Universal Motion Generator: Trajectory Autocompletion by Motion Prompts

Felix Wang, Julie Shah
felixw@mit.edu

1 Introduction

Foundation models [1], which are large neural networks trained on massive datasets, have shown impressive generalization in both the language and vision domain [2][3]. While fine-tuning foundation models for new tasks at test-time is impractical due to billions of parameters in those models, prompts have been employed to re-purpose models for test-time tasks on the fly. In the language domain, a prompt can be an incomplete sentence for a large language model to auto-complete as seen in Fig. 2 (a) [2] [4]. In the vision domain, a prompt can be some pixels added to an image for a generic large vision model to perform specific classification tasks as seen in Fig. 2 (b) [5]. In both cases, prompts re-purpose generically trained foundation models with frozen weights to generate conditional distributions that solve specific downstream tasks.

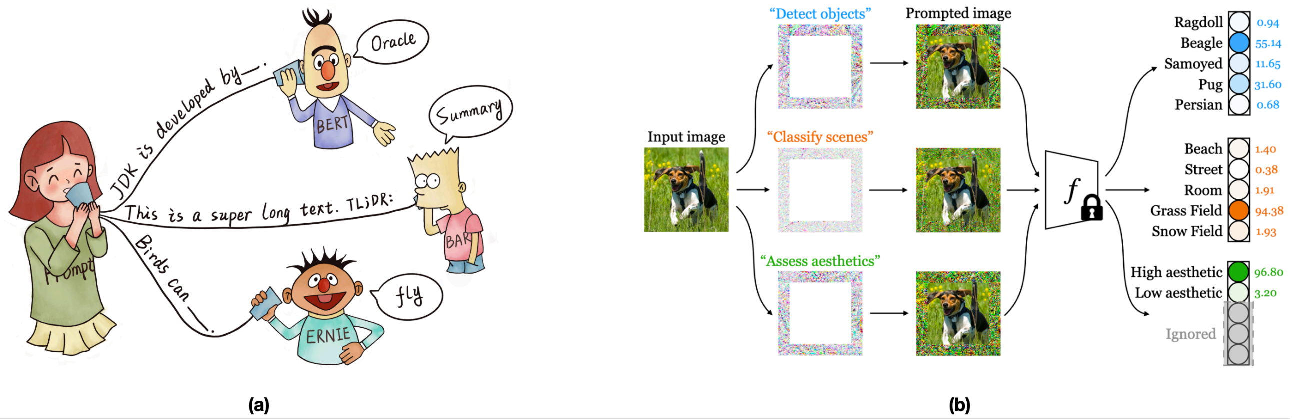

Figure 1: (a) Language prompts in the format of a question turn a foundation model into a question-answering machine, while prompts in the format of a long paragraph followed by a summary request turn the model into a reading comprehension engine. The model was originally trained on neither task. Figure borrowed from [4]. (b) Image prompts in the format of boundary pixels can condition the same model trained for generic classification to detect different but specific aspects of the same image. Figure borrowed from [5].

In this report, we ideate the equivalent foundation model for motion generation and the corresponding formats of prompt that can condition such a model. The central goal is to learn a behavior prior that can be re-used in a novel scene. Note the space of all possible trajectories is exponentially large depending on the number of discrete time steps, and a learned distribution of only meaningful trajectories can significantly reduce the search space given a new scene.

2 Problem Formulation

The ideal behavior of a universal motion generator is illustrated in Fig. 2. The model inputs (1) a RGBD image or a point cloud that encodes the geometry of a scene, and (2) an initial partial motion trajectory that encodes the intention of the user in the scene. The model should then auto-complete a collision-free trajectory that meaningfully manipulates objects in the scene as seen in Fig. 2(a)(b). The scene input should capture sufficient information about how to manipulate objects in the scene in a collision-free manner, while the trajectory input should narrow down the possible set of behaviors a user intends. Note there might be more than one behavior that can complete an initial partial trajectory, and the model should output a distribution over meaningful behaviors, e.g, object rearrangement [6]. Not meaningful behavior includes but is not limited to back-and-forth reaching or jittering without actually rearranging any objects.

Another possible way to use the model is to elicit cooperative/assistive behavior [7]. For example, if the model is trained on trajectories of two hands demonstrating a task together (e.g., setting a table), the user can give the trajectory of one hand as the initial trajectory prompt and the model should output the trajectory of the other hand to help with the task as seen in Fig. 2(c)-(d).

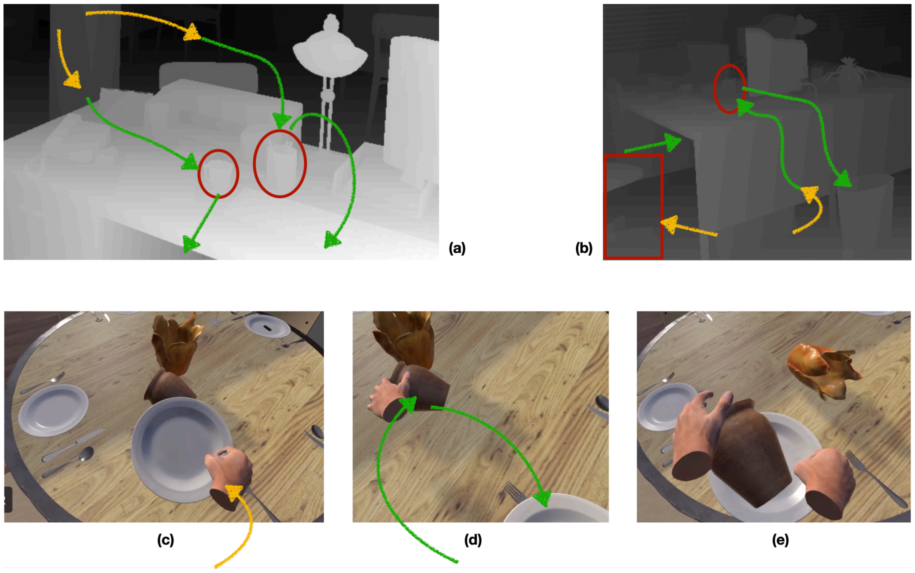

Figure 2: The desired behavior of a learned motion generator. (a)(b) Given the initial trajectory in yellow and the scene geometry in a depth image, the model auto-completes the green trajectory that both continues the intention of the yellow trajectories and avoids collisions in the depth image. Concretely, a sampled behavior from the model includes picking and placing different objects circled in scene (a), and pushing in a chair or tossing trash (b). (c)-(d) Given a demonstration of cooperative behavior with two hands, the model learns to control one hand (green) to help with the human-controlled hand (yellow) in a table-setting task.

3 Method

To model the behavior described in the previous section, we observe trajectory data share the same sequential nature as the language data [8]. We propose using the same architecture of large language models to model the distribution of meaningful behaviors. Specifically, we consider the architecture of [9], which handles both image and language prompts concurrently. In our setting, a similar model should handle both a scene prompt (we will use a depth image as an example) and a motion prompt (a partial trajectory in the form of a sequence of a robot end-effector poses) as shown in Fig. 3. During training, we tokenize the scene prompt and the motion prompt and feed them into a large language model to predict the future trajectory in a dataset autoregressively. During testing, we prompt the model with a scene and a partial trajectory to sample meaningful behaviors. Note the model rollout can deviate from a human user's intention, and in that case, the user will need to inject additional motion projects (e.g., through a VR controller) every a fixed number of time steps.

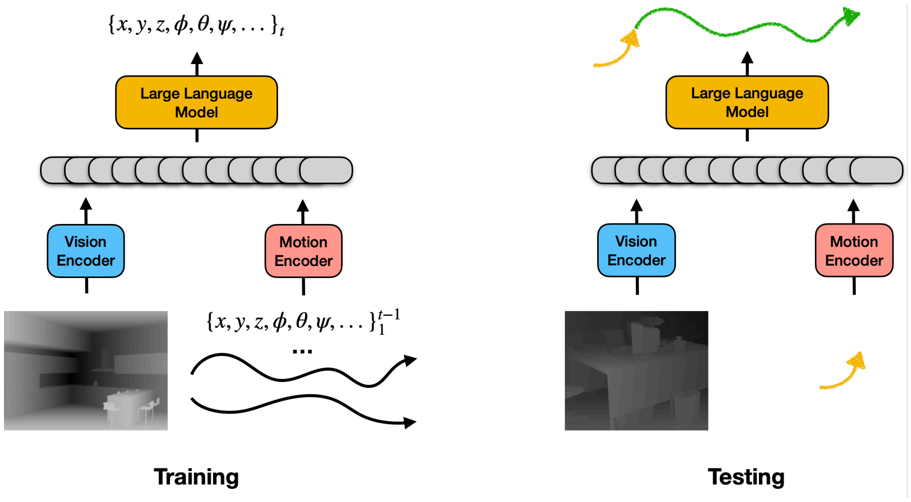

Figure 3: A schematic of model training and testing. Black trajectories come from a dataset, the yellow trajectory is generated by a human user, and the green trajectory is the prediction of the model.

4 Data Collection

In order to test the generalization of a foundation model of motion priors to novel scenes and motion prompts, we first need to collect large motion trajectory data to learn such a model. Existing datasets [10, 11, 12, 13] either focus on macro actions (not at the motion trajectory level) or contain only limited trajectories. Consequently, neither is suitable to train a foundation model at the motion level. We will explore generating data either through a task-and-motion planning (TAMP) algorithm [14] or human agents using a VR hand controller in a visually realistic simulator [15]. The benefits of using a planner to generate data are 1) scalability 2) embodiment in a robot. However, a planner may not be able to solve arbitrarily complex tasks and can be slow if the number of objects is large. Humans, on the other hand, are good at reasoning about a cluttered environment and can generate a demonstration fast. However, human demonstrations are not always directly transferable to a robot (lack of embodiment) and are less scalable.

The tasks we consider for the data generation are household activities in procedurally generated kitchen environments. Specifically, unpacking groceries, cooking a meal, serving a meal, and cleaning up. These long-horizon tasks provide a natural curriculum and hierarchy of subtasks (e.g., pick-and-place, open drawers) commonly tested in skill learning benchmarks [16, 17]. These benchmarks only test single skills in isolated environments (e.g. single kitchen appliance by itself). In contrast, all the motion trajectories are generated in a cluttered kitchen environment with many distractor objects. Thus it is critical for the model to infer intent from human motion prompts. Also giving high-level task goals such as cooking a meal can elicit very different demonstrations from different human agents, making a human dataset more diverse than a TAMP-generated dataset.

5 Conclusion

This blue-sky paper outlines our idea and a plausible implementation of a foundation model that learns re-usable motion priors. We discuss how to collect data to train such a model, and how to prompt the model to generalize to novel scenes as well as novel tasks once we have it. We hope this paper spurs research in the largely overlooked area of using large language models to learn motion trajectories in cluttered kitchen environments so that the community will have a pre-trained model for manipulation skills in the future.

References

[1] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[4] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

[5] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv preprint arXiv:2203.17274*, 2022.

[6] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020.

[7] Henry M Clever, Ankur Handa, Hammad Mazhar, Kevin Parker, Omer Shapira, Qian Wan, Yashraj Narang, Iretiayo Akinola, Maya Cakmak, and Dieter Fox. Assistive tele-op: Leveraging transformers to collect robotic task demonstrations. *arXiv preprint arXiv:2112.05129*, 2021.

[8] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34, 2021.

[9] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

[10] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on Robot Learning*, pages 477–490. PMLR, 2022.

[11] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Manipulathor: A framework for visual object manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4497–4506, 2021.

[12] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in Neural Information Processing Systems*, 34, 2021.

[13] Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Yuan-Hong Liao, Joshua B Tenenbaum, Sanja Fidler, and Antonio Torralba. Watch-and-help: A challenge for social perception and human-ai collaboration. *arXiv preprint arXiv:2010.09890*, 2020.

[14] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *arXiv preprint arXiv:2010.01083*, 2020.

[15] Chuang Gan, Siyuan Zhou, Jeremy Schwartz, Seth Alter, Abhishek Bhandwaldar, Dan Gutfreund, Daniel LK Yamins, James J DiCarlo, Josh McDermott, Antonio Torralba, et al. The threedworld transport challenge: A visually guided task-and-motion planning benchmark for physically realistic embodied ai. *arXiv preprint arXiv:2103.14025*, 2021.

[16] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pages 1094–1100. PMLR, 2020.

[17] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020.