Predicting Trajectories from Internet-Scale Egocentric Pet Videos

Danny Tran
UC Berkeley
dannyltran@berkeley.edu

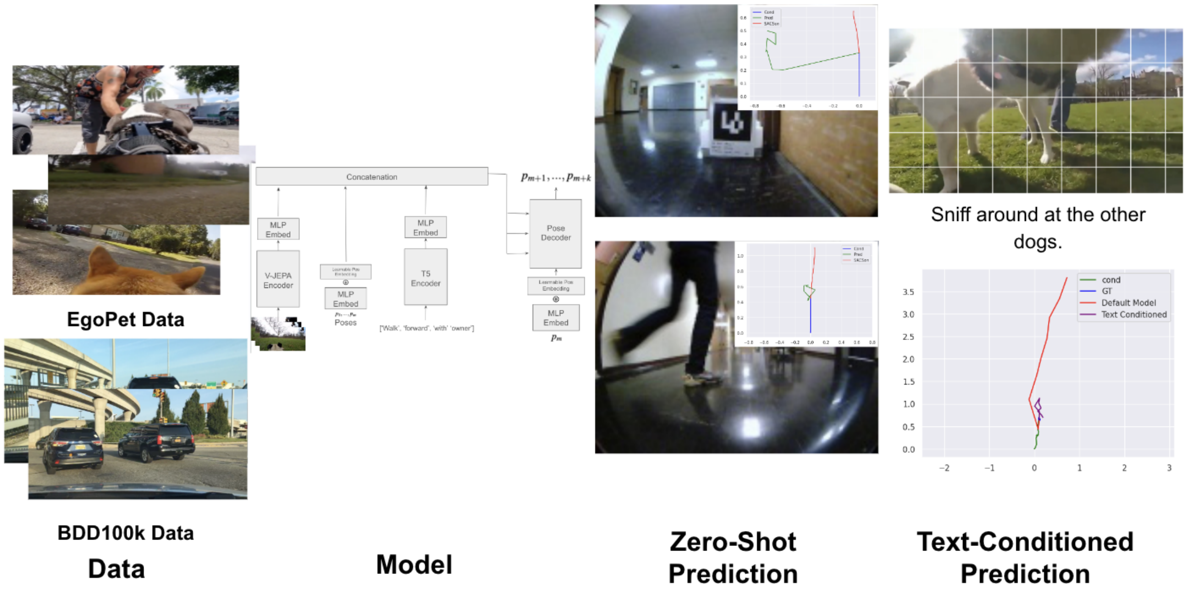


Figure 1. **Overview of our method.** Our model is trained on EgoPet [4] and BDD100k [33] using an autoregressive transformer model. We demonstrate that our model can be applied zero-shot to social navigation. We additionally, annotate a subset of EgoPet and finetune a model to be text-conditioned. However, our text-conditioned model does not quite work properly.

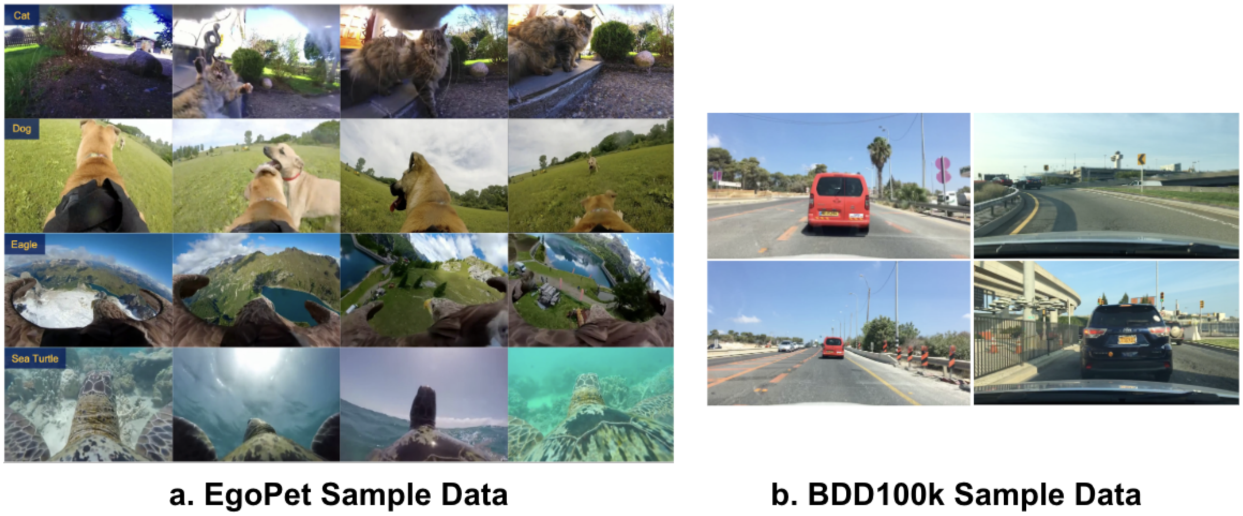
Abstract

Animals perceive the world to plan their actions and interact with other agents to accomplish complex tasks, demonstrating capabilities that are still unmatched by AI systems. To advance our understanding and reduce the gap between the capabilities of animals and AI systems, we make further progress along the locomotion prediction task from the original EgoPet Paper by pretraining with other datasets and various ablation experiments. We find that models trained on EgoPet are able to be applied zero-shot to robotic navigation tasks such as social navigation. We further annotate a subset of the EgoPet dataset with text commands and train a text conditioned model by finetuning our model on these annotations. However, we found that with our current setup this does not work properly with various shortcomings and possible future directions.

1. Introduction

Animals are intelligent agents that exhibit various cognitive and behavioral traits. They plan and act to accomplish complex goals and can interact with objects or other agents. Consider a cat attempting to catch a rat; this requires the cat to execute a precise sequence of actions with impeccable timing, all while responding to the rat's efforts to escape.

Current Artificial Intelligence (AI) systems can synthesize high quality images [24, 25], generate coherent text [7, 32], and even code Python programs [9]. But despite this remarkable progress, learning to understand the world as well as a cat remains hard to achieve. Recently, there has been a significant body of research in robotics aimed at



a. EgoPet Sample Data

b. BDD100k Sample Data

Figure 2. **a. EgoPet [4] video examples**. Footage of four different animal videos from an egocentric view are included. **b. BDD100k [33] video examples** Sample Data from the BDD100k dataset.

learning policies for quadruped locomotion, and other basic actions [2, 3, 10, 15, 16, 19, 21, 27]. However, these works rely on curated datasets which require a lot of effort to collect and highly specialized methods. We aim to leverage the variety of settings in EgoPet such as social interactions with humans with videos of dogs walking in crowds of people and off-road navigation with cats roaming the forests to create a more generalizable robot navigation model.

To achieve this we use an autoregressive transformer with the simple task of given a sequence of past m video frames $\{x_i\}_{i=t-m}^{t}$ and past m poses $\{p_i\}_{i=t-m}^{t}$, the goal is to predict the future trajectory of the agent $\{v_j\}_{j=t+1}^{t+k}$, where $v_j \in \mathbb{R}^{11}$ represents the relative location of the agent at timestep j. The first 2 values are the XZ values and the last 9 values are the unit normalized values of the rotation matrix. In practice, we condition models on $m = 8$ frames and predict $k = 8$ future locations which both correspond to 4 seconds into the future. We utilize a much simpler task compared to other works [12, 13] and hopefully allows us to capture nuances such as collision avoidance implicitly. To demonstrate our model's generalizability we run our model in the zero-shot setting to the HuRoN dataset to demonstrate that social navigation is implicitly learned from EgoPet.

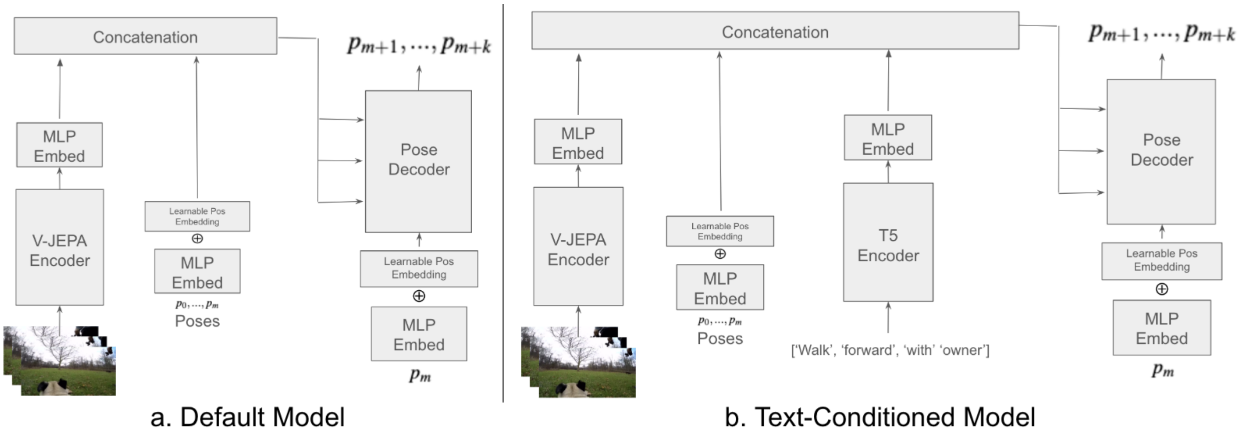
However, the agent's intent in these videos can be difficult to infer in this setting so we aim to better align the trajectory prediction with a set of intentions. To do this, we annotate a subset of the EgoPet dataset with text commands of what the agent does in the video. We then finetune a model on these text command annotations but find that this is a flaw in our method.

To summarize our contributions, we demonstrate that we developed a trajectory prediction model using the EgoPet dataset which demonstrates strong generalizability to robotic tasks such as social navigation when applied in the zero-shot setting. In addition, we additionally annotated a subset of the EgoPet dataset with descriptions of the agent's behavior develop a framework for creating a text-conditioned model with shortcomings.

2. Related Work

Egocentric Trajectory Prediction. Agents interact with the world from a first-person point of view, thus predicting an agent's trajectory can provide valuable insight to an agent's planning and interactions among other things. There are many works which do this with egocentric human videos such as [22, 23] which does this on human walking data, [28] which performs this on months of a single student's egocentric videos, [6] which performs this on egocentric basketball videos. These works explore a variety of different methods such as [22, 28] take a nearest-neighbor approach to this while other works such as [23] take an autoregressive approach to this problem.

Language-Conditioned Policies. Conditioning on language allows humans to have an easier interface to interact with models which makes it an area of interest for study. This can be seen in works such as [14, 18, 20, 26] which aim to condition robotic policies on text. Work such as [20] aims to build robots which can perform a variety of long horizon tasks conditioned on natural language such as robotic arm control. [14, 18] both tackle robotic navigation by fusing the visual and text information together. [26] creates a framework for using VLMs to annotate unstructured data to train a language conditioned robot policy for navigation.



a. Default Model

b. Text-Conditioned Model

Figure 3. **a. Default Model.** Represents that default autoregressive transformer model we employ. **b. Text-Conditioned Model.** Represents that text-conditioned model we use where we simply add a text-encoder.

3. Datasets

3.1. EgoPet [4]

EgoPet Trajectories. The EgoPet dataset is a unique collection of egocentric video footage primarily featuring dogs and cats, along with various other animals like eagles, wolves, turtles, sea turtles, sharks, snakes, cheetahs, pythons, geese, alligators, and dolphins. EgoPet was collected by scraping these egocentric animals videos from various online sources such as TikTok and YouTube. The dataset consists of 64.4% dog videos and 29.6% cat videos and is primarily skewed toward videos shorter than 30 seconds with a total of 84 hours of videos. EgoPet also has extracted trajectories from the videos using an odometry system called DPVO. Upon further analysis of these DPVO extracted trajectories we found that while the X (left-right) and Z (forward-backward) directions were relatively accurate for the majority of videos, the Y (up-down) directions were not accurate so we only use the XZ directions for our trajectory prediction. View Figure 2a. for EgoPet sample data.

EgoPet Text-Conditioned Annotations. In addition to the EgoPet dataset, I annotated the validation set and 200 videos from the training dataset. The annotation would consist of text description of what the agent does in the video and formulates it in a command style. In order to get more variety from the text annotations, for the training dataset we additionally prompt GPT-4 [1] to create 15 more variations of the text annotation.

3.2. BDD100k [33]

The Berkeley Deep Drive 100k dataset (BDD100k) is a driving dataset consisting of 100,000 driving videos for a total of 1,111 hours. BDD100k has driving videos from New York, Berkeley, San Francisco, and the Bay Area in a variety of different scenes and weather conditions. For our use case, we use the GPS location of the car and predict the trajectory from this GPS location. View Figure 2b. for BDD100k sample data.

4. Method

Our approach employs an autoregressive transformer tasked with the following: given a sequence of m past video frames $x_i{}_{i=t-m}^{t}$ and m past poses $p_i{}_{i=t-m}^{t}$, it aims to forecast the future trajectory $v_j{}_{j=t+1}^{t+k}$ of the agent, where $v_j \in \mathbb{R}^{11}$ denotes the agent's relative position at timestep j. The first 2 values of v_j represent the XZ coordinates, and the remaining 9 values are the unit normalized values of the rotation matrix. The models are conditioned on $m = 8$ frames and predict $k = 8$ future locations, which equates to a future timeframe of 4 seconds.

4.1. Architecture

Our trajectory prediction model employs an autoregressive decoder-only transformer. We use a vit-large V-JEPA [5] encoder for our video frames, a simple MLP to embed our past poses. We follow the LLAVA [17] multimodal approach by simply concatenating our frame and pose tokens together and then conditioning our decoder on these tokens. Then for the text-conditioned setting we use the large FLAN-T5 [11] text encoder to encode the text and then additionally concatenate these tokens with the pose and frame tokens. For our decoder, we use 24 layers and 24 heads with an embedding dimension of 1024. View Figure 3 for a general overview of the models we use. For our optimizer we use a schedule free optimizer with learning rate 1e–8.

4.2. Prediction Setting

For our prediction setting we found that the DPVO generated trajectories were inaccurate along the Y-axis so we only predict the XZ axes. Additionally, we predict the el-

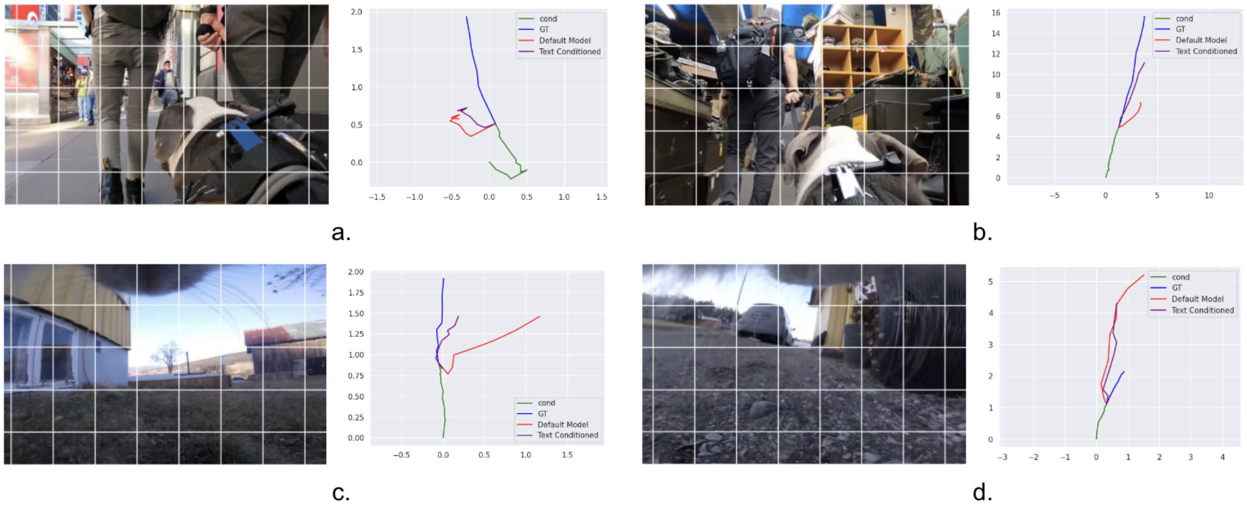


Figure 4. **a.** The text condition is ’Walk around along the sidewalk trailing behind owners.’ and we see the both the default model and text-conditioned model goes toward the left so generally both do not do that well in this case. **b.** The text condition is ’Follow owner forward in the store and turn around with owner.’ and we see that the text-conditioned model goes forward more closely to groundtruth but the default model goes slightly forward to the right. **c.** The text condition is ’Walk forward onto the deck and then veer left to the other edge of the house.’ and we see that the default model goes toward the right into the field but the text-conditioned model does go more forward as similar in groundtruth. **d.** The text condition is ’Walk up to the door on the right and walk into the house while not hitting the baby. Then go up the stairs to your food and eat.’ and we see that both the default model and the text-conditioned model go forward toward the left which is close to where the door is but the pacing is a bit off from groundtruth.

ements of the rotation matrix which we invidually element wise normalize and predict the normalized values. We sample 8 second videos, where the first 4 seconds are our conditioning and we predict the poses for the next 4 seconds. We perform this at 2 Hz for a total of conditioning on 8 poses and frames and predicting the next 8 poses. We set up the prediction as the setting where we predict the poses relative to the first pose. For our text-conditioning we simply add the text tokens to the conditioning and predict the same trajectory. When we train our text-condition model we use a checkpoint of the default model trained to 40 epochs and finetune the model on the text-conditioned clips for 10 epochs.

4.2.1 Zero-Shot HuRoN Dataset Evaluation

5. Experiments

To evaluate our model’s predicted poses 8 timesteps into the future corresponding to 4 secds we form trajectories from these predicted motions and compute the RMSE of the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) metrics against the ground truth trajectories. ATE and RPE are commonly employed metrics for evaluating systems such as SLAM and visual odometry [8, 30, 31, 34]. ATE first aligns the ground truth with the predicted trajectory, and then computes the absolute pose difference. RPE measures the difference between the predicted and ground truth locomotion [29]. In this section, we ablate various parts of our method.

5.1. Model Architecture

In this experiment, we examine our choice of model between the usage of an encoder and decoder for our poses or just an decoder-only model. The encoder + decoder model has 16 layers and 16 heads for both the pose encoder and pose decoder. The decoder only model on the other hand has 24 layers and 24 heads. This experiment is conducted over 40 epochs. We see from Table 1 that while the Encoder + Decoder model performs better on ATE and RPE Rotation, our method prioritizes RPE Translation performance. It is also possible in this experiment that we did not train the model for long enough given that the decoder only model has more capacity.

Table 1. Model Architecture Ablation

	ATE	RPE Trans.	RPE Rot.
Encoder + Decoder	**0.672**	0.587	**10.51**
Decoder Only	0.701	**0.569**	10.90

5.2. Pose Orientation

In this experiment, we examine our choice of predicting the future pose orientation as quaternion coordinates or as the normalized elements of the rotation matrix. In this experiment the model remains the same and both use schedule free

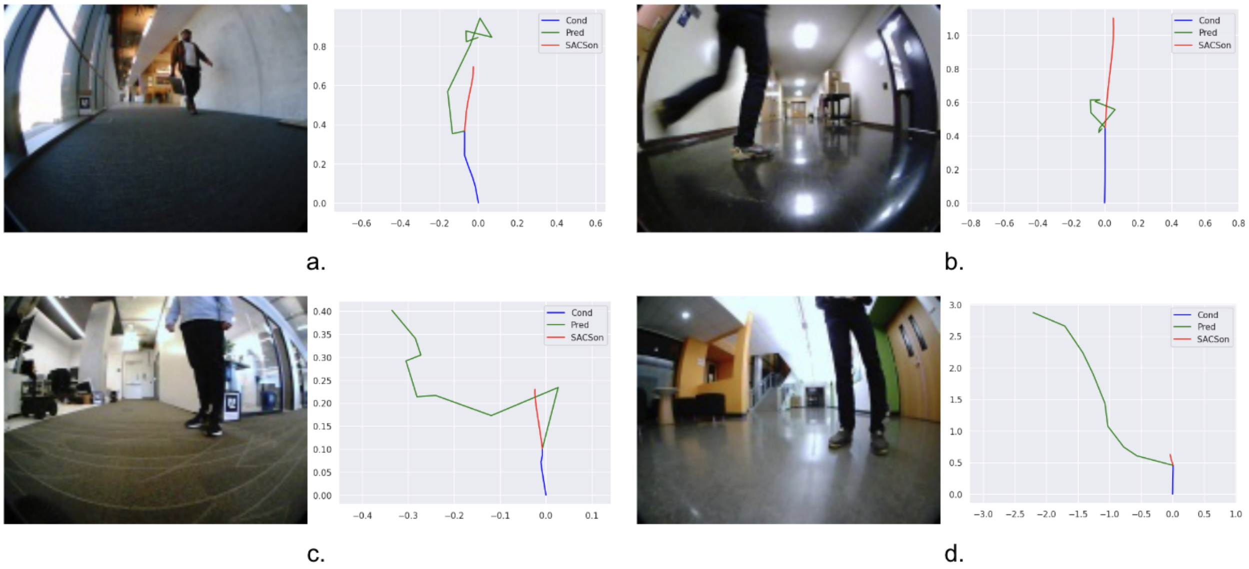


Figure 5. **a.** Our model seems to go forward similar to SACSon but also goes more towards the left possibly due to the person walking to the right. **b.** Our model seems to stay in place to avoid walking into the human in front. **c.** Our model goes left and then forward to walk around the human. **d.** Our model goes left around human.

training and we conduct this experiment over 10 epochs. We find that predicting the rotation matrix elements leads to a considerably lower RPE Rotation value.

Table 2. Pose Orientation Ablation

	RPE Rot.
Quaternion Coordinates	14.96
Rotation Matrix	**9.88**

5.3. Model Conditioning

In this experiment, we study the impact of conditioning our model on the frames and poses. We conduct 3 experiments, 1 conditioned on both frames and poses, 1 conditioned on only frames, and 1 conditioned on only poses. From Figure 3 we can determine that generally the Frames and Poses model performs the best in terms of RPE Translation and does marginally worse than the Frames only model in terms of ATE. The Poses only model performs the best in terms of RPE Rotation. Generally it seems like the Frames and Poses model performs the best but likely reveals a more nuanced impact for these conditionings.

Table 3. Model Conditioning Experiment

	ATE	RPE Trans.	RPE Rot.
Frames & Poses	0.693	**0.564**	10.853
Frames only	**0.692**	0.579	10.828
Poses only	0.841	0.725	**10.607**

5.4. BDD100k Data

In this experiment we experiment the impact of using BDD100k data during training. Also a small note is that during training, since we do not have pose orientations from BDD100k we only have a translation loss for the BDD100k data. For using the BDD100k data we use 50% EgoPet data and 50% BDD100k data. We report the EgoPet model at 40 epochs as the validation metrics start to go up after 40 epochs. Similar, for the EgoPet + BDD100k model we report the model at 80 epochs as the validation metrics start to go up after 80 epochs. We find that while RPE translation is marignally lower when using BDD100k data, ATE and RPE Rotation improve.

Table 4. BDD100k Data Experiment

	ATE	RPE Trans.	RPE Rot.
EgoPet	0.693	**0.564**	10.853
EgoPet + BDD100k	**0.664**	0.566	**10.149**

6. Results

6.1. Quantitative Results

From Table 5 we can see that our Default Model and Text-Conditioned Model both outperform the baseline of averaged trajectory. We also see that our text-conditioned model performs better than the Default Trajectory. However from

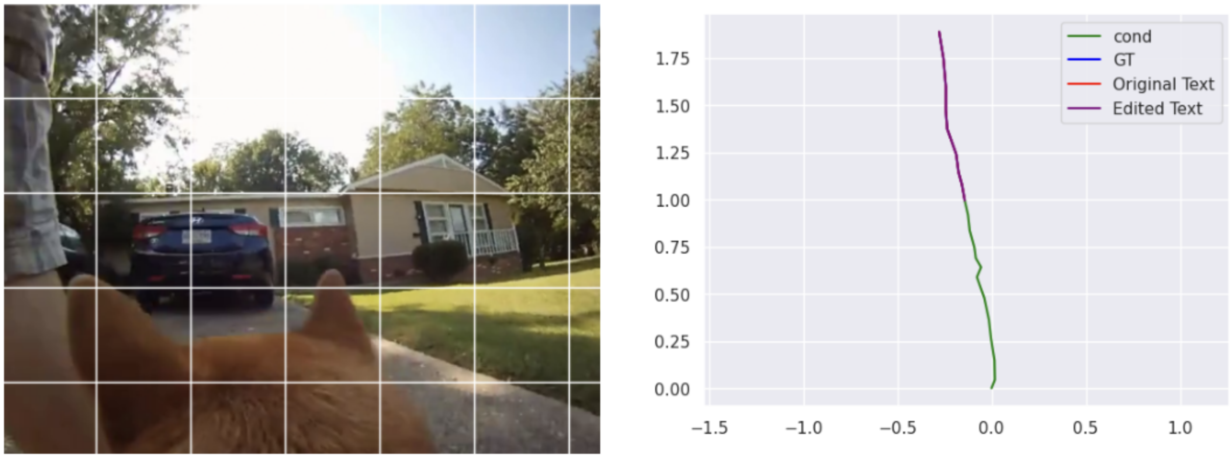


Figure 6. We explore command editing where the actual text is ’Walk forward to the door of the house.’ but we edit it to ’Turn right into the grass field.’ where we see that the model’s prediction does not change when the text conditioning changes which suggests that the model completely ignores the text conditioning.

Figure 6 we can determine that our model is actually ignoring the text-conditioning so the text-conditioning is actually likely not the source of performance boost. We suspect that this performance boost actually comes from when annotating the text-conditiong videos we only annotated videos where the trajectory from DPVO made sense so finetuning on this more accurate subset is a possible source for this boost in performance.

Table 5. Quantitative Results: The Average baseline is simply the average trajectory from the training dataset. The Default Model is our Decoder Only model trained on EgoPet for 40 epochs. The text-conditioned model is the Default Model finetuned for the text-conditioned case for 10 epochs.

	ATE	RPE Trans.	RPE Rot.
Average (Baseline)	0.904	1.332	67.878
Default Model	0.693	0.564	10.853
Text-Conditioned Model	**0.635**	**0.549**	**10.878**

6.2. Qualitative Results

6.2.1 EgoPet Visualizations

In this section, we visualize our model on some of the EgoPet videos with both the default model and the text-conditioned model. From Figure 4 we can see that the predicted trajectories are fairly sensible with navigating the different scenarios. It does seem like the text conditioning improves the qualitative performance as well. However, from Figures 6 it seems the model completely ignores the text conditioning which suggests that the text conditioning was not the cause of the performance boost. I suspect that the performance actually came from the finetuning stage on top of the model where I would only text annotate clips with a good trajectory so it is possible that more clips with more correct trajectories are actually the cause of the performance boost.

6.2.2 Zero-Shot HuRoN Dataset Evaluation

In this section, we apply our model on our model on the HuRoN dataset [13] which is a social navigation dataset and tests the model’s ability to perform in the presence of humans and other obstacles. From the visualizations present in Figure 5 we can see that our model is able to perform relatively well in the zero-shot setting on the HuRoN dataset making sensible choices such as avoiding humans or moving around obstacles.

7. Limitations

This work primarily works with predicting trajectories from the videos and evaluating on videos from robotic datasets but is not actually tested on a real robot limiting us from understanding the true potential of such a model. In addition, the text-condition annotations are very lacking with only 200 training videos and require a lot of time to annotate. Future work could include the usage of works such as [26] which could produce a more scalable annotation process.

8. Conclusion

This work presents a setup to leverage the internet-scale dataset EgoPet [4] by using the DPVO extracted trajectories on the videos and a simple autoregressive transformer model with the simple task of predicting the future poses. We find that this task can be aided by the use of other datasets such as BDD100k [33]. In addition, we attempt to text condition our model but find there are some shortcoming with our method. Finally, we see that our model is able to be applied zero-shot to social navigation in the HuRoN dataset [13].

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3

[2] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on Robot Learning*, pages 403–415. PMLR, 2023. 2

[3] Andrea Bajcsy, Antonio Loquercio, Ashish Kumar, and Jitendra Malik. Learning vision-based pursuit-evasion robot policies. *arXiv preprint arXiv:2308.16185*, 2023. 2

[4] Amir Bar, Arya Bakhtiar, Danny Tran, Antonio Loquercio, Jathushan Rajasegaran, Yann LeCun, Amir Globerson, and Trevor Darrell. Egopet: Egomotion and interaction data from an animal's perspective. *arXiv preprint arXiv:2404.09991*, 2024. 1, 2, 3, 6

[5] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-jepa: Latent video prediction for visual representation learning. 2023. 3

[6] Gedas Bertasius, Aaron Chan, and Jianbo Shi. Egocentric basketball motion planning from a single first-person image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5889–5898, 2018. 2

[7] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023. 1

[8] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 4

[9] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. 1

[10] Suyoung Choi, Gwanghyeon Ji, Jeongsoo Park, Hyeongjun Kim, Juhyeok Mun, Jeong Hyun Lee, and Jemin Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8(74):eade2256, 2023. 2

[11] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. 3

[12] Nitish Dashora, Daniel Shin, Dhruv Shah, Henry Leopold, David Fan, Ali Agha-Mohammadi, Nicholas Rhinehart, and Sergey Levine. Hybrid imitative planning with geometric and predictive costs in off-road environments. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4452–4458. IEEE, 2022. 2

[13] Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. Sacson: Scalable autonomous control for social navigation. *IEEE Robotics and Automation Letters*, 2023. 2, 6, 7

[14] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023. 2

[15] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. 2021. 2

[16] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020. 2

[17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 3

[18] Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, pages 23301–23320. PMLR, 2023. 2

[19] Gabriel B Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. *arXiv preprint arXiv:2205.02824*, 2022. 2

[20] Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters*, 7(4):11205–11212, 2022. 2

[21] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022. 2

[22] Hyun Soo Park, Jyh-Jing Hwang, Yedong Niu, and Jianbo Shi. Egocentric future localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4697–4705, 2016. 2

[23] Jianing Qiu, Lipeng Chen, Xiao Gu, Frank P-W Lo, Ya-Yen Tsai, Jiankai Sun, Jiaqi Liu, and Benny Lo. Egocentric human trajectory forecasting with a wearable camera and multimodal fusion. *IEEE Robotics and Automation Letters*, 7(4): 8799–8806, 2022. 2

[24] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 1

[25] 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. 1

[26] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lmnav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pages 492–504. PMLR, 2023. 2, 6

[27] D Shah, A Sridhar, N Dashora, K Stachowicz, K Black, N Hirose, and S Levine. Vint: A large-scale, multi-task visual navigation backbone with cross-robot generalization. In *7th Annual Conference on Robot Learning*, 2023. 2

[28] Krishna Kumar Singh, Kayvon Fatahalian, and Alexei A Efros. Krishnacam: Using a longitudinal, single-person, egocentric dataset for scene understanding tasks. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016. 2

[29] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 4

[30] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 4

[31] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *Advances in Neural Information Processing Systems*, 36, 2024. 4

[32] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[33] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 1, 2, 3, 7

[34] Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *European Conference on Computer Vision*, pages 523–542. Springer, 2022. 4