LAD: Language Augmented Diffusion for Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Learning skills from language provides a powerful avenue for generalization in reinforcement learning, although it remains a challenging task as it requires agents to capture the complex interdependencies between language, actions, and states. In this paper, we propose leveraging **L**anguage **A**ugmented **D**iffusion models as a planner conditioned on language (LAD). We demonstrate the comparable performance of LAD with the state-of-the-art on the CALVIN language robotics benchmark with a much simpler architecture that contains no inductive biases specialized to robotics, achieving an average success rate (SR) of 72% compared to the best performance of 76%. We also conduct an analysis on the properties of language conditioned diffusion in reinforcement learning.

1 Introduction

It has been a longstanding dream of the AI community to be able to create a household robot that can follow natural language instructions and execute behaviors such as cleaning dishes or organizing the living room [1, 2, 3, 4, 5, 6, 7, 8, 9]. Incorporating language into Reinforcement Learning (RL) has great potential for generalization, enabling agents to utilize common sense priors across tasks and environments. Language provides an expressive abstraction of the environment and systematic generalization to new actions[10, 11, 12, 13, 14]. Given the recent progress in Natural Language Processing, how can one incorporate the powerful capabilities of language models and utilize them for downstream decision-making?

Language-conditioned policies (LCPs) are one class of policies in RL [15, 16, 13] that can be used to formulate this task, through the conditioning of behavior on natural language instruction. In this paper, we consider a novel approach for constructing LCPs by viewing the image as a sequence of state-actions rather than pixels, we reformulate language conditioned decision-making into text-to-image generation.

Diffusion models such as DALL-E 2 [17] and GLIDE [18] have recently shown promise as a generative model, with state-of-the-art text-to-image generation results demonstrating a surprisingly deep understanding of semantic relationships and generation of novel scenes. A key driver of the recent success in generative modeling is the usage of classifier-free guidance, which is amenable to the RL framework through the usage of language as a reward function. Inspired by the recent success of diffusion models in generative modeling [17, 18], we propose a new algorithm for LCPs (LAD) via latent diffusion models [19]. We demonstrate comparable performance to the state of the art on the CALVIN benchmark [20], with average success rate (SR) of 72% compared to the best performance of 76%.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

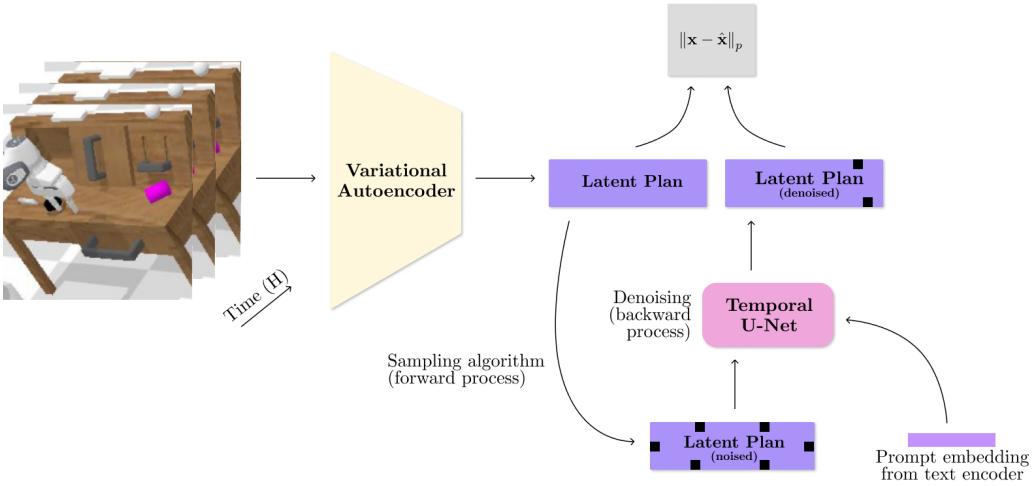

Figure 1: An overview of our training pipeline. The VAE is used to encode a sampled horizon of RGB observations into a lower dimensional latent space. We concatenate this sequence of states with the corresponding sequence of actions to construct the latent plan. We then noise the plan according to a uniformly sampled timestep from the diffusion process' variance schedule. We train a Temporal U-Net to reverse this process when conditioned on an encoded natural language instruction from an upstream language model, effectively learning how to conditionally denoise the latent plan. To train the U-Net, one can simply use the p-norm between the predicted latent plan and the ground truth latent plan as the loss. We set $p = 1$ in practice.

2 Language Augmented Diffusion

2.1 Method Overview

We consider a language-conditioned RL setting where we assume that the true reward function $\mathcal{R}$ is unknown, and must be inferred from a natural language instruction $L \in \mathscr{L}$. Formally, let $\mathcal{F}$ be the function space of $\mathcal{R}$. Then the goal becomes learning a $\phi : \mathscr{L} \mapsto \mathcal{F}$, and maximizing the policy objective conditioned on the reward function $\phi(L)$, $J(\pi(\cdot \mid s, \mathcal{R})) = \mathbb{E}_{a\sim\pi, s\sim p} \sum_{t=0}^{\infty} \gamma^t r_t$. Note that the space of tasks that can be specified by language is much larger than that of reward, due to the Markov restriction of the latter. For example, "pour the milk" and "pour the milk after five o'clock" are both valid instructions, but are indistinguishable from a reward function if the state does not contain temporal information. We assume access to a prior collected dataset $\mathcal{D}$ of N annotated trajectories $\tau_i = \langle (s_0, a_0, ...s_T), L_i \rangle$. The language conditioned policy π_β, or the behavior policy. is defined to be the policy that generates the aforementioned dataset. In this paper, we assume access to a dataset of expert trajectories, such that $\pi_\beta =$ optimal policy $\pi^\star$. In this case, the policy objective reduces to imitation learning, or

$$\min_{\pi} \mathbb{E}_{s,\mathcal{R}\sim\mathcal{D}} \left[D_{\mathrm{KL}} \left(\pi_\beta(\cdot \mid s, \mathcal{R}), \pi(\cdot \mid s, \mathcal{R}) \right) \right] \tag{1}$$

As we tackle the problem from a planning perspective, we define a trajectory generator as $\mathcal{P}$ and switch the atomic object from actions to trajectories $\boldsymbol{\tau}$. Thus we aim to

$$\begin{aligned} &\min_{\mathcal{P}} D_{\mathrm{KL}} \left(\mathcal{P}_\beta(\boldsymbol{\tau} \mid \mathcal{R}), \mathcal{P}(\boldsymbol{\tau} \mid \mathcal{R}) \right) \\ =& \min_{\mathcal{P}} \mathbb{E}_{\boldsymbol{\tau},\mathcal{R}\sim\mathcal{D}} \left[\log \mathcal{P}_\beta(\boldsymbol{\tau} \mid \mathcal{R}) - \log \mathcal{P}(\boldsymbol{\tau} \mid \mathcal{R}) \right] \end{aligned} \tag{2}$$

To model this, we turn to diffusion models [21]. Inspired by non-equilibrium thermodynamics, the common forms of diffusion models [22, 23, 24] propose modeling the data distribution $p(\boldsymbol{\tau})$ as a random process that steadily adds increasingly varied amounts of Gaussian noise to samples from $p(\boldsymbol{\tau})$ until the distribution converges to the standard normal. We denote the forward process as

$f(\boldsymbol{\tau}_t|\boldsymbol{\tau}_{t-1})$, with a sequence of variances $(\beta_0, \beta_1 ... \beta_T)$. We define $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$.

$$f(\boldsymbol{\tau}_{1:T}|\boldsymbol{\tau}_0) = \prod_{t=1}^{T} f(\boldsymbol{\tau}_t|\boldsymbol{\tau}_{t-1}), \quad \text{where } f(\boldsymbol{\tau}_t|\boldsymbol{\tau}_{t-1}) = \mathcal{N}(\boldsymbol{\tau}_t; \sqrt{1-\beta_t}\boldsymbol{\tau}_{t-1}, \beta_t \mathbf{I}). \tag{3}$$

One can tractably reverse this process when conditioned on $\tau_{\mathbf{0}}$, which allows for the construction of a sum of the typical variational lower bounds for learning the backward process' density function [22]. Since the backwards density also follows a Gaussian, it suffices to predict μ_θ and Σ_θ which parameterize the backwards distribution:

$$p_\theta\left(\boldsymbol{\tau}_{t-1} \mid \boldsymbol{\tau}_t\right) = \mathcal{N}\left(\boldsymbol{\tau}_{t-1}; \boldsymbol{\mu}_\theta\left(\boldsymbol{\tau}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\boldsymbol{\tau}_t, t\right)\right) \tag{4}$$

In practice, Σ_θ is often fixed to constants, but can also be learned through reparameterization. Following [23] we consider learning only μ_θ, which can be computed just as a function of τ_t and $\epsilon_\theta(\tau_t, t)$. One can derive that $\boldsymbol{\tau}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{\tau}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ for $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, through a successive reparameterization of (3) until arriving at $f(\boldsymbol{\tau}_t|\boldsymbol{\tau}_0)$. To sample from $p(\tau)$, we need only to learn ϵ_θ, which is done by regressing to the ground truth ϵ given by the tractable backwards density. Assuming we have ϵ_θ, we can then follow a Markov chain of updates that eventually converges to the original data distribution, in a procedure reminiscent of Stochastic Gradient Langevin Dynamics [25]:

$$\boldsymbol{\tau}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left(\boldsymbol{\tau}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{\tau}_t, t\right)\right) + \sigma_t \mathbf{z}, \quad \text{where } \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{5}$$

Thus, by using a variant of ϵ_θ conditioned on language to denoise our latent plans, we can effectively model $-\nabla_\tau \mathcal{P}_\beta(\tau \mid \mathcal{R})$ with our diffusion model, iteratively guiding our generated trajectory towards the optimal trajectories conditioned on language.

2.2 Model Architecture

It is computationally infeasible to operate directly on the pixel space. Instead, we do planning in latent state space by first compressing the visual input with a β-TCVAE [26], similar to latent diffusion models [19]. We choose the β-TCVAE for its efficiency in compression. The disentangled representation is a property that is also worth taking into consideration, and its effects on the denoising training process. It is still unclear whether disentanglement is beneficial or not. We adopt CLIP [27] as our textual encoder. CLIP is trained on large-scale image-text pairs and is able to align visual and textual input in its embedding space. Specifically, we use the Transformer variant [28] as the text encoder with the architecture modifications described in [29]. We use a modified temporal U-Net[30], which performs 1D convolution only across the time dimension, rather than the 2D convolution typical in text-to-image generation. This is motivated by our wish to preserve equivariance along the time dimension but not the state-action dimension. We modify the architecture in [30] by adding conditioning via cross attention in a fashion that resembles the latent diffusion model, but also uses a temporal convolution for projection to the token embeddings rather than the traditional 2D convolution. We use DDIM [24] during inference for increased computational efficiency and faster planning. DDIM uses strided sampling and is able to capture almost the same level of fidelity as typical DDPM sampling [23] with an order of magnitude speedup. For rolling out the latent plans generated by the denoiser, we resample a new plan with the frequency of H, until either the task is completed successfully or the maximum timestep is reached. In between samples, we roll out the open-loop plan without replanning. We set the total time horizon equal to $3H$ in our experiments, which means at maximum we will sample three latent plans from the model.

3 Experiments

3.1 Dataset and Metric

We evaluate a subset of the CALVIN benchmark [20] due to time, selecting the five most common subtasks (i.e tasks with the most pre-collected data) that do not belong to the same subtask category. This metric allows for fast experimentation since there is significantly less data to converge on, while still covering a diverse and broad range of tasks as we enforce that each task will belong to a different category. After pretraining our VAE on all data, we freeze the autoencoder and train the

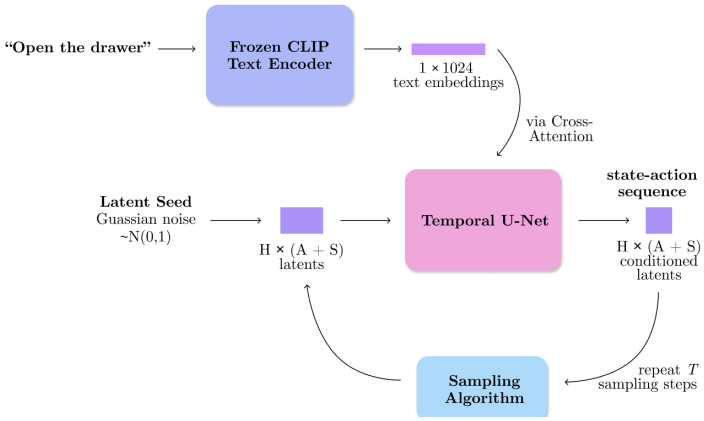

Figure 2: An overview of our inference pipeline. We pass sampled noise to our denoising autoencoder along with an encoded language prompt. The autoencoder is instantiated as a temporal U-Net. By repeating this process iteratively, we are able to generate high-fidelity latent plans conditioned on language. We are also able to decode the latent states into pixel space to analyze and interpret the plans generated by the denoising autoencoder.

U-Net just with data from the five tasks. This setup is motivated by the assumption that we only wish to evaluate the performance of the downstream decision-making, so we assume that there already exists a robust pre-trained autoencoder. In order to provide a fair comparison, we obtain results from our comparisons by freezing the autoencoders of their final checkpoint which was trained on all data and training the rest of the model on just the five tasks. We roll out all evaluated policies for 5 trajectories per task, for a total of 25 rollouts per policy. We compare the two strongest models released on this benchmark so far, HULC and MCIL [9, 31] All comparisons are trained in their official repository[1].

Task	MCIL	HULC	LAD
Place in Slider	1.0	0.8	1.0
Open Drawer	1.0	1.0	1.0
Move Slider Right	0.4	1.0	1.0
Stack Block	0.2	0.4	0.2
Lift Blue Block Table	0.4	0.6	0.4
Total (avg)	**0.60**	**0.76**	**0.72**

Table 1: Comparison of success rates between our diffusion model and prior benchmarks. Although our absolute performance does not beat the prior SOTA, we note that HULC is a significantly more complex model containing many inductive biases for robotics that may not be transferable to other RL environments, such as using a separate logistic loss for modeling the gripper action.

3.2 Effects of Diffusion

An interesting phenomenon we observe when rolling out with the diffusion model is its ability to robustly model arbitrary starting points of trajectories, leading to an ability to recover from failures in rollouts and attempt tasks again. For this reason, we find that simply by replanning more times instead of just rolling out a single plan improves the performance of LAD. There is much more to explore in this direction, and future rollout strategies that include conditioning on more than the last state or replanning more often than just once every H timesteps will likely lead to substantial gains in performance.

4 Conclusion

Learning the atomic sub-skills is critical to solving the multi-task planning problem and enabling the solving of more complex and open environments through state and temporal abstraction. We explore reformulating the language-conditioned planning process as the text-to-plan generation to better learn the alignment between language and state-action pairs. Experiments and qualitative analysis demonstrate the simplicity and effectiveness of our model. Future work looks to extend to the long horizon setting and further probing of the generalization and compositional capabilities of the model through classifier-free guidance or improved sampling methods, as well as incorporating value functions for guiding towards optimality.

[1] `https://github.com/lukashermann/hulc`

Acknowledgments

We would like to thank Ashley Jiang for creating the main figures, Michael Janner and Weixi Feng for fruitful discussions, and Oier Mees and Luka Shermann for clarifications and support on the CALVIN benchmark.

References

[1] Alan M Turing and J Haugeland. Computing machinery and intelligence. *The Turing Test: Verbal Behavior as the Hallmark of Intelligence*, pages 29–56, 1950.

[2] Rainer Bischoff and Volker Graefe. Integrating vision, touch and natural language in the control of a situation-oriented behavior-based humanoid robot. In *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)*, volume 2, pages 999–1004. IEEE, 1999.

[3] S. Levinson, Weiyu Zhu, Danfeng Li, K. Squire, Ruei sung Lin, M. Kleffner, M. McClain, and J. Lee. Automatic language acquisition by an autonomous robot. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 4, pages 2716–2721 vol.4, 2003.

[4] Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of language and perception for grounded attribute learning. *arXiv preprint arXiv:1206.6423*, 2012.

[5] Dipendra K Misra, Jaeyong Sung, Kevin Lee, and Ashutosh Saxena. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. *The International Journal of Robotics Research*, 35(1-3):281–300, 2016.

[6] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022.

[7] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents gpt-2 1.5b.

[8] Yujie Lu, Weixi Feng, Wanrong Zhu, Wenda Xu, Xin Eric Wang, Miguel Eckstein, and William Yang Wang. Neuro-symbolic causal language planning with commonsense prompting, 2022.

[9] Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data, 2022.

[10] Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. Grounded language learning in a simulated 3D world. *arXiv preprint arXiv:1706.06551*, 2017.

[11] Dzmitry Bahdanau, Felix Hill, Jan Leike, Edward Hughes, Arian Hosseini, Pushmeet Kohli, and Edward Grefenstette. Learning to understand goal specifications by modelling reward. *arXiv preprint arXiv:1806.01946*, 2018.

[12] Felix Hill, Andrew Lampinen, Rosalia Schneider, Stephen Clark, Matthew Botvinick, James L. McClelland, and Adam Santoro. Emergent systematic generalization in a situated agent, 2019.

[13] Cédric Colas, Ahmed Akakzia, Pierre-Yves Oudeyer, Mohamed Chetouani, and Olivier Sigaud. Language-conditioned goal generation: a new approach to language grounding for rl. *LAREL Workshop 2020*, abs/2006.07043, 2020.

[14] Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. A survey of reinforcement learning informed by natural language. *arXiv preprint arXiv:1906.03926*, 2019.

[15] Corey Lynch and Pierre Sermanet. Grounding language in play. *CoRR*, abs/2005.07648, 2020.

[16] Yiding Jiang, Shixiang Gu, Kevin Murphy, and Chelsea Finn. *Language as an Abstraction for Hierarchical Deep Reinforcement Learning*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[17] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[18] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR, 17–23 Jul 2022.

[19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[20] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 2022.

[21] Lilian Weng. What are diffusion models? *lilianweng.github.io*, Jul 2021.

[22] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

[24] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[25] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics.

[26] Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2018.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[29] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[30] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis, 2022.

[31] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. *Conference on Robot Learning (CoRL)*, 2019.

[32] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

[33] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

[34] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022.

[35] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation, 2022.

[36] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. 2021.

[37] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning, 2022.

A Qualitative Analysis

Figure 4: Comparison of a ground truth trajectory for the ”place in slider” task (left), the reconstructed trajectory (right), and the generated trajectory through denoising (bottom).

B Background and Related Work

Reinforcement Learning

We formulate the RL framework as Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \gamma, p)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, reward function $\mathcal{R}$, discount factor γ, and transition dynamics p. At each time step t, agents observe a state $s \in \mathcal{S} \subseteq \mathbb{R}^n$, take an action $a \in \mathcal{A} \subseteq \mathbb{R}^m$, and transition to a new state s' with reward r following $s', r \sim p(\cdot, \cdot|s, a)$. The goal of RL is then to learn either a deterministic policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ or a stochastic policy where $a \sim \pi(\cdot|s)$ with $a \in \mathcal{A}$ that maximises the policy objective, $J(\pi) = \mathbb{E}_{a\sim\pi, s'\sim p} \sum_{t=0}^{\infty} \gamma^t r_t$. Notice this is nothing but the expected discounted cumulative reward or expected return.

Language-Conditioned Policy Language-conditioned policies [15, 16, 13] have been explored in the reinforcement learning community to improve the ability to abstract the goal and the generalization to a new environment [10, 11, 12, 13]. However, these LCPs still struggle with long-horizon language commands. We are the first to leverage the advantage of the diffusion model in compositionality and long-horizon decision-making to address such challenges.

Diffusion in Offline Reinforcement Learning

Given the success of the denoising diffusion probabilistic models [23] (DPM) applied in text-to-image synthesis [32], the DPM has been further explored in both discrete and continuous data domains, including image and video synthesis [33, 34], text generation [35], and time series [36]. Diffusion planning [30] first proposed to transform the planning problem into inpainting and utilize diffusion models to solve the problem. Specifically, they diffuse the state and actions jointly to implement imitation learning and goal-conditioned reinforcement learning. This leverage the diffusion to solve long-horizon and compositionality issue in planning. Instead of predicting the whole trajectory for each state, [37] apply the diffusion model to sample a single action at a time conditioned with states. Analogous to the Diffuser [30] in trajectory-planning perspective and [37] in offline policy-optimization perspective, we are in the language-grounding perspective.