recapitulate and Roll Back) which augments the autoregressive decoding with backtracking that enables models to dynamically overcome the problems of overconfidence by deconditioning on the selected branch of output. This decoding algorithm enables extended test time inference scaling bypassing some of the limitations of models context length. Our approach compartmentalizes model decoding in long-form reasoning, bridging it with depth first search. We show that RecRoll improves outcomes on several challenging reasoning tasks even without fine tuning the models. The inductive bias that our decoding scheme imposes unto the language models resembles branch and bound algorithm and improves performance on tasks which could be solved symbolically in such manner. We further discuss several approaches that could be used to fine tune reasoning models for RecRoll.

1 INTRODUCTION

Chain of Thought (CoT) generation is the go-to inference scaling technique implemented in recent major model releases (Kimi Team et al., 2025; DeepSeek-AI et al., 2025; Olmo et al., 2025). It has enabled a considerable boost to performance of the Large Language Models (LLMs) on a large variety of problems that require more complex solution process rather than simple lookup. CoT generation is a form of inference scaling, meaning that it enables an agent to expend a large amount of computation at test time rather than relying on increasing parameter count or pretraining dataset size to improve the outcome (Snell et al., 2024). In this manner the model is also afforded the decoding budget to conduct a more complete information retrieval from its weights.

Such mode of inference scaling is highly demanding, especially in model architectures that have quadratic time complexity such as attention based models (Vaswani et al., 2017). Additionally, long chain of thought traces can be thrown off by high frequency noise in the model's rollouts (Ginart et al., 2025). These effects are known in the literature relating to modeling dynamic systems (Radler et al., 2026; Hess et al., 2023). Another known issue is the difficulty of obtaining diverse samples in the autoregressive predictive distribution space (Aichberger et al., 2026). A minority of tokens determine the flow of the generation (Wang et al., 2026) and most sampled sequences bear a great deal of similarity. These and other difficulties make the generalist reasoning models struggle in few shot optimization scenarios that can often seem trivial to humans (Chollet et al., 2026).

Several approaches introduce training schemes that leverage parallel generation (Zheng et al., 2025) as well as asynchronous generation approaches (e.g. Noukhovitch et al. (2025)). Parallel CoT decoding schemes (Zheng et al., 2025) are a form of randomized depth first search in which the state is reset to the initial state before backtracking. Some models' training and inference have been explicitly tailored to solve complex problems with high branching factor using relatively small parameter counts and hierachical recursive or latent decoding (Wang et al., 2025; Jolicoeur-Martineau, 2025). An option to interrupt the recurrence dynamically such as an exit condition, is used to improve the test time performance of such approaches.

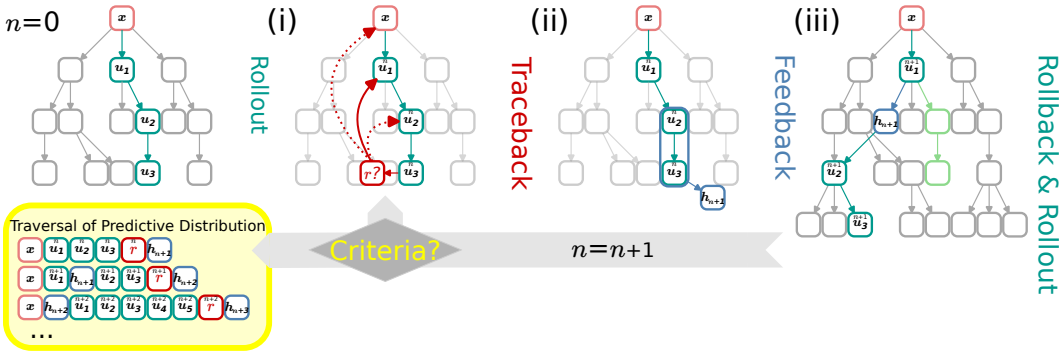

Figure 1: Illustration of the RecRoll decoding. Initially, an autoregressive rollout is generated. Subsequently the traceback r is computed which determines what segment should the output be rolled back to (i). Feedback stage (ii) compresses the information in the selected subbranch of generated outputs into summary $\boldsymbol{h}$. In rollback stage (iii), the output is ablated until the position r and, upon appending the summary $\boldsymbol{h}$, is regenerated autoregressively. The process is then repeated until the stopping criteria is met.

Reasoning post-trained models suffer from degeneration of their predictive distribution (Yue et al., 2025). One can view reasoning fine tuning as trading off the variance of the base models that hypothetically have access to a broader space of generated sequences for bias of the reasoning models which can achieve the satisfactory output in much fewer attempts. We introduce *Recapitulate and Roll Back* (RecRoll) - a novel decoding algorithm for autoregressive models to tackle the above issues. We conduct early experiments to assess the feasibility of this decoding scheme on existing reasoning models.

Our contributions are as follows:

- We introduce RecRoll - a novel algorithmic framework for decoding autoregressive models.
- We compare a simplified instantiation of RecRoll to naive autoregressive generation on a small set of symbolic problems.
- We discuss methodology for effectively fine tuning existing reasoning language models to improve their performance with RecRoll decoding.

2 RecRoll Sampling Algorithm

Definitions. A Markov Decision Process $\mathcal{P} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, p, \gamma)$ where $\mathcal{S}$ is a set of states, $\mathcal{A}$ the set of actions, and $\mathcal{R}$ the reward space. $s_t = (\boldsymbol{x}, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{t-1})$ is the state of order t with $\boldsymbol{x}$ being the original query and each $\boldsymbol{u}_t$ is a generated CoT *segment*. The language model $p(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}, \boldsymbol{w})$ can then be treated as a policy or a transition dynamics in a search space of all possible completions of length up to T. For better intuitive interpretation, we will treat the $\boldsymbol{u}_t$ as segments instead of tokens as individual actions. We use the terms 'step' and 'segment' interchangeably. In the corner cases, the whole generated text would be split into segments consisting of individual tokens or kept together as a large single segment. Segmentation strategies are described in MCTS literature (Kazemnejad et al., 2025). In the subsequent equations, whenever the transitions (e.g. $p(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t)$) are listed without specifying the parameters, $\boldsymbol{w}$ of the generating model are implied.

Criteria, Traceback and Feedback functions. RecRoll introduces three additional mechanisms compared to the ordinary autoregressive generation. We designate them as *criteria*, *traceback* and *feedback*. Criteria maps the state $\boldsymbol{u}_t$ to a boolean action space, indicating the need to terminate the algorithm. Traceback function maps the state $\boldsymbol{u}_t$ to a positive integer r that indicates the number of generated segments to revert or a specific position in the trace to which rollback must be performed. Feedback function compresses the parts that are to be rolled back, which could be the entire CoT trace, into an abstracted form. It is necessary since if we merely roll back the CoT trace to a designated point, we would effectively be implementing resampling of the branch, rather than steering

Algorithm 1 The RecRoll algorithm.

Supplies: Searches the predictive distribution of an autoregressive model for solution and returns the result.

Requires: Autoregressive model $p(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}, \boldsymbol{w})$, maximum number of rollbacks L, summary request prompt $\boldsymbol{h}^{prompt}$, maximum stack depth D.

$\boldsymbol{h}^0 = \emptyset$; $r^0 = 0$; $\boldsymbol{u}^0 = \emptyset$; // initialize the intermediates
for $l = 0$ to $L - 1$ **do**
  $\boldsymbol{u}^{l+1}_{r^l \ldots t} \leftarrow \texttt{completion}(\boldsymbol{h}^l, \boldsymbol{u}^l_{0 \ldots r^l - 1}, \boldsymbol{x}, \boldsymbol{w})$ // complete previous state autoregressively
  $\boldsymbol{u}^{l+1}_{0 \ldots t} \leftarrow \texttt{concatenate}(\boldsymbol{u}^l_{0 \ldots r^l - 1}, \boldsymbol{h}^l, \boldsymbol{u}^{l+1}_{r^l \ldots t})$
  $r^{l+1} \leftarrow \textit{traceback}(\boldsymbol{u}^{l+1}_{0 \ldots t}, \boldsymbol{x}, \boldsymbol{w})$ // determine the amount of 'steps' to roll back
  if *criteria*$(l, r^{l+1}, \boldsymbol{u}^{l+1}_{0 \ldots t}, \boldsymbol{x}, \boldsymbol{w})$ **then**
    return $\{\boldsymbol{u}^*, \boldsymbol{h}^* \ldots\}$ // if the criteria is met, exit early
  end if
  $\boldsymbol{h}^{l+1} \leftarrow$ **feedback**$(r^{l+1}, \boldsymbol{u}^{l+1}_{0 \ldots t}, \boldsymbol{x}, \boldsymbol{w})$ // produce a summary of the 'steps' to be rolled back
end for
return $\{\boldsymbol{u}^*, \boldsymbol{h}^* \ldots\}$ // if the criteria is never met, return all the traversed sequences

Algorithm 2 Depth First Search with Feedback (special case of RecRoll).

Supplies: Searches the predictive distribution of an autoregressive model for solution and returns the result.

Requires: Autoregressive model $p(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}, \boldsymbol{w})$, maximum number of rollbacks L, summary request prompt $\boldsymbol{h}^{prompt}$

$\boldsymbol{h}^0 = \emptyset$ // the summary is none for the first step
for $l = 0$ to $L - 1$ **do**
  $\boldsymbol{y}^{l+1} \sim p(\boldsymbol{y} \mid \boldsymbol{h}, \boldsymbol{x}, \boldsymbol{w})$ // sample a sequence autoregressively
  $\boldsymbol{h}^{l+1} \sim p(\boldsymbol{h}^{l+1} \mid \boldsymbol{h}^{prompt}, \boldsymbol{y}, \boldsymbol{h}, \boldsymbol{x}, \boldsymbol{w})$ // generate feedback for the generated trace
end for
return $\texttt{concatenate}(\boldsymbol{h}^L, \boldsymbol{y}^L)$ // if no solution is recorded explicitly, return the last summary

the generation. At the same time, compressing segments could serve to reduce the effect of any high frequency noise in these segments. Finally, criteria function is used to determine whether the current branch should be used as the models final answer and the iteration stops.

RecRoll decoding algorithm allows us to impose the inductive bias of hierarchical, branch-and-bound problem onto the language model. Such inductive bias is often used in models tailored to solve complex symbolic tasks (Wang et al., 2025). Ideally, the model could dynamically lay out the hierarchy of subgoals and then pass the results of the completed subtasks back to an earlier point of generation. The stopping criteria, although not essential, may permit higher 'dynamic range' for the model, enabling more compute expenditure on more challenging tasks. The iterative nature of the RecRoll algorithm allows to perform the analysis in a smaller sequence length window, easing the memory requirements in case of non-linear model architectures as well as allowing for virtually unlimited inference time computation. This would be practically limited by the amount of information that the model can pass backwards in the summaries.

Case Studies of RecRoll Variants. Semantically Diverse Language Generation (SDLG) sampling algorithm (Aichberger et al., 2026) could be viewed as a special case of RecRoll where the rollback function is set to return the index of the token that is determined by the gradient of the external semantic model and feedback function is set to one that returns the second most likely token at r^{l+1}. Each generated token is treated as an individual step. SDLG manages to prominently improve the entropy estimation on relatively short answers. Another special case is when the rollback function is set to output constant 0. In this scenario, the entire model trace is substituted with its summary at every step l. The algorithm simplifies to Alg. 2. We use this version of the algorithm in our subsequent experiments for simplicity.

Task	Method	Score	Total Toks.	Time Taken (s)	Tokens/s
Sudoku	RecRoll	**0.290**	836,227	**521**	**1605**
	Native CoT	0.075	1,085,627	1781	610
Rotate Matrix	RecRoll	**0.675**	421,179	323	**1303**
	Native CoT	0.561	196,011	**268**	731
Word Sorting	RecRoll	**0.856**	311,475	233	1336
	Native CoT	0.749	105,554	**75**	**1407**

Table 1: Evaluation of RecRoll on three tasks from Reasoning-Gym (Stojanovski et al., 2025). 60 samples were used from each task with 4 more serving as few shot examples. Reasoning model `Qwen3-8B` was used. Native CoT means the standard autoregressive CoT generation.

3 Experiments

We have evaluated RecRoll on 3 problems from the Reasoning Gym dataset (Stojanovski et al., 2025). The problems were picked largely arbitrarily with difficulty and easiness of answer extraction being the important criteria. The RecRoll was performed for 8 iterations with criteria set to no criteria, meaning that the model was forced to recapitulate full 8 times before exiting. The model was prompted appropriately to generate the feedback and accept its reinsertion into the beginning of the reasoning. The RecRoll and Native generation were budget-matched with the maximum possible number of tokens being made equal.

For each of the given problem, 64 examples were generated randomly with 4 of them being used for few shot prompting. For both RecRoll and Native generation the answer candidates were pre-extracted with appropriate regex strings. All candidates were evaluated with the highest score being recorded for each of the problems. The mean score of the model per task was then computed from these scores (Tab. 1).

We can observe that even in the model untrained for RecRoll it provides benefits. It allows a noticeable speedup in sudoku, which is the most challenging of the three tasks while greatly improving the average score. The latter is due to the fact that RecRoll manages the problem of quadratic complexity in transformer models by limiting the search depth and deconditions the later generation from potentially useless branches.

4 Future Work: Fine Tuning RecRoll Specialized Models

While in our experiments we use off-the-shelf reasoning models with minor prompt adjustments together with RecRoll decoding, we could improve the performance if the models were explicitly fine tuned for this decoding scheme. The challenging part in fine tuning the model for RecRoll is to ensure adequate attribution to early branches that may not be exposed to immediate rewards. We propose using intermediate credit assignment techniques (e.g. Khandoga et al. (2026)) to propagate the later rewards into the earlier steps.

The criterion and traceback functions could require auxiliary rewards to train outside of the Alg. 2 setting, where they are trivial. Both functions could benefit from value learning style of rewards similar to Wang et al. (2025). Combining the different reward signal would require advanced reward aggregation technique like GDPO (Liu et al., 2026).

5 Conclusion & Discussion

In this work we have outlined RecRoll - a novel method of decoding autoregressive models. We have provided early evidence of the methods empirical utility, as well as outlined a recipe to fine tune models native to the RecRoll decoding, which may lead to further improvement. The method could allow to alleviate the problems relating to context length of the decoded model and improving the overall decoding speed by confining the model to a relatively short context length region.

ACKNOWLEDGEMENTS

The ELLIS Unit Linz, the LIT AI Lab, the Institute for Machine Learning, are supported by the Federal State Upper Austria. We thank the projects FWF AIRI FG 9-N (10.55776/FG9), AI4GreenHeatingGrids (FFG- 899943), Stars4Waters (HORIZON-CL6-2021-CLIMATE-01-01), FWF Bilateral Artificial Intelligence (10.55776/COE12). We thank NXAI GmbH, Audi AG, Silicon Austria Labs (SAL), Merck Healthcare KGaA, GLS (Univ. Waterloo), TÜV Holding GmbH, Software Competence Center Hagenberg GmbH, dSPACE GmbH, TRUMPF SE + Co. KG.

REFERENCES

Lukas Aichberger, Kajetan Schweighofer, Mykyta Ielanskyi, and Sepp Hochreiter. Semantically Diverse Language Generation for Uncertainty Estimation in Language Models, 2026. URL http://arxiv.org/abs/2406.04306.

Francois Chollet, Mike Knoop, Gregory Kamradt, Bryan Landers, and Henry Pinkard. ARC-AGI-2: A New Challenge for Frontier AI Reasoning Systems, 2026. URL http://arxiv.org/abs/2505.11831.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, and et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, 2025. URL http://arxiv.org/abs/2501.12948.

Antonio A. Ginart, Naveen Kodali, Jason Lee, Caiming Xiong, Silvio Savarese, and John R. Emmons. LZ Penalty: An information-theoretic repetition penalty for autoregressive language models, 2025. URL http://arxiv.org/abs/2504.20131.

Florian Hess, Zahra Monfared, Manuel Brenner, and Daniel Durstewitz. Generalized Teacher Forcing for Learning Chaotic Dynamics, 2023. URL http://arxiv.org/abs/2306.04406.

Alexia Jolicoeur-Martineau. Less is More: Recursive Reasoning with Tiny Networks, 2025. URL http://arxiv.org/abs/2510.04871.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining Credit Assignment in RL Training of LLMs, 2025. URL http://arxiv.org/abs/2410.01679.

Mykola Khandoga, Rui Yuan, and Vinay Kumar Sankarapu. Beyond Uniform Credit: Causal Credit Assignment for Policy Optimization, 2026. URL http://arxiv.org/abs/2602.09331.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, and et al. Kimi k1.5: Scaling Reinforcement Learning with LLMs, 2025.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. GDPO: Group reward-Decoupled Normalization Policy Optimization for Multi-reward RL Optimization. https://arxiv.org/abs/2601.05242v1, 2026.

Michael Noukhovitch, Shengyi Huang, Sophie Xhonneux, Arian Hosseini, Rishabh Agarwal, and Aaron Courville. Asynchronous RLHF: Faster and More Efficient Off-Policy RL for Language Models, 2025. URL http://arxiv.org/abs/2410.18252.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, and et al. Olmo 3, 2025. URL http://arxiv.org/abs/2512.13961.

Andreas Radler, Vincent Seyfried, Johannes Brandstetter, and Thomas Lichtenegger. PAINT: Parallel-in-time Neural Twins for Dynamical System Reconstruction, 2026. URL http://arxiv.org/abs/2510.16004.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters, 2024. URL http://arxiv.org/abs/2408.03314.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. REASONING GYM: Reasoning Environments for Reinforcement Learning with Verifiable Rewards, 2025. URL http://arxiv.org/abs/2505.24760.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, prefix=ukasz useprefix=false family=Kaiser, given=Ł, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical Reasoning Model, 2025. URL http://arxiv.org/abs/2506.21734.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 Rule: High-Entropy Minority Tokens Drive Effective Reinforcement Learning for LLM Reasoning, 2026. URL http://arxiv.org/abs/2506.01939.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?, 2025. URL http://arxiv.org/abs/2504.13837.

Tong Zheng, Hongming Zhang, Wenhao Yu, Xiaoyang Wang, Runpeng Dai, Rui Liu, Huiwen Bao, Chengsong Huang, Heng Huang, and Dong Yu. Parallel-R1: Towards Parallel Thinking via Reinforcement Learning, 2025.