# COrAL: Order-Agnostic Language Modeling for Efficient Iterative Refinement

**Yuxi Xie**[1,2*] **Anirudh Goyal**[3] **Xiaobao Wu**[2,4*] **Xunjian Yin**[2*] **Xiao Xu**[1]
**Min-Yen Kan**[1] **Liangming Pan**[5] **William Yang Wang**[2]
[1] National University of Singapore [2] University of California, Santa Barbara [3] Mila
[4] Nanyang Technological University [5] University of Arizona

## Abstract

Iterative refinement has emerged as an effective paradigm for enhancing the capabilities of large language models (LLMs) on complex tasks. However, existing approaches typically implement iterative refinement at the application or prompting level, relying on autoregressive (AR) modeling. The sequential token generation in AR models can lead to high inference latency. To overcome these challenges, we propose Context-Wise Order-Agnostic Language Modeling (COrAL), which incorporates iterative refinement directly into the LLM architecture while maintaining computational efficiency. Our approach models multiple token dependencies within manageable context windows, enabling the model to perform iterative refinement internally during the generation process. Leveraging the order-agnostic nature of COrAL, we introduce sliding blockwise order-agnostic decoding, which performs multi-token forward prediction and backward reconstruction within context windows. This allows the model to iteratively refine its outputs in parallel in the sliding block, effectively capturing diverse dependencies without the high inference cost of sequential generation. Our findings reveal a quality–speed trade-off, elucidating how COrAL effectively augments the self-enhancement capabilities of conventional autoregressive models without necessitating additional architectural components or extensive pre-training. This work underscores the promise of order-agnostic modeling in advancing LLMs for more efficient and effective natural language processing. Our code is publicly available at https://github.com/YuxiXie/COrAL.

## 1 Introduction

Large language models (LLMs) have recently achieved remarkable success across a wide range of tasks.Strategies that enable LLMs to learn from previous mistakes and iteratively refine their outputs have been particularly effective, achieving human-level performance and transforming both academic research and industrial applications (Pan et al., 2024; OpenAI, 2024). These iterative refinement approaches incorporate feedback—either external or internal—as supervision signals during training (Lightman et al., 2024; Xie et al., 2024), or by developing prompting frameworks that guide the model toward improved generations through

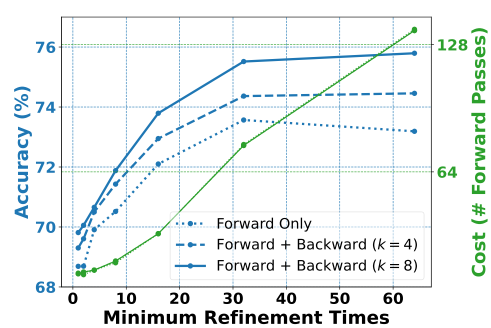



Figure 1: Scaling of performance and inference cost on GSM8K with increasing the minimum refinement times for each output position. $k$ represents the backward context window size. We set the decoding block size as $b = 64$.

*Work done during visiting at UCSB.

Workshop on Adaptive Foundation Models at 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

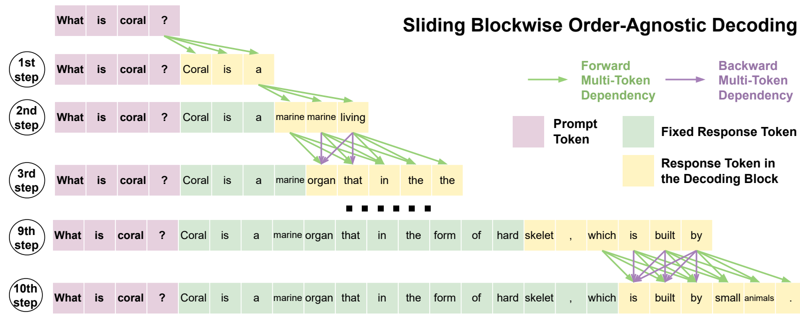



Figure 2: **Sliding Blockwise Order-Agnostic Decoding**. COrAL performs multi-token prediction and refinement in the sliding block with context window size $k=3$ and block size $b=6$.

methods like guided search or self-refine (Yao et al., 2023; Madaan et al., 2023). Despite their effectiveness, these approaches predominantly rely on autoregressive (AR) LLMs, which generate text by predicting the next token in a fixed left-to-right order using causally masked Transformers (Radford, 2018). This sequential generation process inherently limits the model's ability to capture dependencies spanning beyond the immediate next token, especially those that require backward context (Hu et al., 2024). Moreover, the sequential nature of AR models leads to high inference latency, resulting in computational inefficiency for long sequences (Cai et al., 2024).

To address these limitations, researchers have explored order-agnostic architectures that enhance representation learning and accelerate inference. Previous studies mainly focus on two solutions: permutation-based AR and non-autoregressive (NAR) modeling, but each has its own strengths and limitations. For instance, permutation-based models propose diversity-enhanced pretraining objectives that predict multiple subsequent tokens in various orders to capture richer dependencies (Yang et al., 2019; Zhang et al., 2024). Similarly, NAR models generate tokens in parallel, significantly reducing inference time (Gu et al., 2018). However, conventional NAR models often struggle with tasks involving variable-length generation and complex token dependencies, leading to degraded text quality. Given the trade-offs among different models[2], a pivotal question arises: *Can we unify the strengths of denoising techniques with order-agnostic modeling to enhance the capabilities of AR-LLMs while mitigating their respective limitations*?

In this work, we propose **Context-Wise Order-Agnostic Language Modeling** (COrAL), which combines the advantages of AR and order-agnostic modeling. COrAL models token dependencies within manageable context windows, effectively balancing the capture of both local and long-range dependencies with computational efficiency. Through context-wise modeling, COrAL overcomes the limitations of fixed-order generation in AR models and the dependency modeling challenges in NAR models. As shown in Figure 1, this strategy enables the model to perform iterative refinement internally to scale up inference performance. We conduct preliminary experiments on reasoning and code generation tasks to explore the effectiveness and breadth of COrAL.

## 2 Context-Wise Order-Agnostic Language Modeling

We present Context-Wise Order-Agnostic Language Modeling, a generalized AR framework that captures conditional textual distributions based on various orders in context windows.

### 2.1 Objective: Context-Wise Order-Agnostic Autoregressive Modeling

To address the limitations in AR language modeling, we propose Context-Wise Order-Agnostic Language Modeling (COrAL), unifying token-level dependency modeling and sequence-level denoising to advance the capabilities of current LLMs. Previous order-agnostic modeling works attempt to capture various factorization orders involving long dependencies that are difficult to fit. In contrast, COrAL learns the orderless relationships within predetermined context windows. Built on the AR foundation, COrAL leverages the superior capability of sequential language modeling in LLMs.

[2]We make conceptual comparison among different model architectures in Appendix A.

COrAL tackles the problem of generative language modeling by combining forward multi-token prediction with backward denoising in a context-wise and order-agnostic framework. Denoting the context window size as $k$[3], we model the conditional probability distribution of each target token by considering an ensemble of dependencies over all possible positions in the context:

$$\log p_\theta(\boldsymbol{y} \mid \boldsymbol{x}) \geq \sum_{t=1}^{T} \mathbb{E}_{i\in[t-k,t+k]} \mathbb{E}_{l\geq 0} \log p_\theta(y_t \mid \boldsymbol{y}_{\leq i}^{(l)}, \boldsymbol{x}) \tag{1}$$

where $\boldsymbol{y}^{(l)}$ represents an intermediate state of the target output sequence $\boldsymbol{y}$ during iterative refinement. The conventional AR modeling, in comparison, becomes a specific case where only the forward prediction with $k$=1, conditioned on previous tokens in the target sequence $\boldsymbol{y}$, is modeled.

**Forward Prediction and Backward Reconstruction.** We decompose the order-agnostic objective into **forward prediction** and **backward reconstruction**. In forward prediction, COrAL learns to predict multiple future tokens simultaneously given past tokens in the ground-truth sequence. For backward reconstruction, we randomly corrupt tokens in the input sequence to create the intermediate states $\boldsymbol{y}^{(l)}$ in Eq. 1. Similar to BERT (Devlin et al., 2019), we compute the loss only on the corrupted tokens. During training, we use the original data for prediction and the corrupted data for reconstruction. This decomposition disentangles the self-refinement capability from forward prediction, leveraging all data points to enhance sequence modeling.

**Corruption Strategy.** Our corruption and reconstruction process is a form of denoising autoencoding (Vincent et al., 2008) in language modeling. However, instead of representation learning, we aim to endow the model with the self-refinement capability to revise the generated content. Inspired by masked autoencoders (He et al., 2022), we divide the output sequence into non-overlapping patches and randomly sample a subset for corruption. Each patch is a fragment of text containing one or multiple consecutive tokens in the sequence. Specifically, we corrupt a patch by either (**i**) replacing it with a random patch sampled from the current sequence or (**ii**) repeating the first token to replace the other tokens in the patch. This design draws on insight from Ye et al. (2024) that model performance can be significantly improved by simply enhancing consistency across steps.

### 2.2 Architecture: Target-Aware Query Representation for Self-Attention

We build our framework by adapting the standard architecture of LLMs using decoder-only Transformers (Brown et al., 2020). Unlike prior NAR works employing encoder–decoder architectures (Lee et al., 2018; Kasai et al., 2020), the conventional AR foundation predicts the same distribution given the current context regardless of the target token position. While this demonstrates advanced capabilities of sequence modeling and generation, the typical parameterization of next-token distribution constrains its generalizability to the order-agnostic objective in Eq. 1.

Previous works on order-agnostic modeling have explored various ways to incorporate positional information, including scaling up the dimensionality of the final projection layer (Stern et al., 2018) and adding look-ahead tokens (Monea et al., 2023) or extra decoding heads (Cai et al., 2024; Gloeckle et al., 2024). Despite their promising performance, these methods introduce the overhead of additional self-attention network calls and new parameters for multi-position prediction. Instead, we propose a seamless adjustment without adding extra model parameters. Specifically, we apply a generalized Rotary Position Embedding (RoPE) (Su et al., 2024) at the final layer of the decoder-only Transformers to integrate target-aware information into the query representations.

**Target-Aware RoPE.** RoPE encodes positional information into query and key representations, ensuring that their inner product inherently contains relative position information in self-attention: $f(\boldsymbol{q}_m, m)^\top f(\boldsymbol{k}_n, n) = g(\boldsymbol{q}_m, \boldsymbol{k}_n, m-n)$, where $f$ is the positional encoding function applied to the query and key embeddings at $m$-th and $n$-th positions, respectively. Conventional RoPE integrates positional information of the current token to form the query representation. While this effectively enhances the position-aware representation of the input token in intermediate hidden states, it introduces inherent misalignment with the target token position when using the learned representation for output prediction. To avoid this problem, we propose Target-Aware RoPE (Figure 4), which

[3]Without loss of generality, we can set different context window sizes for forward prediction and backward reconstruction in practice. Here, we present the objective with the same hyperparameter $k$ to avoid clutter.

Table 1: Result comparison of performance (accuracy %) and speed (accepted tokens per second) on arithmetic reasoning tasks. We compare against the conventional autoregressive greedy decoding approach as our next-token prediction baseline (NT). "verifier" and "multi-forward" represent the verification stage and multiple forward token prediction in inference.

| Approach | GSM8K | | | MATH | | |
|---|---|---|---|---|---|---|
| | Accu. | Speed | Speedup | Accu. | Speed | Speedup |
| NT | 74.1 | 39.7 | 1.0× | 21.8 | 38.7 | 1.0× |
| Ours | 75.3$\uparrow_{1.2}$ | 43.4 | 1.1× | 22.7$\uparrow_{0.9}$ | 44.4 | 1.1× |
| Ours $_{\text{w/o verifier}}$ | 72.4$\downarrow_{1.7}$ | 156.8 | **3.9**× | 20.0$\downarrow_{1.8}$ | 139.7 | **3.6**× |
| Ours $_{\text{w/o multi-forward}}$ | **78.7**$\uparrow_{4.6}$ | 14.9 | – | **24.3**$\uparrow_{2.5}$ | 11.5 | – |

modifies the positional encoding function at the final layer by considering the target token position in the query representation:

$$f(\boldsymbol{q}_m, \mu)^\top f(\boldsymbol{k}_n, n) = g(\boldsymbol{q}_m, \boldsymbol{k}_n, \mu - n), \quad \mu \in [m-k, m+k] \tag{2}$$

Here, $\mu$ represents the position index of the target token within the context window to be predicted. The rationale behind this modification is that the position encoding in RoPE can adapt the representation of the current token to be tailored for the target position. This simple yet effective adjustment endows the model with the target-aware capability, allowing it to predict tokens at various positions without the overhead of additional entire network calls.

### 2.3 Sliding Blockwise Order-Agnostic Decoding

Leveraging the order-agnostic capabilities of COrAL, we propose Sliding Blockwise Order-Agnostic Decoding, a parallel decoding strategy to enable efficient iterative refinement.

High inference latency significantly hinders the broader application of AR-LLMs. Recent studies have tackled this bottleneck from various angles to accelerate inference. For instance, speculative decoding employs a smaller, faster draft model to propose multiple continuations, which the larger target model then verifies and accepts (Leviathan et al., 2023; Miao et al., 2024). Blockwise parallel decoding directly leverages the large model to generate multiple tokens simutaneously (Stern et al., 2018; Cai et al., 2024). However, these studies increase memory consumption, which thus limits the scalability and impedes distributional deployment. Another promising line of work breaks the sequential dependency by adopting Jacobi decoding (Santilli et al., 2023; Fu et al., 2024) for iterative refinement without architectural add-ons. Kou et al. (2024) propose consistency LLMs to further improve the performance of Jacobi decoding inspired by consistency models (Song et al., 2023).

While these existing approaches improve inference efficiency, they rely on the conventional left-to-right AR foundation with monotonic dependencies. In this work, we leverage the order-agnostic nature of COrAL to perform backward sequence-level refinement and forward multi-token prediction simultaneously, significantly accelerating inference. At each step, we ensemble the output distributions based on multiple possible dependencies and construct a candidate set to fill a block of the output sequence. Furthermore, this process facilitates self-refinement by modifying previous generations at a higher-level horizon, enhancing output quality with advanced inference capabilities. We detail the ensemble strategy for candidate construction and verification in Appendix B.2.

## 3 Experiments

In this section, we demonstrate the efficiency and breadth of COrAL regarding the quality–speed trade-offs across arithmetic, logical reasoning, and code generation. Details of Experimental Setup can be found in Appendx C.

**Arithmetic Reasoning.** As shown in Table 1, COrAL enhances the effectiveness and efficiency through different mechanisms in order-agnostic generation. By ablating the employment of verification and multiple forward token prediction in decoding, COrAL surpasses the corresponding next-token baseline with comparable inference-time cost. Furthermore, by trading inference speed with iterative generation and verification through backward refinement, we observe a substantial

Table 2: Result comparison of performance and speed on logical reasoning tasks.

| Approach | LogiQA | | | ReClor | | |
|---|---|---|---|---|---|---|
| | Accu. | Speed | Speedup | Accu. | Speed | Speedup |
| NT | 55.1 | 33.6 | 1.0× | 63.2 | 33.2 | 1.0× |
| Ours | 58.2↑3.1 | 62.1 | 1.8× | 62.7↓0.5 | 38.2 | 1.2× |
| Ours $_{\text{w/o verifier}}$ | 55.7↑0.6 | 99.1 | **2.9**× | 61.6↓1.6 | 72.0 | **2.2**× |
| Ours $_{\text{w/o multi-forward}}$ | **59.1**↑4.0 | 8.9 | – | **64.7**↑1.5 | 11.3 | – |

Table 3: Result comparison of pass rates and speed on code generation.

| Approach | HumanEval | | |
|---|---|---|---|
| | Pass@1 | Speed | Speedup |
| NT | **64.6** | 42.2 | 1.0× |
| Ours | 13.0↓51.6 | 45.8 | 1.1× |
| Ours $_{\text{w/o verifier}}$ | 6.5↓58.1 | 119.0 | **2.8**× |
| Ours $_{\text{w/o multi-forward}}$ | 61.6↓3.0 | 28.8 | – |

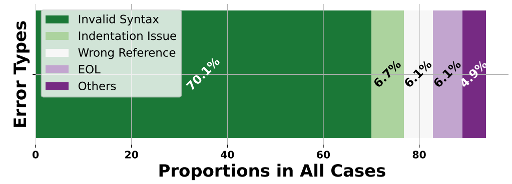



Figure 3: Meso-analysis of error cases in code generation (Ours $_{\text{w/o verifier}}$) on HumanEval. The primary failure cases come from syntax errors.

improvement in accuracy from 74.1% and 21.8% to 78.7% and 24.3% on GSM8K and MATH, respectively. When skipping the verification stage for quality control, our approach significantly speeds up the decoding process up to 3.9×. This demonstrates the flexibility of COrAL in enhancing both the generation quality and inference speed in mathematical reasoning.

**Logical Reasoning.** Table 2 compares the performance and generation speed of model outputs under different decoding settings on logical reasoning tasks. Similarly, COrAL improves the reasoning performance by augmenting next-token prediction exclusively with backward refinement. However, we observe a discrepancy in the performance improvements on LogiQA and ReClor with absolute increases of 4.0% and 1.5% in corresponding accuracies. We attribute this gap to the imbalanced proportions of the two tasks in our SFT data from LogiCoT (Liu et al., 2023b). This also implies the importance of high-quality data selection to boost the effect of order-agnostic training to model different dependencies related to the target tasks.

**Code Generation.** Results on code generation, however, show an opposite effect of order-agnostic modeling on performance. In Table 3, we observe substantial performance drops across different decoding settings using COrAL. For example, without verification, the pass rate on HumanEval decreases to 6.5% from 64.6% of next-token prediction. This gap remains to be large when applying verification for quality control. Error analysis in Figure 3 indicates that the major cause of this drop comes from the erroneous syntax, where the primary error type, *Invalid Syntax*, accounts for 70.1% of the failure cases. To mitigate this issue, we can turn off the mechanism of forward multi-token prediction and increase the threshold $\epsilon$ in Eq. 6 to reject tokens with low confidence scores. For example, with $\epsilon = 0.5$, COrAL achieves a comparable pass rate of 61.6% compared to 64.6% of the baseline. The absolute decrease of 3.0% indicates the deficiency of COrAL in producing incoherent content, showing the importance of specific designs for tasks requiring strict textual formats.

## 4 Discussion and Conclusion

By unifying denoising with context-wise order-agnostic language modeling and introducing target-aware positional encoding, COrAL incorporates iterative refinement directly into the language generation process while keeping inference costs low. This approach offers a promising direction for developing more efficient and capable large language models by effectively capturing local dependencies within context windows and reducing inference latency. The effectiveness and efficiency of COrAL underscores the promise of order-agnostic strategies as a generalized architecture to facilitate generative language modeling and text generation. Specifically, it suggests new opportunities to unify: (**i**) the sequence modeling and varying-length generation abilities of autoregressive modeling, (**ii**) the multi-dependency modeling and multi-token prediction mechanisms in order-agnostic modeling, and (**iii**) the efficient way of iterative refinement in denoising techniques.

## References


Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=PEpbUobfJv.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin T. Vechev. Controlled text generation via language model arithmetic. 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=SLw9fp4yI6.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. Break the sequential dependency of LLM inference using lookahead decoding. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=eDjvSFOkXw.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=pEWAcejiU2.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=B1l8BtlCb.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 15979–15988. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01553. URL https://doi.org/10.1109/CVPR52688.2022.01553.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin

Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

John Hewitt, Christopher D. Manning, and Percy Liang. Truncation sampling as language model desmoothing. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 3414–3427. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-EMNLP.249. URL https://doi.org/10.18653/v1/2022.findings-emnlp.249.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Edward J. Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. Amortizing intractable inference in large language models. 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=Ouj6p4ca60.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. Non-autoregressive machine translation with disentangled context transformer. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5144–5155. PMLR, 2020. URL http://proceedings.mlr.press/v119/kasai20a.html.

Siqi Kou, Lanxiang Hu, Zhezhi He, Zhijie Deng, and Hao Zhang. Cllms: Consistency large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=8uzBOVmh8H.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=UYneFzXSJWh.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 1173–1182. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1149. URL https://doi.org/10.18653/v1/d18-1149.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *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. 12286–12312. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.687. URL https://doi.org/10.18653/v1/2023.acl-long.687.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. 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=v8L0pN6EOi.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. Logiqa 2.0 - an improved dataset for logical reasoning in natural language understanding. *IEEE ACM Trans. Audio Speech Lang. Process.*, 31:2947–2962, 2023a. doi: 10.1109/TASLP.2023.3293046. URL https://doi.org/10.1109/TASLP.2023.3293046.

Hanmeng Liu, Zhiyang Teng, Leyang Cui, Chaoli Zhang, Qiji Zhou, and Yue Zhang. Logicot: Logical chain-of-thought instruction tuning. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pp. 2908–2921. Association for Computational Linguistics, 2023b. doi: 10.18653/V1/2023.FINDINGS-EMNLP.191. URL https://doi.org/10.18653/v1/2023.findings-emnlp.191.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *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. URL http://papers.nips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, Chunan Shi, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In Rajiv Gupta, Nael B. Abu-Ghazaleh, Madan Musuvathi, and Dan Tsafrir (eds.), *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3, ASPLOS 2024, La Jolla, CA, USA, 27 April 2024- 1 May 2024*, pp. 932–949. ACM, 2024. doi: 10.1145/3620666.3651335. URL https://doi.org/10.1145/3620666.3651335.

Giovanni Monea, Armand Joulin, and Edouard Grave. Pass: Parallel speculative sampling. *CoRR*, abs/2311.13581, 2023. doi: 10.48550/ARXIV.2311.13581. URL https://doi.org/10.48550/arXiv.2311.13581.

OpenAI. Learning to reason with llms, 2024. URL https://openai.com/index/learning-to-reason-with-llms/. Accessed: 2024-09-12.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. Automatically correcting large language models: *Surveying the Landscape of Diverse Automated Correction Strategies*. *Trans. Assoc. Comput. Linguistics*, 12:484–506, 2024. doi: 10.1162/TACL_A_00660. URL https://doi.org/10.1162/tacl_a_00660.

Alec Radford. Improving language understanding by generative pre-training. 2018.

Andrea Santilli, Silvio Severino, Emilian Postolache, Valentino Maiorca, Michele Mancusi, Riccardo Marin, and Emanuele Rodolà. Accelerating transformer inference for translation via parallel decoding. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *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. 12336–12355. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.689. URL https://doi.org/10.18653/v1/2023.acl-long.689.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 10107–10116, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/c4127b9194fe8562c64dc0f5bf2c93bc-Abstract.html.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/J.NEUCOM.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Benigno Uria, Iain Murray, and Hugo Larochelle. A deep and tractable density estimator. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pp. 467–475. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/uria14.html.

Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *J. Mach. Learn. Res.*, 17:205:1–205:37, 2016. URL https://jmlr.org/papers/v17/16-272.html.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In William W. Cohen, Andrew McCallum, and Sam T. Roweis (eds.), *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pp. 1096–1103. ACM, 2008. doi: 10.1145/1390156.1390294. URL https://doi.org/10.1145/1390156.1390294.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. Magicoder: Source code is all you need. *CoRR*, abs/2312.02120, 2023. doi: 10.48550/ARXIV.2312.02120. URL https://doi.org/10.48550/arXiv.2312.02120.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning. *CoRR*, abs/2405.00451, 2024. doi: 10.48550/ARXIV.2405.00451. URL https://doi.org/10.48550/arXiv.2405.00451.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 5754–5764, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *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. URL http://papers.nips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. Physics of language models: Part 2.2, how to learn from mistakes on grade-school math problems, 2024. URL https://arxiv.org/abs/2408.16293.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. 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=N8N0hgNDRt.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. Reclor: A reading comprehension dataset requiring logical reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HJgJtT4tvB.

Qi Zhang, Tianqi Du, Haotian Huang, Yifei Wang, and Yisen Wang. Look ahead or look around? A theoretical comparison between autoregressive and masked pretraining. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=2rPoTgEmjV.

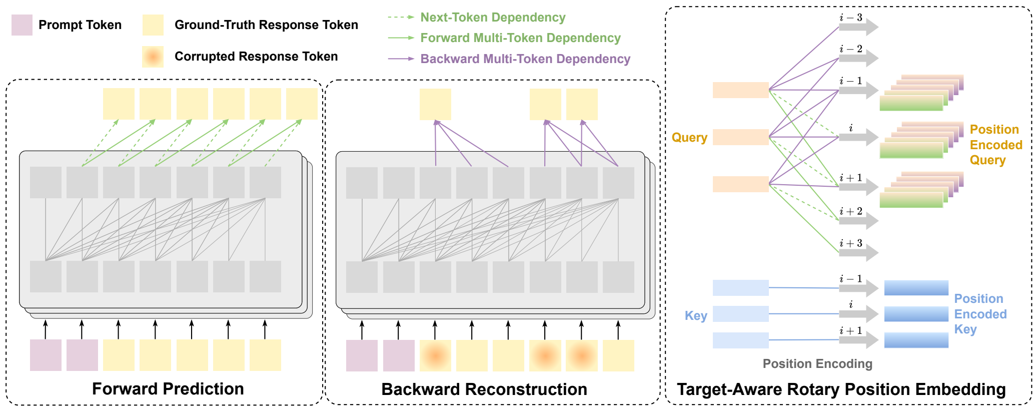



Figure 4: **Context-Wise Order-Agnostic Language Modeling**. We visualize the order-agnostic dependencies within a context window size $k = 2$. For target-aware position encoding, we show how COrAL obtains query representations for multiple positions within a context window size $k = 2$.

## Limitations

Order-agnostic language modeling can struggle with tasks that demand specific output formats or syntax due to inconsistencies in the multi-token predictions. On the one hand, this indicates the importance of task-specific design of the acceptance scheme in order-agnostic decoding. Future work can further explore the potential of incorporating different evaluation heuristics to guide the inference process.

Due to the computation constraint, we explore the model capabilities of order-agnostic modeling with fixed context window sizes at the SFT stage only. For future work, we may explore the effect of scaling context window sizes in both forward and backward directions. Moreover, the increase in the context window sizes can also enlarge the discrepancy between autoregressive pre-training and order-agnostic fine-tuning. We thus anticipate future work to extend COrAL at the pre-training stage to boost model capabilities. We extensively discuss the effect of the two-stage training strategy adopted in our setting in Appendix D.

Incorporating corrupted data may also introduce discrepancies between training- and inference-time objectives. For example, our experiment only explores rule-based context-wise corruption strategies to create noisy data. Future work may scale the difficulty level and proportion of corruption to understand its impact on model capabilities better.

## Potential Broader Impact

Compared to conventional autoregressive modeling, COrAL leverages multi-token prediction and reconstruction to backtrack past generations and refine them iteratively. This strategy mimics humans' decision-making process in real-world task completion. We anticipate COrAL to motivate the community to design more efficient and effective frameworks to enhance interpretability and alignment with humans.

## A Conceptual Comparison among Model Architectures

We consider the properties an ideal architecture should have as follows:

- **VL**: varying-length generation
- **BT**: backtrack / look-ahead
- **MV**: multi-variable generation
- **MD**: multi-dependency (inter-sample connection) modeling
- **FS**: fitting feasibility
- **EF**: inference efficiency

- **IT**: mechanism of iterative refinement

Table 4: Conceptual comparison regarding desired features across different architectures.

| Architectures | VL | BT | MV | MD | FS | EF | IT |
|---|---|---|---|---|---|---|---|
| Next-Token AR (Uria et al., 2016) | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |
| Permutation-Based AR (Uria et al., 2014) | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ |
| NAR (Gu et al., 2018) | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Diffusion (Ho et al., 2020) | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ |
| Consistency Model (Song et al., 2023) | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| COrAL (Ours) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

# B Sliding Blockwise Order-Agnostic Decoding

## B.1 Two-Stage Prediction–Verification Inference

**Prediction.** Given a set of possible distributions $\{p_\theta(y_t \mid \boldsymbol{y}_{\leq i}, \boldsymbol{x})\}_{i=t-k}^{t+k}$ for the $t$-th token in the output sequence, we obtain the ensemble distribution via model arithmetic (Dekoninck et al., 2024). Specifically, we apply different weights to the distributions to prioritize the more accurate dependencies, with distributions based on more qualified content generally leading to better generations:

$$\pi_\theta(y_t) = \mathrm{softmax}\left(\frac{1}{\sum_{i=t-k}^{t+k} \omega_{t-i}(\boldsymbol{y}_{\leq i}, \boldsymbol{x})} \sum_{i=t-k}^{t+k} \omega_{t-i}(\boldsymbol{y}_{\leq i}, \boldsymbol{x}) \log p_\theta(y_t \mid \boldsymbol{y}_{\leq i}, \boldsymbol{x})\right) \tag{3}$$

The weight function $\omega_{t-i}(\boldsymbol{y}_{\leq i}, \boldsymbol{x}) = \lambda_{t-i} \cdot c(\boldsymbol{y}_{\leq i} \mid \boldsymbol{x})$ is determined by the relative distance and direction of the dependency, as well as the confidence of the generated context $\boldsymbol{y}_{\leq i}$. Here, the factor $\lambda_{t-i} \in [0, 1]$ only depends on the relative position of the target token, decaying for longer dependencies. Using order-agnostic modeling, we calculate the confidence score $c$ by gathering the predicted probabilities based on different dependencies, which we obtain in the verification stage. Generally, backward reconstruction and next-token prediction based on iteratively refined content will be associated with higher weights. See Appendix D for a detailed comparison among different dependencies. In practice, some of the distributions in Eq. 3 may not be available for all tokens at each step. We calculate the ensemble utilizing available dependencies within the context window.

**Verification.** Following Cai et al. (2024), we employ tree attention[4] to select from multiple candidates sampled from the ensemble distribution $\pi_\theta$. Each candidate is a combination of tokens used to fill the sliding block. Unlike previous works that only adopt the original next-token probability for verification, we also incorporate the backward reconstruction probabilities to leverage the refinement ability of COrAL. The verification score can thereby be formulated as follows:

$$v_\theta(y_t) = \frac{1}{\sum_{i=t-1}^{t+k} \lambda_{t-i}} \sum_{i=t-1}^{t+k} \lambda_{t-i} \log p_\theta(y_t \mid \boldsymbol{y}_{\leq i}, \boldsymbol{x}) \tag{4}$$

Here, we only consider the next-token and backward predictions for the verification score calculation. This scheme can be further enhanced by introducing a contrastive objective (Li et al., 2023) that penalizes the possible failure cases in forward multi-token prediction:

$$v_\theta^{\mathrm{CD}}(y_t) = \max\left(0, \log p_\theta(y_t \mid \boldsymbol{y}_{\leq t-1}, \boldsymbol{x}) - \frac{1}{\sum_{i=t-k}^{t-2} \lambda'_{t-i}} \lambda'_{t-i} \log p_\theta(y_t \mid \boldsymbol{y}_{\leq i}, \boldsymbol{x})\right) \tag{5}$$

where $\lambda'_{t-i} = 1/\lambda_{t-i}$ to apply more penalization to predictions based on longer dependencies. Combining $v_\theta$ with $v_\theta^{\mathrm{CD}}$, we keep the candidate of the highest average score. We allow several

[4]To balance exploitation and exploration in tree construction, we select nodes according to the estimated accuracy of each token. Detailed considerations of candidate selection can be found in Appendix B.2.

**Algorithm 1** Sliding Blockwise Order-Agnostic Decoding

**Input:** Order-agnostic generator $\pi_\theta$ and verifiers $v_\theta$ and $v_\theta^{\mathrm{CD}}$ based on OA-LLM $p_\theta$, prompt $\boldsymbol{x}$, decoding context window size $k$, decoding block size $b$, maximum output sequence length $T$.
$\triangleright$ Initialize the current length of the output sequence
Initialize $t \leftarrow 0$, $\boldsymbol{y} \leftarrow \emptyset$.
$\triangleright$ Initialize the start and end position indices of the block to predict and refine
Initialize $t_s \leftarrow 1$, $t_e \leftarrow \min(k, b)$.
**while** $t_s < T$ **do**
  $\triangleright$ Collect candidates through tree construction
  Construct $\mathcal{Y}_{t_s:t_e} \leftarrow \left\{ \{\tilde{y}_i\}_{i=t_s}^{t_e}, \tilde{y}_i \sim \pi_\theta(y_i \mid \boldsymbol{y}, \boldsymbol{x}) \right\}$.
  $\triangleright$ Verify and select
  Select $\boldsymbol{y}_{t_s:t_e} \leftarrow \arg\max_{\tilde{\boldsymbol{y}}_{t_s:t_e} \sim \mathcal{Y}_{t_s:t_e}} \frac{1}{t_e - t_s + 1} \sum_{i=t_s}^{t_e} \left( v_\theta(\tilde{y}_i \mid \boldsymbol{y}, \boldsymbol{x}) + v_\theta^{\mathrm{CD}}(\tilde{y}_i \mid \boldsymbol{y}, \boldsymbol{x}) \right)$.
  Update $\boldsymbol{y} \leftarrow \mathrm{concat}(\boldsymbol{y}_{<t_s}, \boldsymbol{y}_{t_s:t_e})$.
  Set $t \leftarrow t_e$.
  $\triangleright$ Slide the decoding block based on rejection sampling
  **for** $i = t_s$ **to** $t_e$ **do**
    Sample $r \sim U[0, 1]$ from a uniform distribution
    **if** $r < c(y_i \mid \boldsymbol{y}, \boldsymbol{x})$ **then**
      Set $t_s \leftarrow t_s + 1$.
      **if** $y_i ==$ `[EOS]` **then**
        Exit while loop.
      **end if**
    **else**
      Exit for loop.
    **end if**
  **end for**
  Set $t_e \leftarrow \min(t_s + b - 1, t + k)$.
**end while**
**Output:** $\boldsymbol{y}$

refinement iterations for each position within a sliding block to enhance the generation quality. Specifically, we propose an ensemble rejection sampling scheme to determine the sliding step size through majority voting across multiple dependencies, where we accept each token with the probability:

$$c(y_t \mid \boldsymbol{y}_{\leq t+k}, \boldsymbol{x}) = \frac{1}{k+2} \sum_{i=t-1}^{t+k} \mathbb{1}_{p_\theta(y_t \mid \boldsymbol{y}_{\leq i}, \boldsymbol{x}) > \min\left(\epsilon, \epsilon \exp\left(-H\left(p_\theta(\cdot \mid \boldsymbol{y}_{\leq i}, \boldsymbol{x})\right)\right)\right)} \tag{6}$$

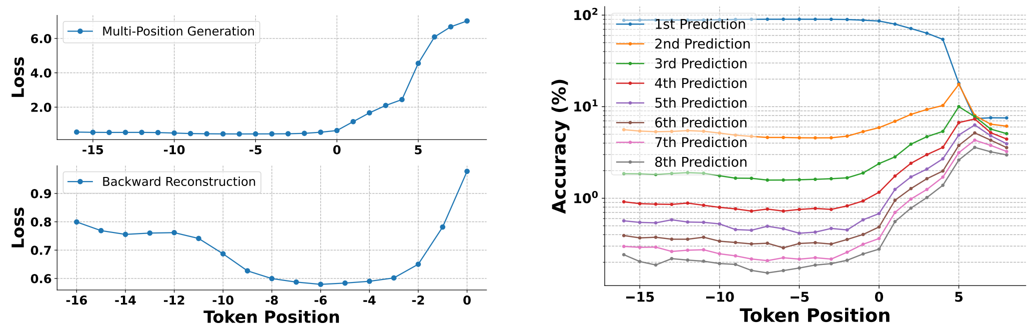
where $H(\cdot)$ is the entropy and $\epsilon$ is a fixed threshold to reject low-probability predictions. This acceptance scheme is inspired by truncation sampling (Hewitt et al., 2022; Cai et al., 2024) to choose candidates that are more likely to be sampled from the reference distributions. The sliding step size for each step is set to the length of the longest accepted prefix of the current block. We detail the sliding decoding procedure in Algorithm 1.

## B.2 Candidate Tree Construction in Order-Agnostic Decoding

Our specific design of tree construction aims to explore promising combinations of multi-position predictions with a fixed budget for the number of total nodes in the tree. Unlike selecting promising nodes based on the accuracies of the top predictions of different heads in Cai et al. (2024), we forego the need of a validation set for accuracy calculation by leveraging the model confidence of each prediction with a dedicated scaling factor. Let $p_t^{(i)}$ denote the model-predicted probability of the $i$-th top candidate for the $t$-th token. For a candidate sequence composed by the top $[i_{t_s}, i_{t_s+1}, \cdots, i_{t_e}]$ predictions of tokens at different positions, we estimate its accuracy as:

$$\prod_{j=t_s}^{t_e} \left( p_j^{(i_j)} / \gamma_j \right) \tag{7}$$

where $\gamma_i$ is a scaling factor to up weight the predictions based on nonconsecutive forward dependencies. As shown in Figure 5, this process benefits from the fact that COrAL obtains higher accuracies




(a) Token-wise loss.

(b) Token-wise accuracy of top-8 predictions.

Figure 5: Token-wise losses and accuracies corresponding to different dependencies.

on non-first predictions on such dependencies. Empirically, we set these factors to be 1.1, 1.2, 1.3 for the second, the third, and the fourth tokens to predict, respectively.

Following Eq. 7, we construct the tree in a greedy manner, adding the node with the highest confidence to the tree one by one. This process considers the token-wise confidence as the expected contribution of each prediction to the tree. We repeat the node-adding process until the total number of nodes reaches the desired number to accommodate the maximum sequence length the model can deal with.

# C Experimental Details

**Datasets.** For arithmetic reasoning, we train COrAL on MetaMathQA (395K) (Yu et al., 2024) and evaluate it using GSM8K (Cobbe et al., 2021) on grade school math word problems and MATH (Hendrycks et al., 2021) of challenging competition mathematics problems. For logical reasoning, we filter LogiCoT (Liu et al., 2023b) with deduplication and reformulation, obtaining 313K training samples. We assess logical reasoning performance with multiple-choice reading comprehension tasks that test interpretation and decision-making skills: LogiQA (Liu et al., 2023a), based on the Chinese Civil Service Examination, and ReClor (Yu et al., 2020), sourced from Law School Admission Council exams. For code generation, we train on Magicoder-Eval-Instruct-110K (Wei et al., 2023) and evaluate using programming tasks from HumanEval (Chen et al., 2021).

**Experimental Protocol.** To address the discrepancy between the pre-trained model based on next-token dependency and the target order-agnostic model, we adopt a two-stage training strategy (Kumar et al., 2022) to progressively enhance order-agnostic modeling. We begin with a domain-specific supervised fine-tuned (SFT) model for each target task. In the first stage, we perform order-agnostic training exclusively on the last target-aware layer, while freezing the other layers to preserve the output quality. In the second stage, following Cai et al. (2024), we train the entire model by focusing on the previously frozen layers first and then unlocking the last layer to train together. We use Mistral-7B-v0.3 and DeepSeek-Coder-6.7B-base as the base models for reasoning and code generation tasks, respectively. During inference, we explore the effect of the verification stage and ablate the values of decoding context window size and block size.

# D Further Analysis

We analyze COrAL's capability to model different dependencies, and the potential computation overhead from order-agnostic modeling. We also extensively discuss the training protocol we design to endow AR-LLMs with order-agnostic ability without pretraining. Lastly, we illustrate how COrAL efficiently corrects mistakes in previous generations in qualitative analysis.

**How does COrAL models order-agnostic dependencies?** We compare the model capabilities across different positions using token-wise losses and accuracies in Figure 5. Generally, COrAL performs better on backward reconstruction than forward prediction, as shown in the lower losses and higher accuracies on backward dependencies. Notably, we see better generalizability of backward reconstruction. For example, given the backward context window size $k = 8$ and forward context window size $k = 4$, we find that the loss and accuracy of backward reconstruction with dependencies longer than the training context window size, such as positions $|-9| > |-8|$, are also at the same level as other backward dependencies. Differently, we observe a dramatic increase in loss and a drop in accuracy from positions $4$ to $5$ on longer dependencies in forward prediction. This explains how backward refinement benefits from more information in sequence-level generation to improve performance. We observe decreased performance for forward prediction as the dependency gets longer, especially when it exceeds the forward context window size in training. However, we can mitigate this issue by aggregating multiple predictions for each position. As shown in Figure 5b, while forward positions with longer dependencies obtain lower accuracies on tghe first prediction, the accumulated accuracies of their non-first predictions are generally higher than those from other dependencies. This illustrates how COrAL can benefit from the tree construction and verification stage in decoding (Section 2.3) by considering multiple candidates for each position.

**Computation Overhead.** One concern regarding order-agnostic modeling is the potential computation overhead to accommodate more dependencies in the context windows. As target-aware RoPE is only applied on the last layer, this overhead scales relatively slower as we increase the number of positions to predict. For example, with forward and backward context window sizes each set as $k = 4$, each forward pass of COrAL costs $5.48$ TFLOPS, compared with $2.81$ TFLOPS of next-token prediction. In other words, COrAL predicts $8\times$ number of tokens with less than $2\times$ overhead in computational cost. This indicates the efficiency of COrAL in leveraging available computation resources to accelerate and enhance inference. Furthermore, we can adjust the forward and backward context window sizes to determine the number of tokens to predict in parallel, demonstrating the flexibility and generalizability of COrAL with target-aware RoPE.

**Effect of Two-Stage Training.** Empirically, we find that a high corruption ratio can cause a collapse in model performance as the noisy data contains corrupted information in a format that the model has not seen in pretraining. Furthermore, we are also faced with the order-agnostic training tax to endow an AR-based LLM with denoising and multi-token prediction abilities. In this section, we elaborate on the two-stage training we designed to mitigate this issue. Following Cai et al. (2024), we first tune the last layer where we apply target-aware RoPE. This adapts the previous parameterization on next-token prediction to target-aware multi-position prediction. Due to the discrepancy of training objectives in pretraining and fine-tuning, full fine-tuning is still essential to ensure better performance on multi-token prediction. To stabilize the training process, we then freeze the last layer and gradually unlock it through the second training stage of full fine-tuning. Empirically, we find this strategy effective for stabilizing the autoregressive loss changes in forward prediction. However, we observe an order-agnostic training tax where the next-token prediction performance drops from $77.0\%$ to $76.5\%$ and then $74.1\%$ after the first and second stages, respectively. This performance degradation possibly comes from two aspects: the difference in training objectives and the incorporation of corrupted data in fine-tuning. We leave it to future work to further explore the effect of applying our order-agnostic framework to the pretraining stage.

**Qualitative Analysis.** Our qualitative analysis on GSM8K and LogiQA showcases how COrAL corrects previously generated mistakes through the iterative internal process. In Figure 6, COrAL obtained a wrong calculated result 72 at the 48-th step. However, the backward refinement mechanism enables it to backtrack and refine the result to the correct number, $74$, as shown at the 49-th step. In contrast, the next-token baseline cannot correct the erroneous 72, leading to the wrong final result. On the other hand, we observe the incoherence in COrAL's generation where COrAL can fail in correcting the mistakes when it happens to skip some positions during generation. For example, at the 1-st step, COrAL outputs "bakeraked" instead of "baker baked". This error incurs a chain reaction where the subsequent outputs all omit the correct token " b" right after "baker", indicating the need for further enhancement on the generation fluency of order-agnostic methods.

On LogiQA, interestingly, we observe a higher frequency of the inconsistencies in COrAL's generation. We attribute this scenario to the relatively low proportion of LogiQA-related training data in LogiCoT, where there are only 5K samples out of the 313K data points. As shown in Figure 7, while the COrAL produces several grammatical errors in a generation, it still achieves the correct result. This indicates the advanced ability of COrAL to sematically escape from paths that may lead to dead ends through iterative refinement.

**Prompt**

The Kennel house keeps 3 German Shepherds and 2 Bulldogs.
If a German Shepherd consumes 5 kilograms of dog food and a bulldog consumes 3 kilograms of dog food per day.
How many kilograms of dog food will they need in a week?

**Responses**

**Sliding Blockwise Order-Agnostic Decoding**

(1) The bakeraked

(2) The bakeraked 200

(3) The bakeraked 200 loaves of bread

......

(6) The bakeraked 200 loaves of bread.
They sold 93 loaves the in3

(7) The bakeraked 200 loaves of bread.
They sold 93 loaves in the morning morning and

(8) The bakeraked 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in

......

(38) The bakeraked 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total of 93 + 39 = 132 loaves.
They returned 6 loaves, so they had 132 - 6 = 1226 loaves

(39) The bakeraked 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total of 93 + 39 = 132 loaves.
They returned 6 loaves, so they had 132 - 6 = 126 loaves. left...

(40) The bakeraked 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total of 93 + 39 = 132 loaves.
They returned 6 loaves, so they had 132 - 6 = 126 loaves left.
Therefore, they266

......

(47) The bakeraked 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total of 93 + 39 = 132 loaves.
They returned 6 loaves, so they had 132 - 6 = 126 loaves left.
Therefore, they had 200 - 126 =

(48) The bakeraked 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total of 93 + 39 = 132 loaves.
They returned 6 loaves, so they had 132 - 6 = 126 loaves left.
Therefore, they had 200 - 126 = 72

(49) The bakeraked 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total of 93 + 39 = 132 loaves.
They returned 6 loaves, so they had 132 - 6 = 126 loaves left.
Therefore, they had 200 - 126 = 74 loaves of bread

......

(59) The bakeraked 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total of 93 + 39 = 132 loaves.
They returned 6 loaves, so they had 132 - 6 = 126 loaves left.
Therefore, they had 200 - 126 = 74 loaves of bread left.
#### 74
The answer is: 74 ✓

**Next-Token Based Greedy Decoding**

(1) The

(2) The b

(3) The bery

......

(6) The bery b had 2

(7) The bery b had 20

(8) The bery b had 200

......

(38) The bery b had 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold

(39) The bery b had 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a

(40) The bery b had 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total

......

(47) The bery b had 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total of 93 + 3

(48) The bery b had 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total of 93 + 39

(49) The bery b had 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total of 93 + 39 =

......

(59) The bery b had 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total of 93 + 39 = 132 loaves.
They also

......

(106) The bery b had 200 loaves of bread.
They sold 93 loaves in the morning and 39 loaves in the afternoon,
so they sold a total of 93 + 39 = 132 loaves.
They also returned 6 unsold loaves,
so they had 200 - 132 + 6 = 72 loaves of bread left.
#### 72
The answer is: 72 ✗

Figure 6: Qualitative result comparison on GSM8K.

**Prompt**

**Passage**:
Youth phase refers to how adolescents perceive their level of youth development to be earlier, more timely or later than their peers.

**Question**: According to the above definition, which of the followings is timely in the phase of youth activation is?

**A**. Junior high school student A is the shortest boy in the class, but his parents think it is normal
**B**. Junior high student B had several zits on his face, while other students did not, which made him feel uncomfortable
**C**. Junior high school students C in the physical health class and other students like the opposite sex of the physiological structure is full of curiosity
**D**. Junior high school students in the adolescent physical health development self-assessment scale carefully tick the normal option

**Responses**

**Sliding Blockwise Order-Agnostic Decoding**

1 Junior high high

2 Junior high school students C in the

3 Junior high school students C in the physical health class class

……

6 Junior high school students C in the physical health class and other students like the opposite of of the physi

7 Junior high school students C in the physical health class and other students like the opposite of of the physiological

8 Junior high school students C in the physical health class and other students like the opposite of of the physiological structure is curious

……

38 Junior high school students C in the physical health class and other students like the opposite of of physiological structure is full of curious curiosity. This curious is about the normal stage of physical development for adolesents, and it

39 Junior high school students C in the physical health class and other students like the opposite of of physiological structure is full of curious curiosity. This curious is about the normal stage of physical development for adolesents, and it

……

48 Junior high school students C in the physical health class and other students like the opposite of of physiological structure is full of curious curiosity. This curious is about the normal stage of physical development for adolesents, and it does not indicate that the student is earlier or

49 Junior high school students C in the physical health class and other students like the opposite of of physiological structure is full of curious curiosity. This curious is about the normal stage of physical development for adolesents, and it does not indicate that the student is earlier or later than their

……

74 Junior high school students C in the physical health class and other students like the opposite of of physiological structure is full of curious curiosity. This curious is about the normal stage of physical development for adolesents, and it does not indicate that the student is earlier than their peers or perceive their development, so thefore, the the correct answer is C. ✓

**Next-Token Based Greedy Decoding**

1 J

2 Jun

3 Junior

……

6 Junior high school students

7 Junior high school students in

8 Junior high school students in the

……

38 Junior high school students in the adolescent physical health development self-ass assessment scale carefully tick the normal option - This choice reflects the perception of the student's level of youth development

39 Junior high school students in the adolescent physical health development self-ass assessment scale carefully tick the normal option - This choice reflects the perception of the student's level of youth development compared

……

48 Junior high school students in the adolescent physical health development self-ass assessment scale carefully tick the normal option - This choice reflects the perception of the student's level of youth development compared to their peers, as they they they normal

49 Junior high school students in the adolescent physical health development self-ass assessment scale carefully tick the normal option - This choice reflects the perception of the student's level of youth development compared to their peers, as they they they normal option

……

73 Junior high school students in the adolescent physical health development self-ass assessment scale carefully tick the normal option - This choice reflects the perception of the student's level of youth development compared to their peers, as they they they normal option indicating that they feel their development is timely to their peers.
Therefore, the correct answer is D. ✗

Figure 7: Qualitative result comparison on LogiQA.