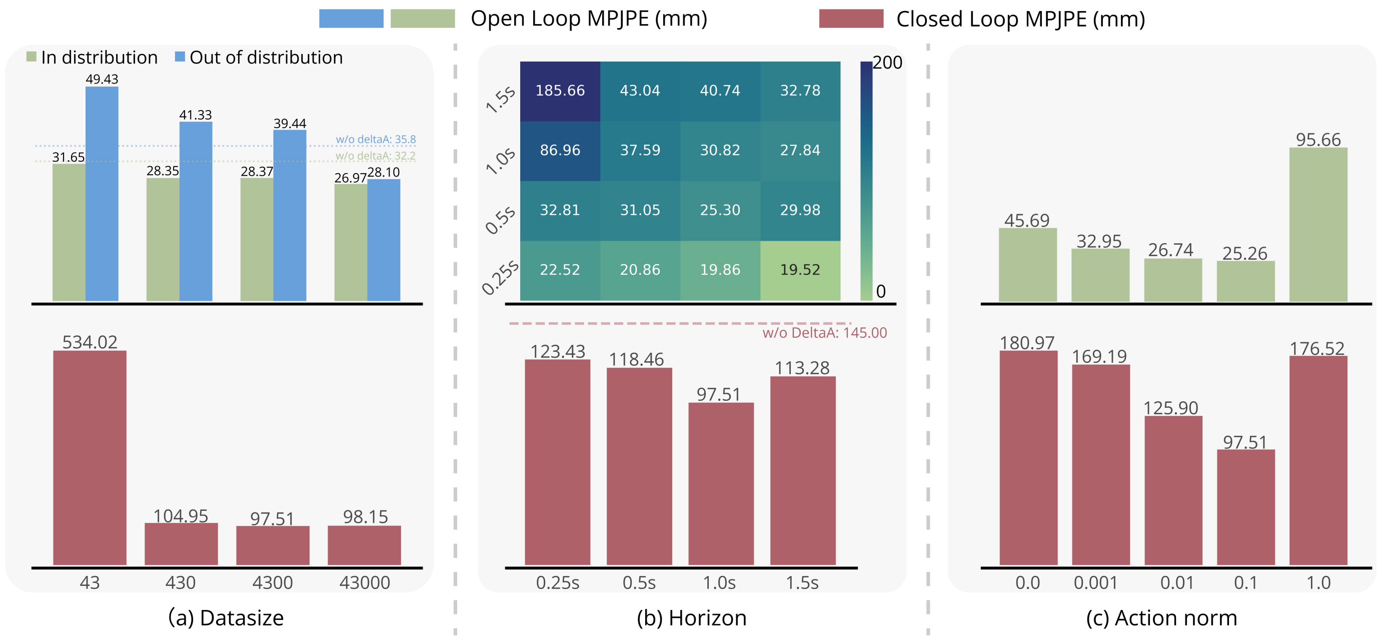

Fig. 10. Analysis of dataset size, training horizon, and action norm on the performance of π^Δ. (a) **Dataset Size**: Mean Per Joint Position Error (MPJPE) is evaluated for both in-distribution (green) and out-of-distribution (blue) scenarios. Increasing dataset size leads to enhanced generalization, evidenced by decreasing errors in out-of-distribution evaluations. Closed-loop MPJPE (red bars) also shows improvement with larger datasets. (b) **Training Horizon**: Open-loop MPJPE (heatmap) improves across evaluation points as training horizons increase, achieving the lowest error at 1.5s. However, closed-loop MPJPE (red bars) shows a sweet spot at a training horizon of 1.0s, beyond which no further improvements are observed. The red dashed line represents the pretrained baseline without π^Δ fine-tuning. (c) **Action Norm**: The action norm weight significantly influences performance. Both open-loop and closed-loop MPJPE decrease as the weight increases up to 0.1, achieving the lowest error. However, further increases in the action norm weight result in degradation of open-loop performance, highlighting the trade-off between action smoothness and policy flexibility.

essential principles for effectively training a high-performing delta action model.

a) Dataset Size: We analyze the impact of dataset size on the training and generalization of π^Δ. Simulation data is collected in Isaac Sim, and π^Δ is trained in Isaac Gym. Open-loop performance is assessed on both in-distribution (training) and out-of-distribution (unseen) trajectories, while closed-loop performance is evaluated using the fine-tuned policy in Isaac Sim. As shown in Figure 10 (a), increasing the dataset size improves π^Δ's generalization, evidenced by reduced errors in out-of-distribution evaluations. However, the improvement in closed-loop performance saturates, with a marginal decrease of only 0.65% when scaling from 4300 to 43000 samples, suggesting limited additional benefit from larger datasets.

b) Training Horizon: The rollout horizon plays a crucial role in learning π^Δ. As shown in Figure 10 (b), longer training horizons generally improve open-loop performance, with a horizon of 1.5s achieving the lowest errors across evaluation points at 0.25s, 0.5s, and 1.0s. However, this trend does not consistently extend to closed-loop performance. The best closed-loop results are observed at a training horizon of 1.0s, indicating that excessively long horizons do not provide additional benefits for fine-tuned policy.

c) Action Norm Weight: Training π^Δ incorporates an action norm reward to balance dynamics alignment and minimal correction. As illustrated in Figure 10 (c), both open-loop and closed-loop errors decrease as the action norm weight increases, reaching the lowest error at a weight of 0.1. However, further increasing the action norm weight causes open-loop errors to rise, likely due to the minimal action norm reward dominates in the delta action RL training. This highlights the importance of carefully tuning the action norm weight to achieve optimal performance.

B. Different Usage of Delta Action Model

To answer **Q5** (How to best use the delta action model of ASAP?), we compare multiple strategies: fixed-point iteration, gradient-based optimization, and reinforcement learning (RL). Given a learned delta policy π^Δ such that:

$$f^{\text{sim}}(s, a + \pi^\Delta(s,a)) \approx f^{\text{real}}(s,a),$$

and a nominal policy $\hat{\pi}(s)$ that performs well in simulation, the goal is to fine-tune $\hat{\pi}(s)$ for real-world deployment.

A simple approach is one-step dynamics matching, which leads to the relationship:

$$\pi(s) = \hat{\pi}(s) - \pi^\Delta(s, \pi(s)).$$

We consider two RL-free methods: fixed-point iteration and gradient-based optimization. Fixed-point iteration refines $\hat{\pi}(s)$ iteratively, while gradient-based optimization minimizes a loss function to achieve a better estimate. These methods are compared against RL fine-tuning, which adapts $\hat{\pi}(s)$ using reinforcement learning in simulation. The detailed derivation of these two baselines is summarized in Section VIII-D.