When Absolute State Fails: Evaluating Proprioceptive Encodings for Robust Manipulation

Maxime Alvarez[1,2], Ryo Watanabe[1], Paul Crook[1], Afshin Zeinaddini Meymand[1], Suvin Kurian[1], Pablo Ferreiro[1], Genki Sano[1]

***Abstract*— As end-to-end robotic policies are progressively deployed in the real world to solve real tasks, they face a gap between the training and inference conditions. Scaling the amount and diversity of the training data has shown some success in improving zero-shot generalization, yet robots still fail when faced with new, unseen test conditions. For instance, while robots with fixed frames of reference are common, those with moving frames pose a greater challenge for deployment. To address this specific instance of the issue, we present a study of strategies for encoding the robot's proprioceptive state to improve both in- and out-of-distribution performance at test time. Through a systematic study of joint representations, we find that a simple episode-wise relative frame provides the best trade-off between task performance and robustness, outperforming the baselines in extensive real-robot experiments conducted in a realistic test environment. The results suggest a practical path to leveraging data collected by robots with varying frames of reference and deployment to unseen test configurations.**

I. INTRODUCTION

The transition of robotic manipulation from static, tightly controlled laboratory workcells to unstructured, dynamic environments represents one of the field's most pressing challenges. As robots are increasingly deployed in real-world settings, such as retail spaces, warehouses, and domestic environments, they are often equipped with mobile bases [1], [2], [3] or linear rail systems [4], [5] to extend their operational workspace. While imitation learning (IL) [6] has demonstrated remarkable success in teaching these robots complex tasks from human demonstrations, the policies inherently struggle to generalize when deployed under initial conditions that differ even slightly from their training data.

A critical but often overlooked source of this brittleness lies in how a robot's own physical state is represented. When a robot operates from a fixed base, its internal coordinate frame aligns consistently with its workspace. However, when a robot features moving frames of reference, such as a manipulator mounted on a mobile carriage or linear rails, the absolute values of its proprioceptive state become highly variable across episodes. If an end-to-end policy is trained on absolute joint positions, an identical manipulation task executed from a starting position offset by just a few centimeters can result in catastrophic failure or erratic, unsafe movements.

[1] TELEXISTENCE Inc, Foundation Model Division, Japan.
[2] The University of Tokyo, Japan.

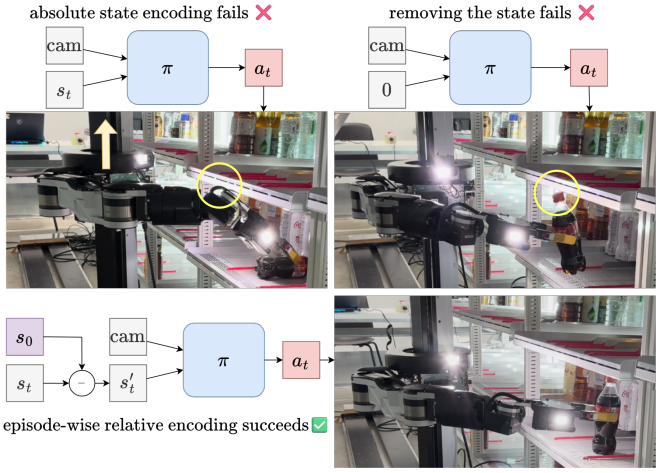

Fig. 1. Comparison of the three state-action representations. Absolute state/action encoding fails and can produce unsafe behavior. Removing the state and using chunk-wise actions improves robustness but still suffers from drift, especially in precision-critical stages. The proposed episode-wise relative encoding strikes a balance between the two and successfully solves the task. In the top-left example, the wrist collides with the upper shelf, leading to failure. In the top-right example, the bottle tilts after collision. A video digest of our work is available here: https://drive.google.com/file/d/10lj__pkWFizmhWWDh1xFBfZVfd41_fuP/

To date, the overwhelming majority of research aimed at closing this train-test distribution gap has focused on the visual domain. Massive strides have been made in extracting robust, invariant visual features [7], [8] ; however, proprioceptive inputs, such as joint positions and velocities, are still frequently fed into neural policies as raw, absolute numeric values. While some recent frameworks have begun exploring action spaces relative to chunk histories [9] or external object poses [10], [11], these methods either lose critical global state information or demand complex external pose estimators. The effectiveness of a purely intrinsic, robust proprioceptive encoding remains underexplored.

In this work, we address the specific challenge of proprioceptive drift and coordinate variability in mobile manipulation. We investigate alternative state- and action-representation strategies for a dual-rail robotic system tasked with precise visual manipulation. By moving away from absolute state encodings, we propose a simple yet highly effective episode-wise relative coordinate frame. This approach dynamically standardizes the robot's starting position at the start of each episode, effectively collapsing variance in ini-

tial conditions without sacrificing the fine-grained precision required for tasks such as grasping and object placement.

We evaluate our representations using an Action Chunking Transformer (ACT) [12] architecture on a real-world bottle recovery task.

Our main contributions are as follows:

- An investigation of three different state and action representations to study their impact on policy performance.
- Demonstration of significant improvement and robustness through extensive testing on a real robot in a realistic test environment.
- Identification that an episode-wise relative frame strikes a balance between generalizing to new states and exploiting state information, effectively collapsing the variance in starting conditions so the model learns more efficiently.

See Figure 1 for an illustration of the representation strategies and instances of failures.

II. RELATED WORKS

As highlighted by [13], integrating proprioception information into vision-based policies can be difficult, and the literature is unclear about the benefits of doing so. As a consequence, various representations have emerged over time.

Relative state representation. Robot-frame relative actions are used in reinforcement learning. The robot defines an egocentric static frame, and actions are specified relative to that frame, and typically do not use world information, except in cases such as multi-agent [14]. This is, for instance, the case in walking [15]. In ZeroMimic [16], an ablation suggests that relative actions are superior to absolute actions in the camera frame. In UMI [9], the authors define a relative frame using historical observations. In our case, we do not have a history of observations, so we replace the chunk-relative state with a constant 0 state and use a similar approach of chunk-relative actions.

Object-oriented frame coordinates. Some works use poses relative to an object in the world, which requires information outside of the robot [10], [11]. Here, we use only the robot's intrinsic state.

State-free policies. By extending the visual observations available to the policy and using delta end-effector actions, State-Free Policies [17] have demonstrated stronger generalization in height and horizontal task placement variations. Although we use a state-free policy with chunk-wise delta actions as one of our baselines, we do not change the hardware to include additional vision inputs. In our task, the "full task observation" (as defined in [17]) is already available through the given visual inputs.

III. METHOD

To study the impact of state and action representations on policy performance in- and out-of-distribution, we evaluate three different state and action representations. The robot's state has 10 components, of which components 0 and 1 correspond to the linear rails controlling the position of the shoulder of the robot on the 2D plane facing the task environment. For simplicity, when referring to the state components 0 and 1 ($s_{\{0,1\}}$) we write s. The other components of the state are only normalized using the training dataset's statistics. For our robot control software, the actions are target joint commands, meaning that the actions are target states. The same components 0 and 1 in the actions refer to the 2D plane control, and we apply the same shortcut notation a for $a_{\{0,1\}}$. We compare:

- **Absolute state and absolute actions**: the state is the current absolute joint values from the robot, and actions are the target absolute joint values sent to the robot system.
- **Episode-wise state and episode-wise actions**: at the beginning of each episode, the current absolute value of the state is defined as the origin, and all states and actions are encoded relatively to it. Essentially, this approach reduces to subtracting the state of each episode at time 0 from the state of the rest of the time steps.

$$s'_{i,t} = s_{i,t} - s_{i,0}$$

$$a'_{i,t} = a_{i,t} - s_{i,0}$$

  where i is the index of the episode, and t is the timestep within the episode. See Figure 2 for an illustration of the episode-wise relative state.
- **No state and chunk-wise actions**: following UMI [9], the state and actions are defined relative to the first state in the observation history. Given that in our setting we use a single time step of observation, the state is effectively 0. The actions are defined relative to the state at time t for the entire action chunk.

$$s'_{i,t} = \emptyset$$

$$a'_{i,t+k} = a_{i,t+k} - s_{i,t},\ \forall k \in [0, H]$$

  where H is the length of a single action chunk and k is the index of the action within the chunk. This also an implementation of [17] in the context of our task.

For all other joints in the robot, the state and actions are represented in absolute values. We keep the other joints absolute as they are independent of the robot's position on the rails at the start of the episode.

IV. EXPERIMENTS

A. Task

We reuse the bottle recovery task and the Ghost robot [5] presented in FTACT [18], but extend the task from a single starting position in front of a table to a variable starting position at the back of a convenience store shelf. See Figure 3 for an illustration of the task.

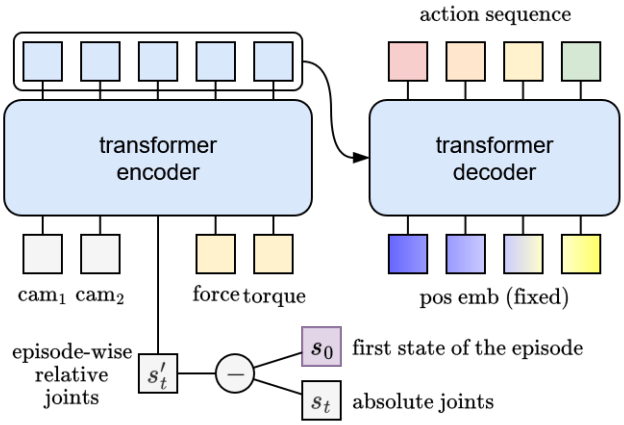

Fig. 2. Illustration of the proposed episode-wise relative state and actions method applied to ACT.

B. Model

We use the ACT model [12], replacing the ResNet vision encoder with DINOv2-small [19] and increasing the decoder's number of layers to 9. We use a cosine decay learning rate scheduler with a warmup phase of 1000 steps and a decay period of 150000 steps. We set the peak learning rate to 1.0e-4 and the decay learning rate to 1.0e-5. We keep the vision encoder frozen during training. We train on a single node with 8 NVIDIA H200 GPUs, using a batch size of 128 per GPU for a total of 200k gradient steps, with full FP32 precision.

C. Dataset

The dataset we use for this experiment has been collected by teleoperators solving the given task. There are 2500 episodes, totaling around 70h of data, at 50 Hz. The policy is fed three images: two images coming from the stereo cameras that form the robot's "head" and which are mounted on the arm carriage, and the third from a wrist camera mounted at the end of the arm. All images are resized to 224x224 to fit the pretrained vision encoder. The state has 10 dimensions: 2 dimensions are the X and Z rails, 7 dimensions correspond to the robot's arm, and the last dimension is the gripper. To that state, we concatenate a 6D force-torque sensor calibrated to 0 at the beginning of the episode.

The data collection was done in 2 environments, A and B, and the evaluation was carried out in environment B only. The difference between the two environments lies in the shelf positions, bottle arrangements, and the distance of the robot rail from the base of the shelves, the latter varying by a few centimeters.

In Figure 4, we show the starting position in X and Z positions for all episodes in the training dataset, when represented in absolute values.

D. Evaluation

We evaluate the 3 policies in-distribution and out-of-distribution, defining out-of-distribution as any X/Z position unseen in the training dataset and unseen bottle types. In total, we evaluate 6 Z values (4 for the in-distribution evaluations and 2 for the out-of-distribution) and 36 X values. The changes in Z values are significant (from one shelf to another) while the changes in X values are small (bottles being next to each other). For the out-of-distribution X-values, we use a different cabinet. See the Appendix for an illustration of the test environment with the robot.

We score each episode on a 0 to 4 scale as such:

- 0 when the robot cannot grasp the bottle
- +1 when the robot can grasp the bottle
- +1 when the robot can upright the bottle
- +1 when the robot puts the bottom of the bottle on the shelf
- +1 when the robot opens the gripper and the bottle is stable (does not fall over)

This leads to a total of 4 points when the episode is successful. Each episode has a timeout of 5 minutes, beyond which the episode is halted, and the score is tallied.

V. RESULTS

In Table I and Figure 5, we present the results from the evaluation for in-distribution and out-of-distribution as well as the average of both. We report the mean score with the standard deviation of the score. In Table II, we present the success rate for a binary success/failure scoring of the same evaluation episodes.

TABLE I
AVERAGE SCORES (MEAN ± SD) FOR ID AND OOD EVALUATIONS.

	Abs/Abs	**Eps/Eps**	**0/Chunk**
In-distribution	0.75 ±1.25	**3.30 ±0.80**	2.25 ±1.52
Out-of-distribution	0.00 ±0.00	**2.94 ±0.85**	2.56 ±0.96
Total	0.38 ±0.99	**3.12 ±0.83**	2.41 ±1.29

TABLE II
SUCCESS RATE FOR ID AND OOD EVALUATIONS.

	Abs/Abs	**Eps/Eps**	**0/Chunk**
In-distribution	5.0%	**50.0%**	20.0%
Out-of-distribution	0.0%	**25.0%**	20.0%
Total	2.5%	**37.5%**	20.0%

The **absolute state and absolute actions** setting (noted Abs/Abs) performs the worst. In distribution, the policy failed consistently to grasp the bottle and succeeded in grasping it 6 times, and managed to fully complete the task only once. During the out-of-distribution evaluation, the policy displayed dangerous movements, prompting us to halt the evaluation and give a 0 score for out-of-distribution.

The **no state and chunk-wise actions** setting (noted 0/Chunk) performs poorly, completely failing to grasp the bottle in 5 out of 36 cases. Interestingly, this policy does not suffer any performance loss from the out-of-distribution cases, even improving the average score (+13%). The average score standard deviation is higher than all other policies, showing unreliable performance.

Fig. 3. Bottle-recovery task decomposed into four stages: (a) *Start*—initial state, (b) *Pick*—approach and grasp the bottle, (c) *Press*—move to the shelf edge and press the bottle against it, (d) *Place*—move to the shelf and set the bottle upright.

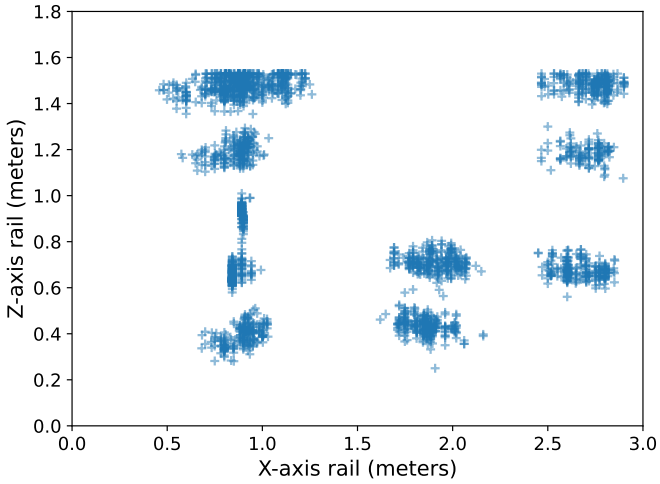

Fig. 4. Distribution of the starting values for each episode in the dataset of the X (on the x-axis) and Z (on the y-axis) rails of the robot.

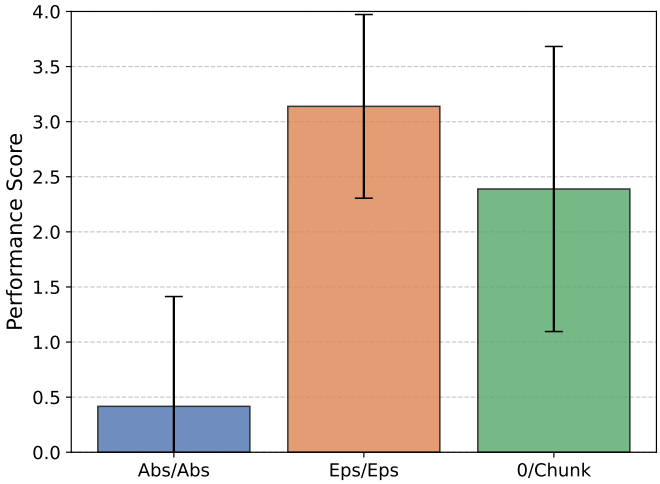

Fig. 5. The **episode-wise state and episode-wise actions** (Eps/Eps) performs the best, with **no state and chunk-wise actions** (0/Chunk) performing slightly lower and with a much higher variance, while **absolute state and absolute actions** (Abs/Abs) fails completely.

The **episode-wise state and episode-wise actions** setting (noted Eps/Eps) performs the best overall but suffers from some performance degradation (-11%) in the out-of-distribution setting. This setting shows the best average performance and seems to strike a compromise between generalizing to new states and exploiting the state information for improved performance in the in-distribution setting. As detailed in Appendix A, this is likely because the episode-wise relative representation effectively collapses the variance in starting conditions, allowing the model to learn more efficiently across all training episodes.

VI. CONCLUSION

In this work, we studied 3 different approaches to encoding the state and actions of linear joints and found that while not inputting the state at all (0/Chunk setting) leads to less performance drop when in the OOD setting, its performance in the ID setting is lower than that of relative episode-wise state and actions. Episode-wise state and actions get the best performance overall at the price of a drop in average score when in the OOD setting.

While this paper studies a specific task and a specific robot, the results are expected to hold with other robots in other settings that also have linear joints, such as the Agitbot G1, where the torso is set on a vertical rail.

The current methods still suffer from generalization issues when out of distribution, as is common for imitation learning-based methods. Also, we leave for future research the study of the case of non-linear joints, as the episode-wise approach might not be applicable.

REFERENCES

[1] R. Takanami, P. Khrapchenkov, S. Morikuni, J. Arima, Y. Takaba, S. Maeda, T. Okubo, G. Sano, S. Sekioka, A. Kadoya, M. Kambara, N. Nishiura, H. Suzuki, T. Yoshimoto, K. Sakamoto, S. Ono, Y. Ko, D. Yashima, A. Horo, T. Motoda, K. Chiyoma, H. Ito, K. Fukuda, A. Goto, K. Morinaga, Y. Ikeda, R. Kawada, M. Yoshikawa, N. Kosuge, Y. Noguchi, K. Ota, T. Matsushima, Y. Iwasawa, Y. Matsuo, and T. Ogata, "Airoa moma dataset: A large-scale hierarchical dataset for mobile manipulation," *arXiv preprint*, 2025.

[2] J. Wu, W. Chong, R. Holmberg, A. Prasad, Y. Gao, O. Khatib, S. Song, S. Rusinkiewicz, and J. Bohg, "Tidybot++: An open-source holonomic mobile manipulator for robot learning," in *Conference on Robot Learning*, 2024.

[3] D. Taranta, F. Marques, A. Lourenço, P. A. Prates, A. Souto, E. Pinto, and J. Barata, "An autonomous mobile robot navigation architecture for dynamic intralogistics," in *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, 2021, pp. 1–6.

[4] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich, "The design of stretch: A compact, lightweight mobile manipulator for indoor human environments," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2022, p. 3150–3157. [Online]. Available: https://doi.org/10.1109/ICRA46639.2022.9811922

[5] Telexistence, "Ghost," https://tx-inc.com/en/technology/, online; accessed 13-Apr-2026.

[6] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 1, p. 1334–1373, Jan. 2016.

[7] K. Kawaharazuka, J. Oh, J. Yamada, I. Posner, and Y. Zhu, "Vision-language-action models for robotics: A review towards real-world applications," *IEEE Access*, vol. 13, pp. 162 467–162 504, 2025.

[8] M. T. Shahria, M. S. H. Sunny, M. I. I. Zarif, J. Ghommam, S. I. Ahamed, and M. H. Rahman, "A comprehensive review of vision-based robotic applications: Current state, components, approaches, barriers, and potential solutions," *Robotics*, vol. 11, no. 6, 2022. [Online]. Available: https://www.mdpi.com/2218-6581/11/6/139

[9] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[10] C. Pan, B. Okorn, H. Zhang, B. Eisner, and D. Held, "Tax-pose: Task-specific cross-pose estimation for robot manipulation," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 1783–1792. [Online]. Available: https://proceedings.mlr.press/v205/pan23a.html

[11] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," *6th Annual Conference on Robot Learning (CoRL)*, 2022.

[12] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[13] J. Lu, W. Xia, Y. Wu, Z. Lu, and D. Hu, "When would vision-proprioception policies fail in robotic manipulation?" 2026. [Online]. Available: https://arxiv.org/abs/2602.12032

[14] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6382–6393.

[15] H. Duan, J. Dao, K. Green, T. Apgar, A. Fern, and J. Hurst, "Learning task space actions for bipedal locomotion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1276–1282.

[16] J. Shi, Z. Zhao, T. Wang, I. Pedroza, A. Luo, J. Wang, J. Ma, and D. Jayaraman, "Zeromimic: Distilling robotic manipulation skills from web videos," in *International Conference on Robotics and Automation (ICRA)*, 2025.

[17] J. Zhao, W. Lu, D. Zhang, Y. Liu, Y. Liang, T. Zhang, Y. Cao, J. Xie, Y. Hu, S. Wang, J. Guo, D. Wang, and Y. Gao, "Do you need proprioceptive states in visuomotor policies?" 2025. [Online]. Available: https://arxiv.org/abs/2509.18644

[18] R. Watanabe, M. Alvarez, P. Ferreiro, P. Savkin, and G. Sano, "Ftact: Force torque aware action chunking transformer for pick-and-reorient bottle task," 2025. [Online]. Available: https://arxiv.org/abs/2509.23112

[19] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2023.

APPENDIX

A. Environment

In Figure 6, we show the environment in which the data was collected and in which the evaluation takes place. The environment is made up of multiple cabinets, each with multiple levels. Out-of-distribution is defined as cabinet and level combinations unseen in the dataset.

Fig. 6. Stitched pictures of the test environment. Multiple shelves spread across multiple cabinets offering a variety of initial conditions for the X and Z rails.

B. Dataset distribution

In Figure 7, we show 5 random trajectories represented in episode-wise relative coordinates. We can see that the Z-axis has some variance intra-episode, while the X-axis stays almost fixed for the duration of the whole episode, as the scale of the X-axis shows (1×10^{-5}). The episode-wise relative representation collapses the starting conditions, allowing the model to learn more efficiently from all episodes.

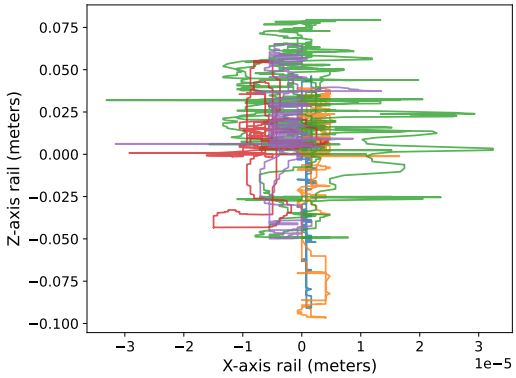

Fig. 7. X and Z values for 5 random episodes from the training dataset, represented in episode-wise relative values.

C. Videos

We provide two videos, one overview of this work: Google Drive link, and one $\times 1.5$ accelerated video of the Eps/Eps configuration solving the task: Google Drive link.