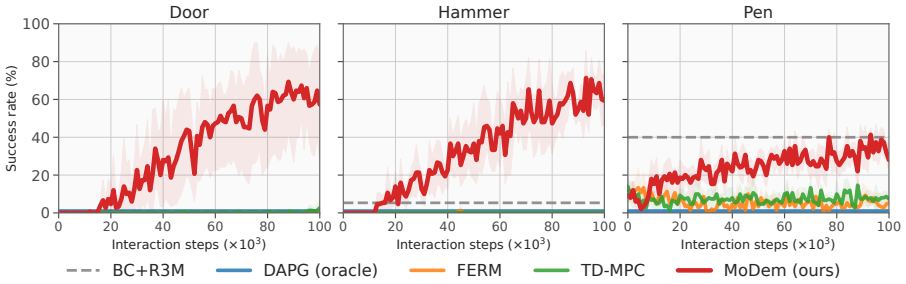

Figure 9. **Adroit.** Success rate across each of the three sparse-reward Adroit dexterous manipulation tasks. Tasks are visualized in Figure 13. Mean of 5 seeds; shaded area is 95% CIs.

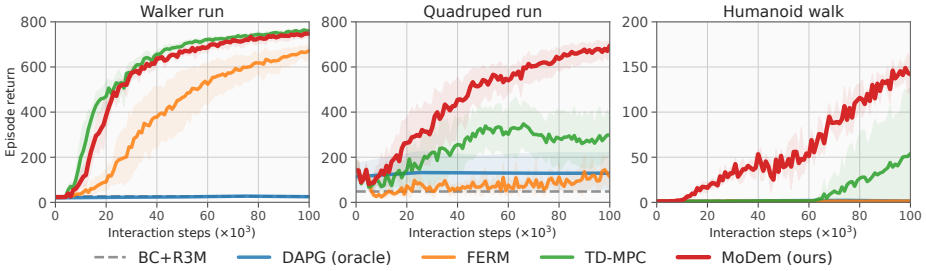

Figure 10. **DMControl.** Episode return across each of the three DMControl locomotion tasks. Quadruped Run and Humanoid Walk are visualized in Figure 13. See Tassa et al. (2018) for task details. Mean of 5 seeds; shaded area is 95% CIs.

A Additional Results

Aggregate results for each of the three domains considered are shown in Figure 4. We additionally provide all individual task results for Adroit tasks in Figure 9, for Meta-World in Figure 5, and for DMControl in Figure 10. Note that Adroit and Meta-World tasks use sparse rewards, whereas DMControl tasks use dense rewards. We also provide additional comparisons to FERM (model-free method that uses demonstrations) and a simpler instantiation of our framework that simply adds demonstrations to TD-MPC (model-based method) across all three domains; see Figure 11. We emphasize that the TD-MPC with demonstrations result is equivalent to the *None* ablation in Figure 6. We find that both aspects of our framework (model learning, and leveraging demonstrations via each of our three phases) are crucial to the performance of MoDem, both in sparse (Adroit, Meta-World) and dense (DMControl) reward domains.

B The Exploration Bottleneck vs. Algorithmic Properties

We compare three model-based methods, Dreamer-V2 (Hafner et al., 2020), MWM (Seo et al., 2022), and TD-MPC (Hansen et al., 2022) on two benchmarks, Meta-World (Yu et al., 2019) and DMControl (Tassa et al., 2018), following the experimental setups of Seo et al. (2022) and Hafner et al. (2020) for the two benchmarks, respectively. All results except for TD-MPC (Meta-World) are obtained from the respective papers and/or by correspondence with the authors; we ran TD-MPC in Meta-World ourselves by closely following the experimental setup of Seo et al. (2022) for an apples-to-apples comparison. Results are shown in Figure 12. Note that we only visualize success rates up to the 100K step mark since we are interested in the low-data regime, and that the experimental setup of Seo et al. (2022) uses shaped rewards (as opposed to our main results that use sparse rewards). We observe that all three methods perform strikingly similar across Meta-World tasks, which suggests that they are all bottlenecked by exploration rather than their individual algorithmic properties. We can therefore expect other model-based methods (besides our algorithm of choice, TD-MPC) to benefit equally well from MoDem. However, we pick TD-MPC as our backbone model and learning algorithm due to its simplicity and generally strong empirical performance as evidenced by the DMControl results shown in Figure 12 *(right)*.

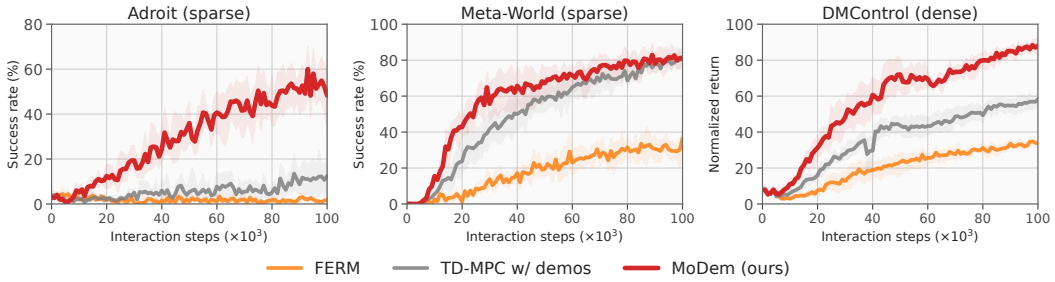

Figure 11. **Ours vs. appending demonstrations to buffer.** Success rate and episode return as a function of interaction steps on all **21** tasks across each of the three domains that we consider (Adroit, Meta-World, DMControl). Mean of 5 seeds; shaded area indicates 95% CIs. We find that both *(i)* using a model-based method, and *(ii)* leveraging demonstrations via our three-phase framework vs. simply appending demonstrations to the interaction buffer is crucial to the performance of MoDem.

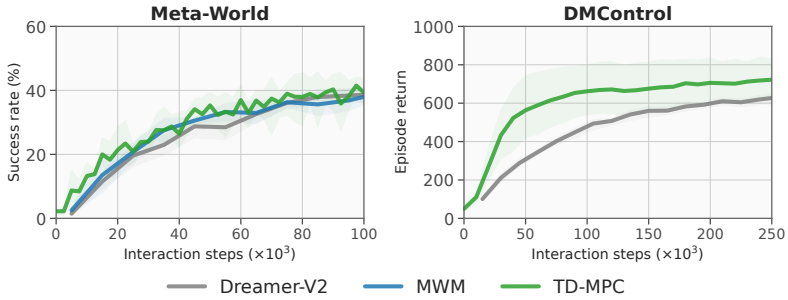

Figure 12. **Comparison of model-based methods:** Dreamer-V2 (Hafner et al., 2020), MWM (Seo et al., 2022), and TD-MPC (Hansen et al., 2022) in a limited data setting on the Meta-World (21 tasks) and DMControl (12 tasks) benchmarks. Results are obtained from Hafner et al. (2020); Seo et al. (2022); Hansen et al. (2022), except TD-MPC results for Meta-World which are run by following the experimental setup of Seo et al. (2022) for an apples-to-apples comparison. Mean of 3 seeds; shaded area is 95% CIs.

C Wall-time

While we are primarily concerned with sample-efficiency (*i.e.*, number of environment interactions required to learn a given task), we here break down the computational cost of each phase of our framework. Wall-times are shown in Table 3. We emphasize that our framework *adds no significant overhead* to phase 2 (seeding) and 3 (interactive learning), *i.e.*, running our baseline TD-MPC takes equally much time for those two phases; the *only* overhead introduced by our framework is the 5 minute BC pretraining of phase 1. Lastly, we remark that wall-time can be reduced significantly by resizing image observations to a smaller resolution for applications that are sensitive to computational cost.

Table 3. **Wall-time** for each of the three phases of MoDem.

Phase	Duration
1	5m
2	34m
3	6h3m

D Extended Experimental Setup

We evaluate methods extensively across three domains: Adroit (Rajeswaran et al., 2018), Meta-World (Yu et al., 2019), and DMControl (Tassa et al., 2018). See Figure 13 for task visualizations. In this section, we provide further details on our experimental setup for each domain.

D.1 Adroit

We consider three tasks from Adroit: Door, Hammer, Pen. Our experimental setup for Adroit closely follows Nair et al. (2022); we use 224×224 RGB frames and proprioceptive information as input,

Table 4. **Meta-World tasks.** We select 15 tasks from Meta-World based on task difficulty following the categorization of Seo et al. (2022). We experiment with all tasks from the *medium*, *hard*, and *very hard* categories that we are able to solve using MPC with a ground-truth model and a computational budget of 12 hours per demonstration. Note that the majority of Meta-World tasks are categorized as *easy*.

Difficulty	Tasks
`easy`	–
`medium`	Basketball, Box Close, Coffee Push, Peg Insert Side, Push Wall, Soccer, Sweep, Sweep Into
`hard`	Assembly, Hand Insert, Pick Place, Push
`very hard`	Stick Pull, Stick Push, Pick Place Wall

adopt their proposed `view_1` viewpoint in all three tasks, and use episode lengths of 200 for Door, 250 for Hammer, and 100 for Pen. We use an action repeat of 2 for all tasks and methods, which we find to improve sample-efficiency slightly across the board. Our evaluation constrains the sample budget to 5 demonstrations and 100K interaction steps (equivalent to 200K environment steps), whereas prior work commonly use 25-100 demonstrations (Parisi et al., 2022; Nair et al., 2022) and/or 4M environment steps (Rajeswaran et al., 2018; Shah & Kumar, 2021; Wang et al., 2022). To construct a sparse reward signal for the Adroit tasks, we provide a per-step reward of 1 when the task is solved and 0 otherwise. For Pen we use the same success criterion as in Rajeswaran et al. (2018); for Door and Hammer we relax the success criteria to the second-to-last reward stage since we find that less than 5 of the human demonstrations achieve success within the given episode length using the stricter success criteria. We use these success criteria across all methods for a fair comparison.

D.2 META-WORLD

We consider a total of 15 tasks from Meta-World. Tasks are selected based on their difficulty according to Seo et al. (2022), which categorize tasks into *easy*, *medium*, *hard*, and *very hard* categories; we discard *easy* tasks and select all tasks from the remaining 3 categories that we are able to generate demonstrations for using MPC with a ground-truth model and a computational budget of 12 hours per demonstration. This procedure yields the task set shown in Table 4. We follow the experimental setup of Seo et al. (2022) and use the same camera across all tasks: a modified `corner_2` camera where the position is adjusted with `env.model.cam pos[2] = [0.75, 0.075, 0.7]` as in prior work. We adopt the same action repeat (2) in all tasks, and use an episode length of 200 as we find that all of our considered tasks are solved within this time frame. Unlike Seo et al. (2022) that uses only RGB frames as input, we also provide proprioceptive state information (end-effector position and gripper openness) since it is readily available and requires minimal architectural changes. To construct a sparse reward signal for the Meta-World tasks, we provide a per-step reward of 1 when the task is solved according to the success criteria of Yu et al. (2019) and 0 otherwise. For completeness, we also provide an apples-to-apples comparison to Dreamer-V2 (Hafner et al., 2020) and MWM (Seo et al., 2022) by evaluating TD-MPC following their exact experimental setup; results are shown in Figure 12. We observe that all three methods perform equally well in Meta-World in a low-data setting with shaped rewards and image observations, which suggests that they are bottlenecked by exploration rather than algorithmic innovations – even when trained using shaped rewards.

D.3 DMCONTROL

We consider a total of 3 locomotion tasks from DMControl: Walker Run, Quadruped Run, Humanoid Walk. We select tasks based on diversity in embodiments and task difficulty: Walker Run and Quadruped Run are categorized as *medium* difficulty tasks, and Humanoid Walk as *hard* difficulty according to Yarats et al. (2021a). We follow the experimental setup of Hansen et al. (2022) for DMControl experiments and adopt both camera settings, hyperparameters, and their action repeat of 2 in all tasks. To be consistent across all three domains, observations include 224×224 RGB frames as well as proprioceptive state features provided by DMControl. Since rewards are only a function of the proprioceptive state in locomotion tasks, we evaluate DMControl tasks using the default, shaped rewards proposed by Tassa et al. (2018). We observe that TD-MPC generally performs slightly better than Dreamer-V2 on DMControl in the low data regime – see Figure 12 for a comparison.

Table 5. **MoDem hyperparameters.** We list all relevant hyperparameters for our proposed method below. Highlighted rows are unique to MoDem, whereas the remainder are inherited from TD-MPC but included for completeness.

Hyperparameter	Value
Discount factor (γ)	0.99
Image resolution	224×224
Frame stack	2
Data augmentation	± 10 pixel image shifts
Action repeat	2
Seed steps	$5,000$
Pretraining objective	Behavior cloning
Seeding policy	Behavior cloning
Number of demos	5
Demo sampling ratio	$75\% \rightarrow 25\%$ (100K steps)
Replay buffer size	Unlimited
Sampling technique	PER ($\alpha = 0.6, \beta = 0.4$)
Planning horizon (H)	5
Initial parameters (μ^0, σ^0)	$(0, 2)$
Population size	512
Elite fraction	64
Iterations	8 (Humanoid, Adroit) 4 (Meta-World) 6 (otherwise)
Policy fraction	5%
Number of particles	1
Momentum coefficient	0.1
Temperature (τ)	0.5
MLP hidden size	512
MLP activation	ELU
Latent dimension	100 (Humanoid) 50 (otherwise)
Learning rate	3e-4
Optimizer (θ)	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)
Temporal coefficient (λ)	0.5
Reward loss coefficient (c_1)	0.5
Value loss coefficient (c_2)	0.1
Consistency loss coefficient (c_3)	2
Exploration schedule (ϵ)	$0.1 \rightarrow 0.05$ (25k steps)
Batch size	256
Momentum coefficient (ζ)	0.99
Steps per gradient update	1
$\bar{\theta}$ update frequency	2

E Implementation Details

Environment and hyperparameters. Human demonstrations for Adroit are sourced from Rajeswaran et al. (2018) which recorded them via teleoperation. In lieu of human demonstrations for Meta-World and DMControl, we collect demonstrations for those tasks using MPC with a ground-truth model. We follow the experimental setup of Nair et al. (2022) for Adroit, Seo et al. (2022) for Meta-World, and Yarats et al. (2021a) for DMControl when applicable, but choose to use a unified multi-modal observation space across all domains. Observations are a stack of the two most recent 224×224 RGB images from a third-person camera, and also include proprioceptive information (Adroit: finger joint positions, Meta-World: end-effector position and gripper openness, DMControl: state features) as it can be assumed readily available even in real-world robotics applications. Demonstrations are of the same length as episodes during interaction and include observations, actions, and rewards for each step. We consider only sparse reward variants of Adroit and Meta-World tasks since dense rewards are typically impractical to obtain for real-world manipulation tasks, and consider dense rewards in DMControl locomotion tasks where reward is only a function of the robot state. We use an action repeat of 2 in all tasks (*i.e.*, 100K interactions = 200K environment steps). Following Hansen et al. (2022) we apply image shift augmentation (Kostrikov et al., 2020) to all observations. As observations are 224×224 as opposed to 84×84 as used in prior work, we shift images by ± 10 pixels to maintain the same ratio. Table 5 lists all relevant hyperparameters. We closely follow the original hyperparameters of TD-MPC and emphasize that we use the same hyperparameters across nearly all tasks, but list them for completeness; hyperparameters specific to MoDem are highlighted.

Network architecture. We adopt the network architecture of TD-MPC but modify the encoder to accommodate high-resolution images and proprioceptive state information as input. Specifically, we modify the encoder h_θ to consist of three components: an image encoder, a proprioceptive state encoder, and a modality fusion module. We embed image and proprioceptive state into separate feature vectors, sum them element-wise, and project them into the latent representation $\mathbf{z}$ using a 2-layer MLP. Total parameter count of model and policy is 1.6M. We provide a PyTorch-like overview of our architecture below. We here denote the latent state dimension as Z, the proprioceptive state dimension as Q, and the action dimension as A for simplicity. As in Hansen et al. (2022), the Q-function is implemented using clipped double Q-learning (Fujimoto et al., 2018).

```
Total parameters: approx. 1.6M
(h):
  (image): Sequential(
    (0): Conv2d(kernel_size=(7,7), stride=2)
    (1): ReLU()
    (2): Conv2d(kernel_size=(5,5), stride=2)
    (3): ReLU()
    (4): Conv2d(kernel_size=(3,3), stride=2)
    (5): ReLU()
    (6): Conv2d(kernel_size=(3,3), stride=2)
    (7): ReLU()
    (8): Conv2d(kernel_size=(3,3), stride=2)
    (9): ReLU()
    (10): Conv2d(kernel_size=(3,3), stride=2)
    (11): ReLU()
    (12): Linear(in_features=128, out_features=Z))
  (prop_state): Sequential(
    (0): Linear(in_features=Q, out_features=256)
    (1): ELU(alpha=1.0)
    (2): Linear(in_features=256, out_features=Z))
  (fuse): Sequential(
    (0): Linear(in_features=Z, out_features=256)
    (1): ELU(alpha=1.0)
    (2): Linear(in_features=256, out_features=Z)
(d): Sequential(
  (0): Linear(in_features=Z+A, out_features=512)
  (1): ELU(alpha=1.0)
  (2): Linear(in_features=512, out_features=512)
  (3): ELU(alpha=1.0)
  (4): Linear(in_features=512, out_features=Z))
(R): Sequential(
  (0): Linear(in_features=Z+A, out_features=512)
  (1): ELU(alpha=1.0)
  (2): Linear(in_features=512, out_features=512)
  (3): ELU(alpha=1.0)
  (4): Linear(in_features=512, out_features=1))
(pi): Sequential(
  (0): Linear(in_features=Z, out_features=512)
```

```
  (1): ELU(alpha=1.0)
  (2): Linear(in_features=512, out_features=512)
  (3): ELU(alpha=1.0)
  (4): Linear(in_features=512, out_features=A))
(Q1): Sequential(
  (0): Linear(in_features=Z+A, out_features=512)
  (1): LayerNorm((512,))
  (2): Tanh()
  (3): Linear(in_features=512, out_features=512)
  (4): ELU(alpha=1.0)
  (5): Linear(in_features=512, out_features=1))
(Q2): Sequential(
  (0): Linear(in_features=Z+A, out_features=512)
  (1): LayerNorm((512,))
  (2): Tanh()
  (3): Linear(in_features=512, out_features=512)
  (4): ELU(alpha=1.0)
```

F TASK VISUALIZATIONS

We visualize demonstration trajectories in Figure 13 for 8 of the tasks that we consider. Each frame corresponds to raw 224×224 RGB image observations that our model takes as input together with proprioceptive information. Adroit human demonstrations are visualized at key time steps, whereas Meta-World and DMControl demonstrations are shown at regular intervals of 20 interaction steps starting from a (randomized) initial state.

Visualizations are shown on the following page $\downarrow$

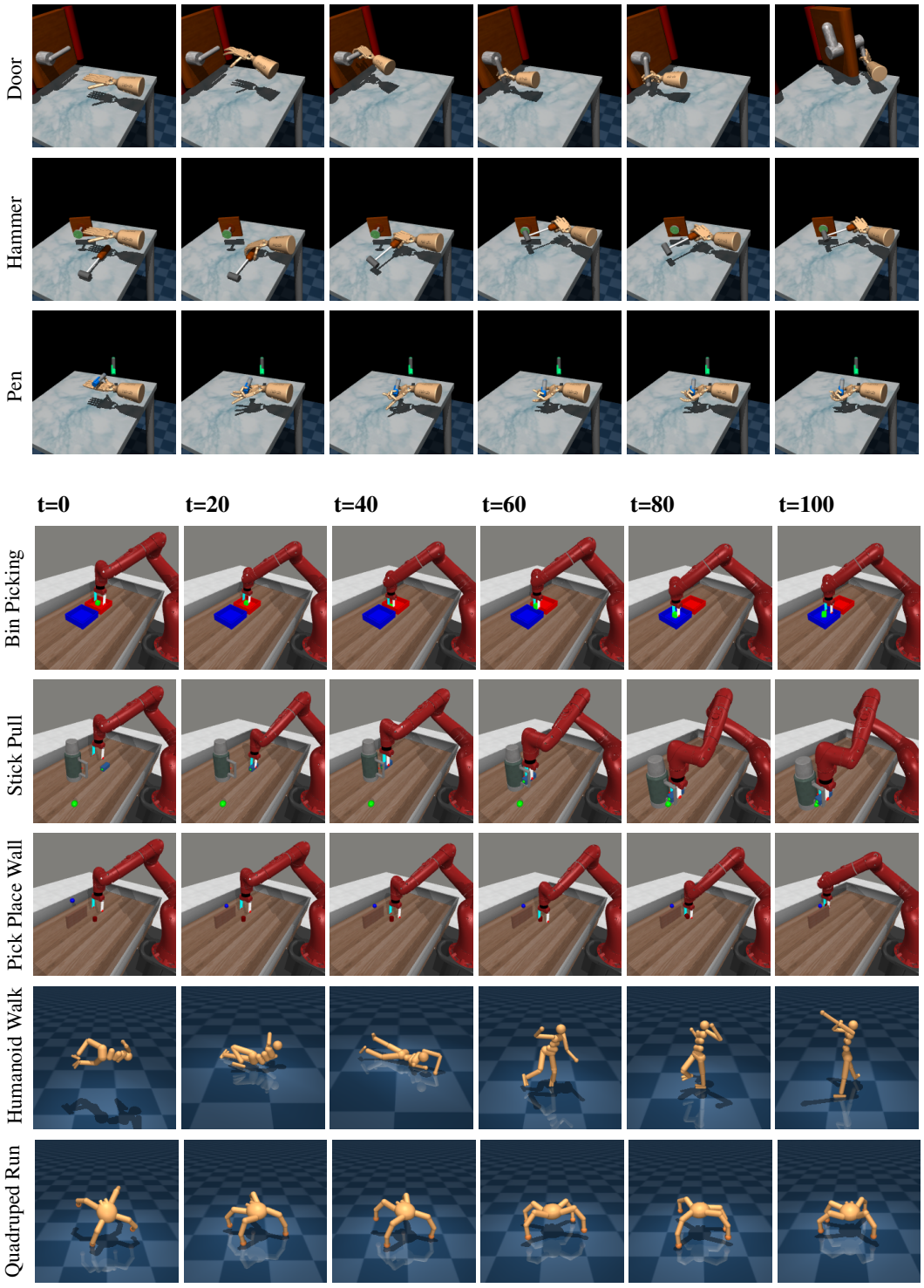

Figure 13. **Task visualizations.** We visualize demonstration trajectories for 8 of the total of 21 tasks that we consider. The raw 224×224 RGB image observations that our model takes as input together with proprioceptive information are shown; Adroit human demonstrations are visualized at key time steps, whereas Meta-World and DMControl observations are visualized at equal time intervals of 20 interaction steps, starting at a random initial state. Actual episode lengths are 100 for Adroit Pen, 200 for Adroit Door, 250 for Adroit Hammer, 200 for Meta-World tasks, and 1000 for DMControl.