
Supplementary Material of SPMDM

Anonymous Author(s)

Affiliation

Address

email

1 A Related Works

2 **Autoregressive Models.** The autoregressive modeling paradigm, where the prediction of a token
3 only depends on the preceding context, is widely adopted in modeling language. Autoregressive
4 models have catalyzed significant advances in artificial intelligence, achieving state-of-the-art results
5 across a range of tasks, including fluent text synthesis [24, 1, 32, 5], program generation [25], and
6 chain-of-thought reasoning in mathematical domains [33]. However, despite their transformative im-
7 pact and broad deployment in real-world systems, ARMs are inherently constrained by their sequential,
8 left-to-right generation strategy [18, 30, 6, 17]. This unidirectional nature presents persistent chal-
9 lenges in scenarios requiring foresight, multi-step reasoning, and iterative self-correction [19, 4, 34].

10 **Diffusion Language Models.** Continuous diffusion models have demonstrated remarkable perfor-
11 mance and controllability in image generation tasks [16, 31, 10, 15]. Building on these successful
12 practices, several works have extended continuous diffusion models to text generation [20, 12, 11, 14].
13 Among them, Plaid [14] is a notable approach that maps discrete text into a continuous embedding
14 space and constructs a continuous diffusion framework in that space. Given the inherently discrete
15 nature of language, Austin et al. [3] proposed D3PM, a diffusion framework tailored to discrete data
16 domains. They incorporate an absorbing [MASK] state as noise, laying the foundation for discrete
17 diffusion models. This framework has been further developed by [21], [22], [26], and [29]. Among
18 these, the MDLM [26] framework is one of the most widely adopted, offering a simple and efficient
19 training objective. More recently, BDLM [2] combines ARMs with MDMs through interpolation,
20 integrating the left-to-right generation paradigm of ARMs into MDMs and proposing a novel diffusion
21 modeling framework. Furthermore, Ye et al. [35] have shown that MDMs significantly outperform
22 ARMs in complex reasoning and global planning tasks.

23 B Simple Path Mask Diffusion Model

24 Recall that, under the SPMDM framework, the input token sequence \mathbf{x} (of total length N) is divided
25 into K non-overlapping subsequences, each of length L . Thus, $K = N/L$, assuming N is an integer
26 multiple of L . The k -th subsequence, denoted as \mathbf{x}^k for $k \in \{1, \dots, K\}$, comprises tokens from the
27 original sequence. For convenience, we refer to the ℓ -th token within the k -th subsequence as $\mathbf{x}^{k,\ell}$.

28 B.1 Forward Process

29 The forward noise process applied independently for each token is defined as follows:

$$q_{\mathbf{t}|0}(\mathbf{x}_{\mathbf{t}} | \mathbf{x}_0) = \prod_k \prod_{\ell} q_{t_k|0}(\mathbf{x}_{t_k}^{k,\ell} | \mathbf{x}_0^{k,\ell}), \quad q_{t_k|0}(\mathbf{x}_{t_k}^{k,\ell} | \mathbf{x}_0^{k,\ell}) = \text{Cat}(\alpha_{t_k} \mathbf{x}_0^{k,\ell} + (1 - \alpha_{t_k}) \mathbf{m}), \quad (1)$$

30 where $\mathbf{t} = t_1, \dots, t_K$, and t_k denotes the noising step applied to \mathbf{x}^k .

31 B.2 Reverse Process

32 Under the framework of MDLM [26], the reverse process iteratively recover values for masked
 33 tokens, starting from a mask sequence $\mathbf{x}_1 = [\mathbf{m}, \dots, \mathbf{m}]$. Let $0 \leq s_k < t_k \leq 1$, the reverse process
 34 is given by:

$$\begin{aligned}
 q_{\mathbf{s}|\mathbf{t}}(\mathbf{x}_{\mathbf{t}} | \mathbf{x}_0) &= \prod_k^K \prod_{\ell}^L q_{s_k|t_k}(\mathbf{x}_{s_k}^{k,\ell} | \mathbf{x}_{t_k}^{k,\ell}, \mathbf{x}^{k,\ell}), \\
 q_{s_k|t_k}(\mathbf{x}_{s_k}^{k,\ell} | \mathbf{x}_{t_k}^{k,\ell}, \mathbf{x}^{k,\ell}) &= \begin{cases} \text{Cat}(\mathbf{x}_{t_k}^{k,\ell}) & \mathbf{x}_{t_k}^{k,\ell} \neq \mathbf{m}; \\ \text{Cat}\left(\frac{1-\alpha_{t_s}}{1-\alpha_{t_k}} \mathbf{m} + \frac{\alpha_{t_s}-\alpha_{t_k}}{1-\alpha_{t_k}} \mathbf{x}^{k,\ell}\right) & \mathbf{x}_{t_k}^{k,\ell} = \mathbf{m}. \end{cases} \quad (2)
 \end{aligned}$$

35 B.3 Simple Path Mask Diffusion NELBO

36 We provide the negative evidence lower bound (NELBO) for the simple path masked diffusion
 37 parameterization. We firstly perform diffusion in each block over T discretization steps. Let $\text{D}_{\text{KL}}[\cdot]$
 38 denote the Kullback-Leibler divergence, t_k, s_k be shorthand for $t_k(i) = i/T$ and $s_k(i) = t(i-1)/T$,
 39 $\forall i \in [1, T]$. We derive the NELBO as follows:

$$\begin{aligned}
 -\log p_{\theta}(\mathbf{x}) &= -\sum_{k=1}^K \log \mathbb{E}_q \left[\frac{p_{\theta}(\mathbf{x}_{t_k(1):t_k(T)}^k | \mathbf{x}_{\mathbf{t}}^{-k})}{q(\mathbf{x}_{t_k(1):t_k(T)}^k | \mathbf{x}^k)} \right] \\
 &= -\sum_{k=1}^K \log \mathbb{E}_q \left[\frac{p_{\theta}(\mathbf{x}_{t_k(T)}^k | \mathbf{x}_{\mathbf{t}}^{-k}) \prod_{i=1}^T p_{\theta}(\mathbf{x}_{s_k(i)}^k | \mathbf{x}_{t_k(i)}^k, \mathbf{x}_{\mathbf{t}}^{-k})}{\prod_{i=1}^T q(\mathbf{x}_{s_k(i)}^k | \mathbf{x}_{t_k(i)}^k)} \right] \\
 &\leq \sum_{k=1}^K \left[\underbrace{-\mathbb{E}_q \log p_{\theta}(\mathbf{x}^k | \mathbf{x}_{t_k=\frac{1}{T}}^k, \mathbf{x}_{\mathbf{t}}^{-k})}_{\mathcal{L}_{\text{recons}}} \right. \\
 &\quad + \underbrace{\mathbb{E}_{t_k \in \{\frac{2}{T}, \dots, \frac{T-1}{T}, 1\}} \mathbb{E}_q T \text{D}_{\text{KL}}(q(\mathbf{x}_{s_k}^k | \mathbf{x}_{t_k}^k, \mathbf{x}^k) \| p_{\theta}(\mathbf{x}_{s_k}^k | \mathbf{x}_{t_k}^k, \mathbf{x}_{\mathbf{t}}^{-k}))}_{\mathcal{L}_{\text{diffusion}}} \\
 &\quad \left. + \underbrace{\text{D}_{\text{KL}}(q(\mathbf{x}_{t_k=1}^k | \mathbf{x}^k) \| p_{\theta}(\mathbf{x}_{t_k=1}^k))}_{\mathcal{L}_{\text{prior}}} \right] \quad (3)
 \end{aligned}$$

40 B.4 Simplified NELBO

41 We adopt the SUBS parameterization proposed by Sahoo et al. [26]. Specifically, we impose the
 42 following constraints on the design of p_{θ} by exploiting the fact that, at any timestep t , each token \mathbf{x}_t^{ℓ}
 43 can only reside in one of two states: the original token \mathbf{x}^{ℓ} or the mask token \mathbf{m} , i.e., $\mathbf{x}_t^{\ell} \in \mathbf{x}^{\ell}, \mathbf{m}$ for
 44 all $\ell \in 1, \dots, L$:

45 1. **Zero Masking Probability.** Since the clean target sequence \mathbf{x} does not contain any mask tokens,
 46 we enforce $p_{\theta}(\mathbf{x}^{\ell} = \mathbf{m} | \mathbf{x}_t^{\ell}) = 0$, ensuring that the model never predicts a mask token during
 47 denoising.

48 2. **Carry-Over Unmasking.** Once a token is unmasked in the reverse process, it is never remasked.
 49 Accordingly, we simplify the denoising model by enforcing $p_{\theta}(\mathbf{x}_s^{\ell} = \mathbf{x}_t^{\ell} | \mathbf{x}_t^{\ell} \neq \mathbf{m}) = 1$, meaning
 50 that any token already unmasked remains unchanged in subsequent steps.

Table 1: Dataset Deatails. Intra and Inter refer to toy datasets designed for intra- and inter-subsequence modeling, respectively. CD is an abbreviation for Countdown..

	Intra	Inter	CD3	CD4	CD5	Sudoku
Train Entries	50k	50k	500k	500k	500k	100k
Test Entries	1k	1k	1k	1k	1k	1k
Avg Input Token	8	8	11	13	16	81
Avg Output Token	-	-	12	25	35	81
Max Input Token	8	8	16	15	18	81
Max Output Token	-	-	22	35	52	81

As a result, we will only approximate the posterior $p_\theta(\mathbf{x}_s^\ell = \mathbf{x}^\ell \mid \mathbf{x}_t^\ell = \mathbf{m})$. The diffusion loss term becomes:

$$\begin{aligned}
\mathcal{L}_{\text{diffusion}} &= \sum_{k=1}^K \mathbb{E}_{t_k} \mathbb{E}_q T \left[\text{D}_{\text{KL}} \left(q(\mathbf{x}_{s_k}^k \mid \mathbf{x}_{t_k}^k, \mathbf{x}^k) \parallel p_\theta(\mathbf{x}_{s_k}^k \mid \mathbf{x}_{t_k}^k, \mathbf{x}_t^{-k}) \right) \right] \\
&= \sum_{k=1}^K \mathbb{E}_{t_k} \mathbb{E}_q T \left[\sum_{\ell=1}^L \text{D}_{\text{KL}} \left(q(\mathbf{x}_{s_k}^{k,\ell} \mid \mathbf{x}_{t_k}^{k,\ell}, \mathbf{x}^{k,\ell}) \parallel p_\theta(\mathbf{x}_{t_k}^{k,\ell} \mid \mathbf{x}_{t_k}^{k,\ell}, \mathbf{x}_t^{-b}) \right) \right] \\
&= \sum_{k=1}^K \mathbb{E}_{t_k} \mathbb{E}_q T \left[\sum_{\ell=1}^L \frac{\alpha_{t_k} - \alpha_{s_k}}{1 - \alpha_{t_k}} \log p_\theta(\mathbf{x}^{k,\ell} \mid \mathbf{x}_{t_k}^{k,\ell}, \mathbf{x}_t^{-b}) \right] \\
&= \sum_{k=1}^K \mathbb{E}_{t_k} \mathbb{E}_q T \left[\frac{\alpha_t - \alpha_s}{1 - \alpha_t} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}_{t_k}^b, \mathbf{x}_t^{-b}) \right]
\end{aligned} \tag{4}$$

Previous works have shown empirically and mathematically that increasing the number of steps T yields a tighter approximation to the ELBO [8]. Following a similar argument, we form an continuous time extension by taking $T \rightarrow \infty$, which yields the following diffusion loss term:

$$\mathcal{L}_{\text{diffusion}} = \sum_{k=1}^K \mathbb{E}_{t_k \sim [0,1]} \mathbb{E}_q \left[\frac{\alpha'_{t_k}}{1 - \alpha_{t_k}} \log p_\theta(\mathbf{x}^k \mid \mathbf{x}_{t_k}^k, \mathbf{x}_t^{-k}) \right] \tag{5}$$

For the continuous time case, we have $\mathbf{x}_{t_k=\frac{1}{T}}^k \sim \lim_{T \rightarrow \infty} \text{Cat} \left(\mathbf{x}_{t_k=\frac{1}{T}}^k \right) = \text{Cat}(\mathbf{x}^k)$. Then, the reconstruction loss term becomes:

$$\mathcal{L}_{\text{recons}} = -\mathbb{E}_q \log p_\theta(\mathbf{x}^k \mid \mathbf{x}_{t_k=\frac{1}{T}}^k, \mathbf{x}_t^{-k}) = -\log p_\theta(\mathbf{x}^k \mid \mathbf{x}_{t_k=\frac{1}{T}}^k = \mathbf{x}^k, \mathbf{x}_t^{-k}) = 0 \tag{6}$$

The prior loss also reduces to 0 because $\alpha_{t=1} = 0$, which ensures $q(\mathbf{x}_{t_k=1}^k \mid \mathbf{x}^k) = \text{Cat}(\mathbf{m})$ and $p_\theta(\mathbf{x}_{t_k=1}^k) = \text{Cat}(\mathbf{m})$.

Finally, we obtain a simple objective as follows:

$$\mathcal{L}_{\text{NELBO}} = \sum_{k=1}^K \mathbb{E}_{t_k \sim [0,1]} \mathbb{E}_q \left[\frac{\alpha'_t}{1 - \alpha_t} \log p_\theta(\mathbf{x}^k \mid \mathbf{x}_{t_k}^k, \mathbf{x}_t^{-k}) \right] \tag{7}$$

C Experimental Details

C.1 Dataset Details

We present the detailed specifications of the toy datasets and problem-solving task datasets in Table 1.

C.2 Implementation Details

Toy Examples. We conduct all toy example experiments using four RTX 4090 GPUs. MDLM [26], BDLM [2], and SPMDM are all implemented using a tiny model with 6M parameters. For BDLM, the block size is set to 2, and for SPMDM, the subsequence length is also set to 2. We use a learning rate of 1×10^{-3} and a batch size of 1024. All models are trained for 10 epochs on the training set. Additionally, the number of sampling steps is fixed to 32 for all models.

Problem-solving Tasks. We conduct all experiments related to problem-solving tasks using eight RTX 4090 GPUs. Both ARMs and MDMs are implemented based on the GPT-2 architecture. Across all datasets, we use a learning rate of 1×10^{-3} for the 6M-parameter tiny models and 3×10^{-4} for the 85M-parameter models. The batch size is set to 512. For the countdown task, we train for 150 epochs, and for the sudoku task, we train for 100 epochs. Specifically, for the Countdown task, we set the block size of BDLM to 4, and for the Sudoku task, we set it to 9. During sampling, we fix the number of denoising steps to 32 for all MDMs across all tasks. For LLaMA, we follow the results reported in the work of Ye et al. [35]; detailed fine-tuning settings can be found in Appendix C of [35].

Reasoning Tasks. For GPT-2, SEDD, and DiffuGPT, we borrow the results reported by Gong et al. [13]. Following their experimental setup, we also use the advanced FineWeb2 corpus [23], which is derived from Common Crawl, as the training dataset for both MDLM and SPMDM. Training and sampling are conducted on eight A100 GPUs with 40GB of memory. For models with 127M and 355M parameters, we use a learning rate of 3×10^{-4} with a cosine scheduler. The batch size is set to 512, and training is performed for a total of 400K iterations. During inference, the number of sampling steps is fixed to 256.

C.3 Evaluation Details

For the intra- and inter-subsequence modeling tasks, we perform unconditional generation using MDMs, generating 1,000 samples per task and evaluating performance by counting the number of samples that satisfy the predefined structural constraints. For the countdown and sudoku tasks, we conduct conditional generation using the questions from the test set as prompts, and measure performance by the number of correctly solved instances. For common sense reasoning tasks—HellaSwag [36], Winogrande [27], SIQA [28], and PIQA [7]—we use answer accuracy as the evaluation metric. For GSM8K [9], we take the questions from the test set as prompts for conditional generation and report the accuracy of the final predicted answers as the evaluation metric.

D Limitation

SPMDM framework is based on factorization assumptions. In a state space of length N , the transition matrix [8] contains an exponential number of possible states, making it computationally expensive to reverse. To alleviate this issue, existing works [8, 21, 26, 29] assume independence between dimensions, treating each dimension as an independent one-dimensional diffusion process with the same transition rate matrix. Admittedly, in language modeling, tokens are not entirely independent, there exist complex dependencies between them. However, without this independence assumption, the computational cost of training would become astronomical, and the modeling complexity would increase significantly. Despite this simplification, extensive prior work [8, 21, 26, 29] and our own experiments demonstrate that under this assumption, the model achieves strong performance with practically acceptable results for real-world applications. At the same time, this presents an interesting research direction—exploring ways to explicitly model the conditional dependencies between tokens. By leveraging these dependencies to strategically plan the denoising process of DDMs, we can potentially unlock significant improvements in the model’s generative capabilities.

E Impact

Ethical Impacts. This study does not pose any ethical concerns. It does not involve subjective assessments or the use of private data, as all experiments are conducted on publicly available datasets.

Expected Societal Implications. The primary potential societal impact of SPMDM lies in its possible misuse, particularly in generating false or misleading information, which could contribute to misinformation, privacy violations, and other harmful consequences. To mitigate these risks, it is essential to establish robust ethical guidelines and implement continuous monitoring to ensure the responsible and ethical deployment of such generative models.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.
- [3] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.
- [4] Gregor Bachmann and Vaishnavh Nagarajan. The pitfalls of next-token prediction. *arXiv preprint arXiv:2403.06963*, 2024.
- [5] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.
- [6] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.
- [7] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.
- [8] Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.
- [9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukas Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [11] Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.
- [12] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.
- [13] Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. *arXiv preprint arXiv:2410.17891*, 2024.
- [14] Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36:16693–16715, 2023.
- [15] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

- [17] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.
- [18] Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021.
- [19] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.
- [20] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022.
- [21] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. 2023.
- [22] Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.
- [23] Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Martin Jaggi, Leandro von Werra, and Thomas Wolf. Fineweb2: A sparkling update with 1000s of languages, December 2024. URL <https://huggingface.co/datasets/HuggingFaceFW/fineweb-2>.
- [24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [25] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.
- [26] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024.
- [27] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.
- [28] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialliqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.
- [29] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K Titsias. Simplified and generalized masked diffusion for discrete data. *arXiv preprint arXiv:2406.04329*, 2024.
- [30] Andy Shih, Dorsa Sadigh, and Stefano Ermon. Training and inference on any-order autoregressive models the right way. *Advances in Neural Information Processing Systems*, 35:2762–2775, 2022.
- [31] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.
- [32] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- [33] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [34] Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622*, 2024.

- 213 [35] Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng
214 Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *arXiv*
215 *preprint arXiv:2410.14157*, 2024.
- 216 [36] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a
217 machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.