Appendix

A. Additional Experiments

In this section we show the performance for Augmented World Models with different training ranges for the DAS augmentation (z train in Table 4). We train with adaptive context on the HalfCheetah mixed dataset, and present the results in Fig. 13. As we see, $[0.75, 1.25]$ and $[0.5, 1.5]$ perform the best. Based on this, we use $[0.5, 1.5]$ for our experiments as we believe this helps us sample a wider set of dynamics, helping us generalize better across all environments and data sets.

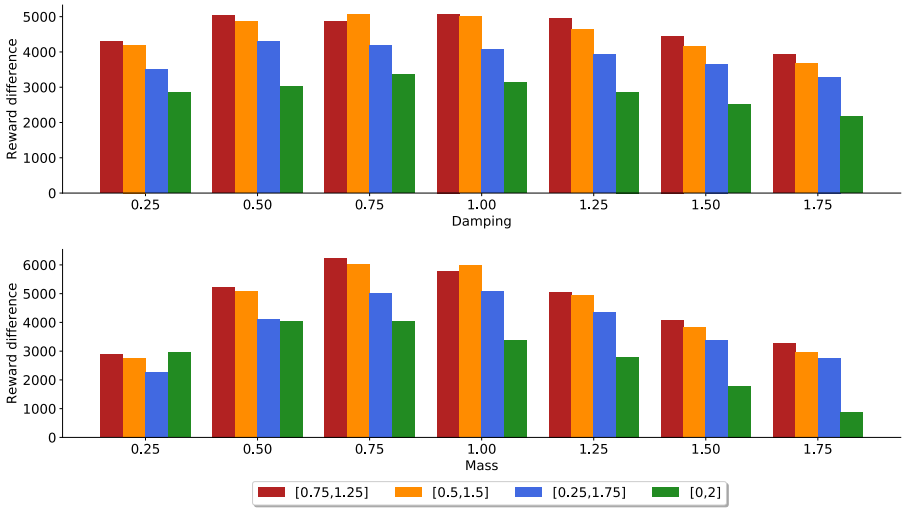

Figure 13. Performance for Augmented World Models with the DAS augmentation. Each plot shows different values for a and b, the ranges for the sampled noise.

B. Implementation Details

B.1. Hyperparameters

Our algorithm is based on MOPO (Yu et al., 2020) with values for the rollout length h and penalty coefficient λ shown in Table 3.

Table 3. Hyperparameters used in the D4RL datasets.

Dataset Type	**Environment**	**MOPO (h, λ)**
random	halfcheetah	5, 0.5
random	walker2d	1, 1
medium	halfcheetah	1, 1
medium	walker2d	1, 1 [4]
mixed	halfcheetah	5, 1
mixed	walker2d	1, 1
med-expert	halfcheetah	5, 1
med-expert	walker2d	1, 2

AugWM specific hyperparameters are listed in Table 4. For each evaluation rollout, we clear the buffer of stored true modified environment transitions to measure zero-shot performance. We adapt using the context after a set number of steps, k, in the environment to train the linear model. The two ranges used for the context z during training and test time are different. At test time, the estimated context is clipped to remain within the given bounds.

[4] We follow the original MOPO hyperparameters for all datasets except for walker2d-medium where we found (1, 1) worked better for both MOPO and our method than (5, 5).

Table 4. AugWM Hyperparameters

Parameter	Value
evaluation rollouts	5
MOPO offline epochs	400
AugWM offline epochs	900
k, steps for adaptation	300
z train range	[0.5, 1.5]
z test range	[0.93, 1.07]

B.2. D4RL dataset

We evaluate our method on D4RL (Fu et al., 2021) datasets based on the MuJoCo continuous control tasks (halfcheetah and walker2d). The four dataset types we evaluate on are:

- **random:** roll out a randomly initialized policy for 1M steps.
- **medium:** partially train a policy using SAC, then roll it out for 1M steps.
- **mixed:** train a policy using SAC until a certain (environment-specific) performance threshold is reached, and take the replay buffer as the batch.
- **medium-expert:** combine 1M samples of rollouts from a fully-trained policy with another 1M samples of rollouts from a partially trained policy or a random policy.

This gives us a total of 8 experiments.

B.3. Ant Environment

For the Ant experiments, we follow the Ant Changed Direction approach in MOPO (Yu et al., 2020). Since this offline dataset is not provided in the authors' code, nor is it in the standard D4RL library (Fu et al., 2021), we were required to generate our own offline Ant dataset. Since the authors' did not outline certain details in their experiment, we found the following was required to match their performance with our codebase: 1) Training our SAC policy for 1×10^6 timesteps in the Ant environment provided by the authors' code in (Yu et al., 2020); 2) relabelling each reward in the buffer using the new direction, without the living reward; 3) training a world model over this offline dataset; 4) training a policy in the world model, adding in living reward post-hoc; 5) evaluating the policy with the living reward.

B.4. HalfCheetah Modified Agent

We use the modified HalfCheetah environments from (Henderson et al., 2017). In each setting one body part of the agent is changed, from following set: {Foot, Leg, Thigh, Torso, Head}. The body part can either be "Big" or "Small", where Big bodyparts involve scaling the mass and width of the limb by 1.25 and Small bodyparts are scaled by 0.75. In Table 2 we show the mean over each of these five body parts, for agents trained on each of the D4RL datasets, repeated for five seeds.