A Appendix structure

This Appendix provides additional proofs and experimental results. Here we provide an overview on the organization of the section.

- Appendix B reports omitted proofs of Section 4.
- Appendix C discusses the computational improvement for DDPG with MixedNE-LD.
- Appendix D contains DDPG experimental results and hyperparameter details.
- Appendix E contains TD3 experimental results and hyperparameter details.
- Appendix F discusses the experimental setup for VPG experiments.
- Appendix G contains VPG experimental results and hyperparameter details.

The code repository (for all the experiments): `https://github.com/DaDaCheng/LIONS-RL/tree/master/Robust-Reinforcement-Learning-via-Adversarial-training-with-Langevin-Dynamics`.

B Algorithms and Omitted Proofs for Section 4

B.1 Algorithms and Hyperparameters

The pseudocode of the algorithms can be found in **Algorithm 2** (the symbol Π denotes the projection). The hyperparameter setting for experiments in Section 4 is:

- Algorithm 2 with GDA, and $\eta_t = 0.1$
- Algorithm 2 with EG, and $\eta_t = 0.1$
- Algorithm 2 with MixedNE-LD, $\eta_t = 0.1$, $\epsilon_t = 0.01$, $K_t = 50$, and $\beta = 0.5$.

We also note that we focus on the "last iterate" convergence for EG [34, 35], instead of the usual ergodic average in convex optimization literature. This is because, in practice, people almost exclusively use the last iterate.

B.2 Proof of Theorem 1

We will focus on the case $f(\theta, \omega) = \theta^2\omega^2 - \theta\omega$. Without loss of generality, we may also assume that $\omega(0) > \theta(0) > 0$; the proof of the other cases follows the same argument.

Let $(\theta(t), \omega(t))$ follow the dynamics (11) with $\theta(0) \cdot \omega(0) > 0.5$. Assume, for the moment, that both θ and ω are without constraint. Then we have

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\theta(t)^2 + \omega(t)^2\right) &= \theta\frac{\mathrm{d}\theta}{\mathrm{d}t} + \omega\frac{\mathrm{d}\omega}{\mathrm{d}t} \\ &= 2\theta^2\omega^2 - \theta\omega + (-2\theta^2\omega^2 + \theta\omega) \\ &= 0\end{aligned}$$

implying that $\theta^2(t) + \omega^2(t) = \theta^2(0) + \omega^2(0)$ for all t. Therefore $(r\cos(t+\phi_1), r\sin(t+\phi_2))$, where $(r\cos\phi_1, r\sin\phi_2) = (\theta(0), \omega(0))$, is a solution for dynamics for small enough t.

On the other hand, we have

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\theta(t)\omega(t)\right) &= \frac{\mathrm{d}\theta}{\mathrm{d}t}(t)\cdot\omega(t) + \theta(t)\cdot\frac{\mathrm{d}\omega}{\mathrm{d}t}(t) \\ &= 2\theta(t)\omega^3(t) - \omega^2(t) + \left(-2\theta^3(t)\omega(t) + \theta^2(t)\right) \\ &= \left(\theta^2(t) - \omega^2(t)\right)\left(1 - 2\theta(t)\omega(t)\right) \\ &= \left(\theta^2(t) - \omega^2(t)\right)\left(1 - 2r^2\cos(t+\phi_1)\sin(t+\phi_2)\right).\end{aligned}$$

When $t = 0$, we have $1 - 2r^2\cos(t+\phi_1)\sin(t+\phi_2) = 1 - 2\theta(0)\omega(0) < 0$. When $t = \frac{\pi}{t}$, we have

$$2r^2\cos(t+\phi_1)\sin(t+\phi_2) = 2r^2\left(\frac{\sqrt{2}}{2}\cos\phi_1 - \frac{\sqrt{2}}{2}\sin\phi_1\right)\left(\frac{\sqrt{2}}{2}\cos\phi_2 + \frac{\sqrt{2}}{2}\sin\phi_2\right)$$

Algorithm 2 Algorithms in Section 4 (MixedNE-LD / GAD / EG)

Input: step-size $\{\eta_t\}_{t=1}^T$, thermal-noise $\{\epsilon_t\}_{t=1}^T$, warmup steps $\{K_t\}_{t=1}^T$, exponential damping factor β.
for $t = 1, 2, \ldots, T-1$ **do**

MixedNE-LD:
$\bar{\omega}_t, \omega_t^{(1)} \leftarrow \omega_t$; $\bar{\theta}_t, \theta_t^{(1)} \leftarrow \theta_t$
for $k = 1, 2, \ldots, K_t$ **do**
$\xi, \xi' \sim \mathcal{N}(0, I)$
$\theta_t^{(k+1)} \leftarrow \Pi_\Theta \left(\theta_t^{(k)} + \eta_t \nabla_\theta f(\theta_t^{(k)}, \omega_t) + \epsilon_t \sqrt{2\eta_t} \xi' \right)$
$\omega_t^{(k+1)} \leftarrow \Pi_\Omega \left(\omega_t^{(k)} - \eta_t \nabla_\omega f(\theta_t, \omega_t^{(k)}) + \epsilon_t \sqrt{2\eta_t} \xi \right)$
$\bar{\omega}_t \leftarrow (1-\beta)\, \bar{\omega}_t + \beta \omega_t^{(k+1)}$
$\bar{\theta}_t \leftarrow (1-\beta)\, \bar{\theta}_t + \beta \theta_t^{(k+1)}$
end for
$\theta_{t+1} \leftarrow (1-\beta)\, \theta_t + \beta \bar{\theta}_t$
$\omega_{t+1} \leftarrow (1-\beta)\, \omega_t + \beta \bar{\omega}_t$

GAD (Gradient Ascent Descent):

$$\begin{aligned} \theta_{t+1} &\leftarrow \Pi_\Theta \left(\theta_t + \eta_t \nabla_\theta f(\theta_t, \omega_t) \right) \\ \omega_{t+1} &\leftarrow \Pi_\Omega \left(\omega_t - \eta_t \nabla_\omega f(\theta_{t+1}, \omega_t) \right) \end{aligned}$$

EG (Extra-Gradient):

$$\begin{aligned} \theta_{t+\frac{1}{2}} &\leftarrow \Pi_\Theta \left(\theta_t + \eta_t \nabla_\theta f(\theta_t, \omega_t) \right) \\ \omega_{t+\frac{1}{2}} &\leftarrow \Pi_\Omega \left(\omega_t - \eta_t \nabla_\omega f(\theta_t, \omega_t) \right) \\ \theta_{t+1} &\leftarrow \Pi_\Theta \left(\theta_t + \eta_t \nabla_\theta f(\theta_{t+\frac{1}{2}}, \omega_{t+\frac{1}{2}}) \right) \\ \omega_{t+1} &\leftarrow \Pi_\Omega \left(\omega_t - \eta_t \nabla_\theta f(\theta_{t+\frac{1}{2}}, \omega_{t+\frac{1}{2}}) \right) \end{aligned}$$

end for
Output: ω_T, θ_T.

$$\begin{aligned} &= \left(\theta(0) - \sqrt{r^2 - \theta(0)^2} \right) \left(\sqrt{r^2 - \omega(0)^2} + \omega(0) \right) \\ &= (\theta^2(0) - \omega^2(0)) < 0 \end{aligned}$$

whence $1 - 2r^2 \cos(t + \phi_1) \sin(t + \phi_2) > 0$. The intermediate value theorem then implies that there exists a $\tilde{t}$ such that $1 - 2\theta(\tilde{t})\omega(\tilde{t}) = 0$. But since $\{(\theta, \omega) \mid 2\theta\omega = 1\}$ are the stationary points of the dynamics (11), we conclude that $\frac{\mathrm{d}}{\mathrm{d}t} (\theta(t)\omega(t)) = 0$ whenever $t \geq \tilde{t}$; that is, $(\theta(t), \omega(t))$ gets trapped at the stationary point $(\theta(\tilde{t}), \omega(\tilde{t}))$. The concludes the first part the theorem when there is no boundary.

If the boundary is present, the dynamics (11) should be modified to the *projected dynamics* [36] and the proof remains the same, except that when $(\theta(t), \omega(t))$ hits the boundary, the curve needs to traverse along the boundary to decrease the norm.

We now turn to the statement for MixedNE-LD. Let (θ_1, ω_1) be initialized at any stationary point: $\theta_1 \omega_1 = 0.5$. Consider the two-step evolution of MixedNE-LD:

$$\begin{aligned} \theta_2 &= \theta_1 + \sqrt{2\eta}\xi, \\ \omega_2 &= \omega_1 + \sqrt{2\eta}\xi', \\ \theta_3 &= \theta_2 + \eta \left(2\theta_2 \omega_2^2 - \omega_2 \right) + \sqrt{2\eta}\xi'', \\ \omega_3 &= \omega_2 - \eta \left(2\theta_2^2 \omega_2 - \theta_2 \right) + \sqrt{2\eta}\xi''' \end{aligned}$$

where ξ, ξ', ξ'', and ξ''' are independent standard Gaussian. Since we initialize at a stationary point $\theta_1\omega_1 = 0.5$, we have

$$\begin{aligned} 2\theta_2\omega_2 - 1 &= 2\theta_1\omega_1 + \sqrt{2\eta}\omega_1\xi + \sqrt{2\eta}\theta_1\xi' + 2\eta\xi\xi' - 1 \\ &= \sqrt{2\eta}\omega_1\xi + \sqrt{2\eta}\theta_1\xi' + 2\eta\xi\xi'. \end{aligned} \tag{15}$$

Using the towering property of the expectation, (15), and the fact that ξ, ξ', ξ'', and ξ''' are independent standard Gaussian, we compute

$$\begin{aligned} \mathbb{E}\theta_3\omega_3 &= \mathbb{E}\left[\mathbb{E}\left[\theta_3\omega_3 \mid \theta_2, \omega_2\right]\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\left(\theta_2 + \eta\left(2\theta_2\omega_2^2 - \omega_2\right) + \sqrt{2\eta}\xi''\right)\left(\omega_2 - \eta\left(2\theta_2^2\omega_2 - \theta_2\right) + \sqrt{2\eta}\xi'''\right) \mid \theta_2, \omega_2\right]\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\left(\theta_2 + \eta\left(2\theta_2\omega_2^2 - \omega_2\right)\right)\left(\omega_2 - \eta\left(2\theta_2^2\omega_2 - \theta_2\right)\right) \mid \theta_2, \omega_2\right]\right] \\ &= \mathbb{E}\left[\left(\theta_2 + \eta\omega_2\left(2\theta_2\omega_2 - 1\right)\right)\left(\omega_2 - \eta\theta_2\left(2\theta_2\omega_2 - 1\right)\right)\right] \\ &= \mathbb{E}\left[\theta_2\omega_2 - \eta\theta_2^2\left(2\theta_2\omega_2 - 1\right) + \eta\omega_2^2\left(2\theta_2\omega_2 - 1\right) - \eta^2\theta_2\omega_2\left(2\theta_2\omega_2 - 1\right)^2\right] \\ &= \mathbb{E}\Big[\theta_1\omega_1 - \eta\left(\theta_1^2 + 2\eta\xi^2 + 2\sqrt{2\eta}\theta_1\xi - \omega_1^2 - 2\eta\xi'^2 - 2\sqrt{2\eta}\omega_1\xi'\right)\left(\sqrt{2\eta}\omega_1\xi + \sqrt{2\eta}\theta_1\xi' + 2\eta\xi\xi'\right) \\ &\qquad - 4\eta^2\left(\sqrt{2\eta}\omega_1\xi + \sqrt{2\eta}\theta_1\xi' + 2\eta\xi\xi'\right) \\ &\qquad\qquad \left(2\eta\omega_1^2\xi^2 + 2\eta\theta_1^2\xi'^2 + 4\eta^2\xi^2\xi'^2 + 2\eta\xi\xi' + 4\sqrt{2}\eta^{\frac{3}{2}}\theta_1\xi\xi'^2 + 4\sqrt{2}\eta^{\frac{3}{2}}\omega_2\xi^2\xi'\right)\Big] \\ &= \theta_1\omega_1 - 0 - 4\eta^2\left(\eta\omega_2^2 + \eta\theta_1^2 + 2\eta^2 + 4\eta^2 + 4\eta^2 + 4\eta^2\right) \\ &= \theta_1\omega_1 - 4\eta^2\left(\eta\left(\theta_1^2 + \omega_1^2\right) + 14\eta^2\right) \end{aligned}$$

which is (12).

B.3 Proof of Theorem 2

Spelling out the Newton dynamics (13), we get

$$\begin{aligned} \frac{\mathrm{d}\theta}{\mathrm{d}t}(t) &= \frac{1}{2\omega^2(t)}\left(2\theta(t)\omega^2(t) - \omega(t)\right) \\ &= \theta(t) - \frac{1}{2\omega(t)} \end{aligned}$$

and similarly $\frac{\mathrm{d}\omega}{\mathrm{d}t}(t) = -\omega(t) + \frac{1}{2\theta(t)}$. As a result, we have

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\left(\theta(t)\omega(t)\right) &= \frac{\mathrm{d}\theta}{\mathrm{d}t}(t)\cdot\omega(t) + \theta(t)\cdot\frac{\mathrm{d}\omega}{\mathrm{d}t}(t) \\ &= \theta(t)\omega(t) - \frac{1}{2} - \theta(t)\omega(t) + \frac{1}{2} \\ &= 0 \end{aligned}$$

which concludes the proof.

C One-Player DDPG with SGLD

We evaluate the robustness of one-player variants of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), i.e., we consider the NR-MDP setting with $\delta = 0$. In this case, we set $K_t = 1$ for Algorithm 3 (this choice of K_t makes the computational complexity of both algorithms equal). The corresponding results are presented in Figures 7 and 8 (also *cf.* Appendix D).

Here, we remark that Algorithm 3 with $\delta = 0$, and $K_t = 1$ is simply the standard DDPG with actor being updated by preconditioned version of SGLD. Thus we achieve robustness under different testing conditions with just a simple change in the DDPG algorithm and without additional computational cost.

C.1 Robustness of One-Player MixedNE-LD

Consider the standard (non-robust) RL objective of maximizing $J(\theta) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid \mu_\theta, \mathcal{M}_1\right]$. We can translate this non-convex problem into an infinite dimensional convex-problem by considering a distribution over deterministic policies as follows [37]:

$$\max_{p \in \mathcal{P}(\Theta)} \mathop{\mathbb{E}}_{\theta \sim p} [J(\theta)] + \lambda H(p),$$

where $H(p) = \mathop{\mathbb{E}}_{\theta \sim p} [-\log p(\theta)]$ is the entropy of the distribution p. The robust behavior of this objective (in the context of loss surface) is discussed in [38]. The optimal solution to the above problem takes the form: $p^*_\lambda(\theta) \propto \exp\left(\frac{1}{\lambda} J(\theta)\right)$. For a given λ, SGLD can be used to draw samples from $p^*_\lambda(\theta)$.

Then, the resulting algorithm is equivalent to Algorithm 3 with $\delta = 0$. Note that in our one-player DDPG experiments, we obtained significant improvement over both robust and non-robust baselines even with single inner loop iteration ($K_t = 1$). Since the Algorithm 3 is computationally demanding even though it uses a mean approximation in its inner loop, this new approximation by setting $\delta = 0$ and $K_t = 1$ is preferred in practice.

D DDPG Experiments: Algorithms, Hyperparameters, and Results

- Algorithms:
  1. MixedNE-LD: Algorithm 3
  2. Baselines: Algorithm 4 (with GAD and Extra-Adam)
- Hyperparameters:
  1. Common hyperparameters for Algorithm 3 and Algorithm 4: Table 1
  2. Exploration-related hyperparameters for Algorithm 3 and Algorithm 4 (the best performing values for every environment are presented): Tables 2 and 3
- Results:
  1. Heat maps (mass-noise) for NR-MDP setting with $\delta = 0.1$ (Figures 15 and 16)
  2. Heat maps (mass-noise) for NR-MDP setting with $\delta = 0$ (Figures 17 and 18)
  3. Heat maps (friction-noise) for NR-MDP setting with $\delta = 0.1$ (Figures 19 and 20)
  4. Heat maps (friction-noise) for NR-MDP setting with $\delta = 0$ (Figures 21 and 22)
  5. Heat maps (mass-friction) for NR-MDP setting with $\delta = 0.1$ (Figures 23 and 24)
  6. Heat maps (mass-friction) for NR-MDP setting with $\delta = 0$ (Figures 25 and 26)

E TD3 Experiments: Algorithms, Hyperparameters, and Results

- Algorithms:
  1. MixedNE-LD: Algorithm 5
  2. Baselines: Algorithm 6 (with GAD and Extra-Adam)
- Hyperparameters:
  1. Common hyperparameters for Algorithm 5 and Algorithm 6: Table 4
  2. Exploration-related hyperparameters for Algorithm 5 and Algorithm 6 (the best performing values for every environment are presented): Tables 5, 6 and 7
- Results:
  1. Mass uncertainty: Figures 5 and 9
  2. Friction uncertainty: Figures 10 and 11
  3. Comparison with SAC: Figures 12 and 13

F VPG Experiments: Setup and Evaluation

In addition to the DDPG (off-policy) experiments, we test the effectiveness of the MixedNE-LD strategy with the vanilla policy gradient (VPG) method on a toy MDP problem. In particular, we design a two-player variant of VPG [1] algorithm (*cf.* Algorithm 7) by adapting the Algorithm 1.

Setup. We compare the performance of Algorithm 7 and Algorithm 8 (with GAD and Extra-Adam) on a parametrized class of MDPs $\{\mathcal{M}_\rho = (\mathcal{S}, \mathcal{A}, T_\rho, \gamma, P_0, R) : \rho \in [0, 0.4]\}$. Here $\mathcal{S} = [-10, 10]$, $\mathcal{A} = [-1, 1]$, and $R(s) = \sin(\sqrt{1.7}s) + \cos(\sqrt{0.3}s) + 3$. The transition dynamics T_ρ is defined as follows: given the current state and action (s_t, a_t), the next state is $s_{t+1} = s_t + a_t$ with probability $1 - \rho$, and $s_{t+1} = s_t + a'$ (where $a' \sim \text{unif}([-1, 1])$) with probability ρ. We also ensure that $s_{t+1} \in [-10, 10]$.

For all the algorithms, we use a two-layer feedforward neural network structure of (16, 16, relu) for both actors (agent and adversary). The relevant hyperparameters are given in Tables 8, 9, and 10. Each algorithm is trained for 5000 steps. We run our experiments with 5 different seeds.

Evaluation. We train the algorithms with a nominal environment parameter $\rho = 0.2$, and evaluate the learned policies on a range of $\rho \in [0, 0.4]$ values. As shown in Figure 14 (*cf.* Appendix G), our Algorithm 7 outperforms the baselines Algorithm 8 (with GAD and Extra-Adam) in terms of robustness (in both two-player and one-player settings).

G VPG Experiments: Algorithms, and Hyperparameters, and Results

- Algorithms:
  1. MixedNE-LD: Algorithm 7
  2. Baselines: Algorithm 8 (with GAD and Extra-Adam)
- Hyperparameters:
  1. Common hyperparameters for Algorithm 7 and Algorithm 8: Table 8
  2. Additional hyperparameters for Algorithm 7 and Algorithm 8 (the best performing values are presented): Tables 9 and 10
- Results:
  1. NR-MDP setting with $\delta = 0.1$ (Figure 14a)
  2. NR-MDP setting with $\delta = 0$ (Figure 14b)

Table 1: Common hyperparameters for Algorithm 3 and Algorithm 4, where most of the values are chosen from [39].

Hyperparameter	Value
critic optimizer	Adam
critic learning rate	10^{-3}
target update rate τ	0.999
mini-batch size N	128
discount factor γ	0.99
damping factor β	0.9
replay buffer size	10^6
action noise parameter σ	$\{0, 0.01, 0.1, 0.2, 0.3, 0.4\}$
RMSProp parameter α	0.999
RMSProp parameter ϵ	10^{-8}
RMSProp parameter η	10^{-4}
thermal noise σ_t (Algorithm 3)	$\sigma_0 \times (1 - 5 \times 10^{-5})^t$, where $\sigma_0 \in \left\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\right\}$
warmup steps K_t (Algorithm 3)	$\min\left\{15, \lfloor(1 + 10^{-5})^t\rfloor\right\}$

Table 2: Exploration-related hyperparameters for Algorithm 3 and Algorithm 4 chosen via grid search (for NR-MDP setting with $\delta = 0.1$).

	Algorithm 3: (σ_0, σ)	Algorithm 4 (with GAD): σ	Algorithm 4 (with Extra-Adam): σ
Walker-v2	$(10^{-2}, 0.01)$	0	0.3
HalfCheetah-v2	$(10^{-2}, 0)$	0.2	0.01
Hopper-v2	$(10^{-3}, 0.2)$	0.2	0.3
Ant-v2	$(10^{-4}, 0.2)$	0.4	0.01
Swimmer-v2	$(10^{-5}, 0.4)$	0.4	0.4
Reacher-v2	$(10^{-3}, 0.2)$	0.4	0.2
Humanoid-v2	$(10^{-4}, 0.01)$	0	0.01
InvertedPendulum-v2	$(10^{-3}, 0.01)$	0.1	0.01

Table 3: Exploration-related hyperparameters for Algorithm 3 and Algorithm 4 chosen via grid search (for NR-MDP setting with $\delta = 0$).

	Algorithm 3: (σ_0, σ)	Algorithm 4 (with GAD): σ	Algorithm 4 (with Extra-Adam): σ
Walker-v2	$(10^{-2}, 0.1)$	0.01	0.2
HalfCheetah-v2	$(10^{-2}, 0.01)$	0.4	0.01
Hopper-v2	$(10^{-5}, 0.3)$	0.4	0.1
Ant-v2	$(10^{-2}, 0.4)$	0.4	0.01
Swimmer-v2	$(10^{-2}, 0.2)$	0.3	0.3
Reacher-v2	$(10^{-3}, 0.2)$	0.3	0.2
Humanoid-v2	$(10^{-2}, 0.1)$	0	0.01
InvertedPendulum-v2	$(10^{-3}, 0)$	0.01	0.01

Table 4: Common hyperparameters for Algorithm 5 and Algorithm 6, where most of the values are chosen from [27].

Hyperparameter	Value
critic optimizer	Adam
critic learning rate	3×10^{-4}
target update rate τ	0.995
mini-batch size N	128
discount factor γ	0.99
damping factor β	0.9
replay buffer size	10^6
action noise parameter σ	$\{0.005, 0.01, 0.1\}$
RMSProp parameter α	0.999
RMSProp parameter ϵ	10^{-8}
RMSProp parameter η	10^{-4}
thermal noise σ_t (Algorithm 3)	$\sigma_0 \times (1 - 5 \times 10^{-5})^t$, where $\sigma_0 \in \left\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\right\}$
warmup steps K_t (Algorithm 3)	$\min\left\{15, \lfloor(1 + 10^{-5})^t\rfloor\right\}$

Table 5: Exploration-related hyperparameters for Algorithm 5 and Algorithm 6 chosen via grid search (for NR-MDP setting with $\delta = 0.2$).

	Algorithm 5: (σ_0, σ)	Algorithm 6 (with GAD): σ	Algorithm 6 (with Extra-Adam): σ
Walker-v2	$(10^{-4}, 0.005)$	0.005	0.005
HalfCheetah-v2	$(10^{-5}, 0.005)$	0.005	0.1
Hopper-v2	$(10^{-4}, 0.1)$	0.01	0.1
Ant-v2	$(10^{-4}, 0.005)$	0.005	0.1
Swimmer-v2	$(10^{-4}, 0.005)$	0.01	0.01
Reacher-v2	$(10^{-2}, 0.1)$	0.005	0.1
Humanoid-v2	$(10^{-3}, 0.005)$	0.01	0.01
InvertedPendulum-v2	$(10^{-4}, 0.01)$	0.005	0.01

Table 6: Exploration-related hyperparameters for Algorithm 5 and Algorithm 6 chosen via grid search (for NR-MDP setting with $\delta = 0.1$).

	Algorithm 5: (σ_0, σ)	Algorithm 6 (with GAD): σ	Algorithm 6 (with Extra-Adam): σ
Walker-v2	$(10^{-3}, 0.01)$	0.01	0.01
HalfCheetah-v2	$(10^{-5}, 0.1)$	0.01	0.01
Hopper-v2	$(10^{-4}, 0.01)$	0.01	0.1
Ant-v2	$(10^{-3}, 0.01)$	0.005	0.005
Swimmer-v2	$(10^{-4}, 0.005)$	0.1	0.005
Reacher-v2	$(10^{-4}, 0.005)$	0.005	0.01
Humanoid-v2	$(10^{-5}, 0.1)$	0.01	0.01
InvertedPendulum-v2	$(10^{-3}, 0.01)$	0.01	0.01

Algorithm 3 DDPG with MixedNE-LD (pre-conditioner = RMSProp)

Hyperparameters: see Table 1
Initialize (randomly) policy parameters ω_1, θ_1, and Q-function parameter ϕ.
Initialize the target network parameters $\omega_{\text{targ}} \leftarrow \omega_1$, $\theta_{\text{targ}} \leftarrow \theta_1$, and $\phi_{\text{targ}} \leftarrow \phi$.
Initialize replay buffer $\mathcal{D}$.
Initialize $m \leftarrow \mathbf{0}$; $m' \leftarrow \mathbf{0}$.
$t \leftarrow 1$.
repeat
Observe state s, and select actions $a = \mu_{\theta_t}(s) + \xi$; $a' = \nu_{\omega_t}(s) + \xi'$, where $\xi, \xi' \sim \mathcal{N}(0, \sigma I)$
Execute the action $\bar{a} = (1-\delta)a + \delta a'$ in the environment.
Observe reward r, next state s', and done signal d to indicate whether s' is terminal.
Store $(s, \bar{a}, r, s', d)$ in replay buffer $\mathcal{D}$.
If s' is terminal, reset the environment state.
if it's time to update **then**
for however many updates **do**
$\bar{\omega}_t, \omega_t^{(1)} \leftarrow \omega_t$; $\bar{\theta}_t, \theta_t^{(1)} \leftarrow \theta_t$
for $k = 1, 2, \ldots, K_t$ **do**
Sample a random minibatch of N transitions $B = \{(s, \bar{a}, r, s', d)\}$ from $\mathcal{D}$.
Compute targets $y(r, s', d) = r + \gamma(1-d)\, Q_{\phi_{\text{targ}}}\left(s', (1-\delta)\mu_{\theta_{\text{targ}}}(s') + \delta\nu_{\omega_{\text{targ}}}(s')\right)$.
Update critic by one step of (preconditioned) gradient descent using $\nabla_\phi L(\phi)$, where

$$L(\phi) \;=\; \frac{1}{N} \sum_{(s,\bar{a},r,s',d) \in B} \left(y(r, s', d) - Q_\phi(s, \bar{a})\right)^2.$$

Compute the (agent and adversary) policy gradient estimates:

$$\widehat{\nabla_\theta J(\theta, \omega_t)} \;=\; \frac{1-\delta}{N} \sum_{s \in \mathcal{D}} \nabla_\theta \mu_\theta(s)\, \nabla_{\bar{a}} Q_\phi(s, \bar{a})\,|_{\bar{a} = (1-\delta)\mu_\theta(s) + \delta\nu_{\omega_t}(s)}$$

$$\widehat{\nabla_\omega J(\theta_t, \omega)} \;=\; \frac{\delta}{N} \sum_{s \in \mathcal{D}} \nabla_\omega \nu_\omega(s)\, \nabla_{\bar{a}} Q_\phi(s, \bar{a})\,|_{\bar{a} = (1-\delta)\mu_{\theta_t}(s) + \delta\nu_\omega(s)}.$$

$g \leftarrow \left[\widehat{\nabla_\theta J(\theta, \omega_t)}\right]_{\theta = \theta_t^{(k)}}$; $m \leftarrow \alpha m + (1-\alpha)\, g \odot g$; $C \leftarrow \text{diag}\left(\sqrt{m + \epsilon}\right)$
$\theta_t^{(k+1)} \leftarrow \theta_t^{(k)} + \eta C^{-1} g + \sqrt{2\eta}\sigma_t C^{-\frac{1}{2}} \xi$, where $\xi \sim \mathcal{N}(0, I)$
$g' \leftarrow \left[\widehat{\nabla_\omega J(\theta_t, \omega)}\right]_{\omega = \omega_t^{(k)}}$; $m' \leftarrow \alpha m' + (1-\alpha)\, g' \odot g'$; $D \leftarrow \text{diag}\left(\sqrt{m' + \epsilon}\right)$
$\omega_t^{(k+1)} \leftarrow \omega_t^{(k)} - \eta D^{-1} g' + \sqrt{2\eta}\sigma_t D^{-\frac{1}{2}} \xi'$, where $\xi' \sim \mathcal{N}(0, I)$
$\bar{\omega}_t \leftarrow (1-\beta)\,\bar{\omega}_t + \beta\omega_t^{(k+1)}$; $\bar{\theta}_t \leftarrow (1-\beta)\,\bar{\theta}_t + \beta\theta_t^{(k+1)}$
Update the target networks:

$$\begin{aligned}
\phi_{\text{targ}} &\leftarrow \tau\phi_{\text{targ}} + (1-\tau)\phi \\
\theta_{\text{targ}} &\leftarrow \tau\theta_{\text{targ}} + (1-\tau)\theta_t^{(k+1)} \\
\omega_{\text{targ}} &\leftarrow \tau\omega_{\text{targ}} + (1-\tau)\omega_t^{(k+1)}
\end{aligned}$$

end for
$\omega_{t+1} \leftarrow (1-\beta)\,\omega_t + \beta\bar{\omega}_t$; $\theta_{t+1} \leftarrow (1-\beta)\,\theta_t + \beta\bar{\theta}_t$
$t \leftarrow t + 1$.
end for
end if
until convergence
Output: ω_T, θ_T.

Algorithm 4 DDPG with GAD (pre-conditioner = RMSProp) / Extra-Adam

Hyperparameters: see Table 1
Initialize (randomly) policy parameters ω_1, θ_1, and Q-function parameter ϕ.
Initialize the target network parameters $\omega_{\text{targ}} \leftarrow \omega_1$, $\theta_{\text{targ}} \leftarrow \theta_1$, and $\phi_{\text{targ}} \leftarrow \phi$.
Initialize replay buffer $\mathcal{D}$.
Initialize $m \leftarrow \mathbf{0}$; $m' \leftarrow \mathbf{0}$.
$t \leftarrow 1$.
repeat
Observe state s, and select actions $a = \mu_{\theta_t}(s) + \xi$; $a' = \nu_{\omega_t}(s) + \xi'$, where $\xi, \xi' \sim \mathcal{N}(0, \sigma I)$
Execute the action $\bar{a} = (1-\delta)a + \delta a'$ in the environment.
Observe reward r, next state s', and done signal d to indicate whether s' is terminal.
Store $(s, \bar{a}, r, s', d)$ in replay buffer $\mathcal{D}$.
If s' is terminal, reset the environment state.
if it's time to update **then**
for however many updates **do**
Sample a random minibatch of N transitions $B = \{(s, \bar{a}, r, s', d)\}$ from $\mathcal{D}$.
Compute targets $y(r, s', d) = r + \gamma(1-d) Q_{\phi_{\text{targ}}}\left(s', (1-\delta)\mu_{\theta_{\text{targ}}}(s') + \delta\nu_{\omega_{\text{targ}}}(s')\right)$.
Update critic by one step of (preconditioned) gradient descent using $\nabla_\phi L(\phi)$, where

$$L(\phi) \;=\; \frac{1}{N} \sum_{(s,\bar{a},r,s',d) \in B} \left(y(r, s', d) - Q_\phi(s, \bar{a})\right)^2.$$

Compute the (agent and adversary) policy gradient estimates:

$$\widehat{\nabla_\theta J(\theta, \omega_t)} \;=\; \frac{1-\delta}{N} \sum_{s \in \mathcal{D}} \nabla_\theta \mu_\theta(s) \nabla_{\bar{a}} Q_\phi(s, \bar{a})\,|_{\bar{a}=(1-\delta)\mu_\theta(s)+\delta\nu_{\omega_t}(s)}$$

$$\widehat{\nabla_\omega J(\theta_t, \omega)} \;=\; \frac{\delta}{N} \sum_{s \in \mathcal{D}} \nabla_\omega \nu_\omega(s) \nabla_{\bar{a}} Q_\phi(s, \bar{a})\,|_{\bar{a}=(1-\delta)\mu_{\theta_t}(s)+\delta\nu_\omega(s)}.$$

GAD (pre-conditioner = RMSProp):
$g \leftarrow \left[\widehat{\nabla_\theta J(\theta, \omega_t)}\right]_{\theta=\theta_t}$; $m \leftarrow \alpha m + (1-\alpha) g \odot g$; $C \leftarrow \text{diag}\left(\sqrt{m + \epsilon}\right)$
$\theta_{t+1} \leftarrow \theta_t + \eta C^{-1} g$
$g' \leftarrow \left[\widehat{\nabla_\omega J(\theta_t, \omega)}\right]_{\omega=\omega_t}$; $m' \leftarrow \alpha m' + (1-\alpha) g' \odot g'$; $D \leftarrow \text{diag}\left(\sqrt{m' + \epsilon}\right)$
$\omega_{t+1} \leftarrow \omega_t - \eta D^{-1} g'$
Extra-Adam: use Algorithm 4 from [20].
Update the target networks:

$$\begin{aligned}
\phi_{\text{targ}} &\leftarrow \tau\phi_{\text{targ}} + (1-\tau)\phi \\
\theta_{\text{targ}} &\leftarrow \tau\theta_{\text{targ}} + (1-\tau)\theta_{t+1} \\
\omega_{\text{targ}} &\leftarrow \tau\omega_{\text{targ}} + (1-\tau)\omega_{t+1}
\end{aligned}$$

$t \leftarrow t + 1$.
end for
end if
until convergence
Output: ω_T, θ_T.

Algorithm 5 TD3 with MixedNE-LD (pre-conditioner = RMSProp)

Hyperparameters: see Table 4
Initialize (randomly) policy parameters ω_1, θ_1, and Q-function parameters ϕ_1, ϕ_2.
Initialize the target network parameters $\omega_{\text{targ}} \leftarrow \omega_1$, $\theta_{\text{targ}} \leftarrow \theta_1$, and $\phi_{\text{targ},1} \leftarrow \phi_1$, $\phi_{\text{targ},2} \leftarrow \phi_2$.
Initialize replay buffer $\mathcal{D}$.
Initialize $m \leftarrow \mathbf{0}$; $m' \leftarrow \mathbf{0}$.
$t \leftarrow 1$.
repeat
Observe state s, and select actions $a = \text{clip}\left(\mu_{\theta_t}(s) + \xi, a_{\text{Low}}, a_{\text{High}}\right)$; $a' = \text{clip}(\nu_{\omega_t}(s) + \xi', a_{\text{Low}}, a_{\text{High}})$, where $\xi, \xi' \sim \mathcal{N}(0, \sigma I)$
Execute the action $\bar{a} = (1-\delta)a + \delta a'$ in the environment.
Observe reward r, next state s', and done signal d to indicate whether s' is terminal.
Store $(s, \bar{a}, r, s', d)$ in replay buffer $\mathcal{D}$.
If s' is terminal, reset the environment state.
if it's time to update **then**
for however many updates **do**
$\bar{\omega}_t, \omega_t^{(1)} \leftarrow \omega_t$; $\bar{\theta}_t, \theta_t^{(1)} \leftarrow \theta_t$
for $k = 1, 2, \ldots, K_t$ **do**
Sample a random minibatch of N transitions $B = \{(s, \bar{a}, r, s', d)\}$ from $\mathcal{D}$.
Compute target actions

$$\tilde{a} = \text{clip}((1-\delta)\mu_{\theta_{\text{targ}}}(s') + \delta\nu_{\omega_{\text{targ}}}(s') + \epsilon, a_{\text{Low}}, a_{\text{High}}), \text{where } \epsilon \sim \text{clip}(\mathcal{N}(0, \sigma I), -c, c)$$

Compute targets $y(r, s', d) = r + \gamma \min_{i=1,2}(1-d) Q_{\phi_{\text{targ,i}}}(s', \tilde{a})$.
Update critic by one step of (preconditioned) gradient descent using $\nabla_\phi L(\phi)$, where

$$L(\phi) = \frac{1}{N} \sum_{(s,\bar{a},r,s',d) \in B} \left(y(r, s', d) - Q_{\phi_i}(s, \bar{a})\right)^2. \qquad \text{for } i = 1, 2$$

if t mod policy_delay = 0 **then**
Compute the (agent and adversary) policy gradient estimates:

$$\widehat{\nabla_\theta J(\theta, \omega_t)} = \frac{1-\delta}{N} \sum_{s \in \mathcal{D}} \nabla_\theta \mu_\theta(s) \nabla_{\bar{a}} Q_{\phi_1}(s, \bar{a}) |_{\bar{a}=(1-\delta)\mu_\theta(s)+\delta\nu_{\omega_t}(s)}$$

$$\widehat{\nabla_\omega J(\theta_t, \omega)} = \frac{\delta}{N} \sum_{s \in \mathcal{D}} \nabla_\omega \nu_\omega(s) \nabla_{\bar{a}} Q_{\phi_1}(s, \bar{a}) |_{\bar{a}=(1-\delta)\mu_{\theta_t}(s)+\delta\nu_\omega(s)}.$$

$g \leftarrow \left[\widehat{\nabla_\theta J(\theta, \omega_t)}\right]_{\theta=\theta_t^{(k)}}$; $m \leftarrow \alpha m + (1-\alpha) g \odot g$; $C \leftarrow \text{diag}\left(\sqrt{m + \epsilon}\right)$
$\theta_t^{(k+1)} \leftarrow \theta_t^{(k)} + \eta C^{-1} g + \sqrt{2\eta}\sigma_t C^{-\frac{1}{2}} \xi$, where $\xi \sim \mathcal{N}(0, I)$
$g' \leftarrow \left[\widehat{\nabla_\omega J(\theta_t, \omega)}\right]_{\omega=\omega_t^{(k)}}$; $m' \leftarrow \alpha m' + (1-\alpha) g' \odot g'$; $D \leftarrow \text{diag}\left(\sqrt{m' + \epsilon}\right)$
$\omega_t^{(k+1)} \leftarrow \omega_t^{(k)} - \eta D^{-1} g' + \sqrt{2\eta}\sigma_t D^{-\frac{1}{2}} \xi'$, where $\xi' \sim \mathcal{N}(0, I)$
$\bar{\omega}_t \leftarrow (1-\beta)\bar{\omega}_t + \beta\omega_t^{(k+1)}$; $\bar{\theta}_t \leftarrow (1-\beta)\bar{\theta}_t + \beta\theta_t^{(k+1)}$
Update the target networks:

$$\begin{aligned} \phi_{\text{targ,i}} &\leftarrow \tau\phi_{\text{targ,i}} + (1-\tau)\phi_i \qquad \text{for } i = 1, 2 \\ \theta_{\text{targ}} &\leftarrow \tau\theta_{\text{targ}} + (1-\tau)\theta_t^{(k+1)} \\ \omega_{\text{targ}} &\leftarrow \tau\omega_{\text{targ}} + (1-\tau)\omega_t^{(k+1)} \end{aligned}$$

end if
end for
$\omega_{t+1} \leftarrow (1-\beta)\omega_t + \beta\bar{\omega}_t$; $\theta_{t+1} \leftarrow (1-\beta)\theta_t + \beta\bar{\theta}_t$
$t \leftarrow t + 1$.
end for
end if
until convergence
Output: ω_T, θ_T.

Algorithm 6 TD3 with GAD (pre-conditioner = RMSProp) / Extra-Adam

Hyperparameters: see Table 4
Initialize (randomly) policy parameters ω_1, θ_1, and Q-function parameters ϕ_1, ϕ_2.
Initialize the target network parameters $\omega_{\text{targ}} \leftarrow \omega_1$, $\theta_{\text{targ}} \leftarrow \theta_1$, and $\phi_{\text{targ},1} \leftarrow \phi_1$, $\phi_{\text{targ},2} \leftarrow \phi_2$.
Initialize replay buffer $\mathcal{D}$.
Initialize $m \leftarrow \mathbf{0}$; $m' \leftarrow \mathbf{0}$.
$t \leftarrow 1$.
repeat
Observe state s, and select actions $a = \text{clip}\left(\mu_{\theta_t}(s) + \xi, a_{\text{Low}}, a_{\text{High}}\right)$; $a' = \text{clip}(\nu_{\omega_t}(s) + \xi', a_{\text{Low}}, a_{\text{High}})$, where $\xi, \xi' \sim \mathcal{N}(0, \sigma I)$
Execute the action $\bar{a} = (1-\delta)a + \delta a'$ in the environment.
Observe reward r, next state s', and done signal d to indicate whether s' is terminal.
Store $(s, \bar{a}, r, s', d)$ in replay buffer $\mathcal{D}$.
If s' is terminal, reset the environment state.
if it's time to update **then**
for however many updates **do**
Sample a random minibatch of N transitions $B = \{(s, \bar{a}, r, s', d)\}$ from $\mathcal{D}$.
Compute target actions

$$\tilde{a} = \text{clip}((1-\delta)\mu_{\theta_{\text{targ}}}(s') + \delta\nu_{\omega_{\text{targ}}}(s') + \epsilon, a_{\text{Low}}, a_{\text{High}}), \text{where } \epsilon \sim \text{clip}(\mathcal{N}(0, \sigma I), -c, c)$$

Compute targets $y(r, s', d) = r + \gamma \min_{i=1,2}(1-d) Q_{\phi_{\text{targ},i}}(s', \tilde{a})$.
Update critic by one step of (preconditioned) gradient descent using $\nabla_\phi L(\phi)$, where

$$L(\phi) = \frac{1}{N} \sum_{(s,\bar{a},r,s',d) \in B} \left(y(r, s', d) - Q_{\phi_i}(s, \bar{a})\right)^2. \qquad \text{for } i = 1, 2$$

if t mod policy_delay = 0 **then**
Compute the (agent and adversary) policy gradient estimates:

$$\widehat{\nabla_\theta J(\theta, \omega_t)} = \frac{1-\delta}{N} \sum_{s \in \mathcal{D}} \nabla_\theta \mu_\theta(s) \nabla_{\bar{a}} Q_{\phi_1}(s, \bar{a}) |_{\bar{a}=(1-\delta)\mu_\theta(s)+\delta\nu_{\omega_t}(s)}$$

$$\widehat{\nabla_\omega J(\theta_t, \omega)} = \frac{\delta}{N} \sum_{s \in \mathcal{D}} \nabla_\omega \nu_\omega(s) \nabla_{\bar{a}} Q_{\phi_1}(s, \bar{a}) |_{\bar{a}=(1-\delta)\mu_{\theta_t}(s)+\delta\nu_\omega(s)}.$$

GAD (pre-conditioner = RMSProp):
$g \leftarrow \left[\widehat{\nabla_\theta J(\theta, \omega_t)}\right]_{\theta=\theta_t}$; $m \leftarrow \alpha m + (1-\alpha) g \odot g$; $C \leftarrow \text{diag}\left(\sqrt{m + \epsilon}\right)$
$\theta_{t+1} \leftarrow \theta_t + \eta C^{-1} g$
$g' \leftarrow \left[\widehat{\nabla_\omega J(\theta_t, \omega)}\right]_{\omega=\omega_t}$; $m' \leftarrow \alpha m' + (1-\alpha) g' \odot g'$; $D \leftarrow \text{diag}\left(\sqrt{m' + \epsilon}\right)$
$\omega_{t+1} \leftarrow \omega_t - \eta D^{-1} g'$
Extra-Adam: use Algorithm 4 from [20].
Update the target networks:

$$\begin{aligned} \phi_{\text{targ,i}} &\leftarrow \tau\phi_{\text{targ,i}} + (1-\tau)\phi_i \qquad \text{for } i = 1, 2 \\ \theta_{\text{targ}} &\leftarrow \tau\theta_{\text{targ}} + (1-\tau)\theta_{t+1} \\ \omega_{\text{targ}} &\leftarrow \tau\omega_{\text{targ}} + (1-\tau)\omega_{t+1} \end{aligned}$$

end if
$t \leftarrow t + 1$.
end for
end if
until convergence
Output: ω_T, θ_T.

Algorithm 7 VPG with MixedNE-LD (pre-conditioner = RMSProp)

Hyperparameters: see Table 8
Initialize (randomly) policy parameters θ_0, w_0
for $k = 0, 1, 2, \dots$ **do**
  $\bar{\theta}_k, \theta_k^{(0)} \leftarrow \theta_k$; $\bar{w}_k, w_k^{(0)} \leftarrow w_k$
  for $n = 0, 1, \dots, N_k$ **do**
    Collect set of trajectories $\mathcal{D}_k^{(n)} = \{(\dots, s_t^{(\tau)}, \bar{a}_t^{(\tau)}, r_t^{(\tau)}, \dots)\}_\tau$ by running $\pi_{\theta_k^{(n)}}$, and $\pi'_{w_k^{(n)}}$ in $\mathcal{M}$, i.e., $a_t \sim \pi_{\theta_k^{(n)}}(s_t)$, $a'_t \sim \pi'_{w_k^{(n)}}(s_t)$, $\bar{a}_t = (1-\delta)a_t + \delta a'_t$, and $s_{t+1} \sim T_\rho(\cdot \mid s_t, \bar{a}_t)$.
    Estimate the policy gradient (where $G_t = \sum_{s=0}^{T} \gamma^s r_{t+s}$)

$$g = \frac{1-\delta}{\left|\mathcal{D}_k^{(n)}\right|} \sum_{\tau \in \mathcal{D}_k^{(n)}} \sum_t \gamma^t G_t^{(\tau)} \left[\nabla_\theta \log \pi_\theta(a_t^{(\tau)} \mid s_t^{(\tau)})\right]_{\theta=\theta_k^{(n)}}$$

$$g' = \frac{\delta}{\left|\mathcal{D}_k^{(n)}\right|} \sum_{\tau \in \mathcal{D}_k^{(n)}} \sum_t \gamma^t G_t^{(\tau)} \left[\nabla_w \log \pi_w(a_t'^{(\tau)} \mid s_t^{(\tau)})\right]_{w=w_k^{(n)}}$$

    $m \leftarrow \alpha m + (1-\alpha)\, g \odot g$; $C \leftarrow \operatorname{diag}\left(\sqrt{m+\epsilon}\right)$
    $\theta_k^{(n+1)} \leftarrow \theta_k^{(n)} + \eta C^{-1} g + \sqrt{2\eta}\sigma_k C^{-\frac{1}{2}} \xi$, where $\xi \sim \mathcal{N}(0, I)$
    $\bar{\theta}_k \leftarrow (1-\beta)\,\bar{\theta}_k + \beta \theta_k^{(n+1)}$
    $m' \leftarrow \alpha m' + (1-\alpha)\, g' \odot g'$; $D \leftarrow \operatorname{diag}\left(\sqrt{m'+\epsilon}\right)$
    $w_k^{(n+1)} \leftarrow w_k^{(n)} - \eta D^{-1} g + \sqrt{2\eta}\sigma_k D^{-\frac{1}{2}} \xi'$, where $\xi' \sim \mathcal{N}(0, I)$
    $\bar{w}_k \leftarrow (1-\beta)\,\bar{w}_k + \beta w_k^{(n+1)}$
  end for
  $\theta_{k+1} \leftarrow (1-\beta)\,\theta_k + \beta \bar{\theta}_k$
  $w_{k+1} \leftarrow (1-\beta)\, w_k + \beta \bar{w}_k$
end for

Algorithm 8 VPG with GAD (pre-conditioner = RMSProp) / Extra-Adam

Hyperparameters: see Table 8
Initialize (randomly) policy parameters θ_0, w_0
for $k = 0, 1, 2, \dots$ **do**
  Collect set of trajectories $\mathcal{D}_k = \{(\dots, s_t^{(\tau)}, \bar{a}_t^{(\tau)}, r_t^{(\tau)}, \dots)\}_\tau$ by running π_{θ_k}, and π'_{w_k} in $\mathcal{M}$, i.e., $a_t \sim \pi_{\theta_k}(s_t)$, $a'_t \sim \pi'_{w_k}(s_t)$, $\bar{a}_t = (1-\delta)a_t + \delta a'_t$, and $s_{t+1} \sim T_\rho(\cdot \mid s_t, \bar{a}_t)$.
  Estimate the policy gradient (where $G_t = \sum_{s=0}^{T} \gamma^s r_{t+s}$)

$$g = \frac{1-\delta}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_t \gamma^t G_t^{(\tau)} \left[\nabla_\theta \log \pi_\theta(a_t^{(\tau)} \mid s_t^{(\tau)})\right]_{\theta=\theta_k}$$

$$g' = \frac{\delta}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_t \gamma^t G_t^{(\tau)} \left[\nabla_w \log \pi'_w(a_t'^{(\tau)} \mid s_t^{(\tau)})\right]_{w=w_k}$$

  GAD (pre-conditioner = RMSProp):
  $m \leftarrow \alpha m + (1-\alpha)\, g \odot g$; $C \leftarrow \operatorname{diag}\left(\sqrt{m+\epsilon}\right)$
  $\theta_{k+1} \leftarrow \theta_k + \eta C^{-1} g$
  $m' \leftarrow \alpha m' + (1-\alpha)\, g' \odot g'$; $D \leftarrow \operatorname{diag}\left(\sqrt{m'+\epsilon}\right)$
  $w_{k+1} \leftarrow w_k - \eta D^{-1} g'$
  Extra-Adam: use Algorithm 4 from [20].
end for

Table 7: Exploration-related hyperparameters for Algorithm 5 and Algorithm 6 chosen via grid search (for NR-MDP setting with $\delta = 0$).

	Algorithm 5: (σ_0, σ)	Algorithm 6 (with GAD): σ	Algorithm 6 (with Extra-Adam): σ
Walker-v2	$(10^{-5}, 0.01)$	0.01	0.1
HalfCheetah-v2	$(10^{-5}, 0.01)$	0.01	0.001
Hopper-v2	$(10^{-4}, 0.1)$	0.1	0.005
Ant-v2	$(10^{-3}, 0.1)$	0.1	0.1
Swimmer-v2	$(10^{-5}, 0.01)$	0.01	0.005
Reacher-v2	$(10^{-4}, 0.1)$	0.1	0.1
Humanoid-v2	$(10^{-4}, 0.1)$	0.1	0.005
InvertedPendulum-v2	$(10^{-4}, 0.01)$	0.01	0.005

Table 8: Common hyperparameters for Algorithm 7 and Algorithm 8.

Hyperparameter	Value
discount factor γ	0.99
trajectory length H	500
number of trajectories per step $\lvert\mathcal{D}_k\rvert$	1
RMSProp parameter α	0.99
RMSProp parameter ϵ	10^{-8}
learning rate η	$\{10^{-3}, 10^{-4}, 10^{-5}\}$
damping factor β	0.9

Table 9: Additional hyperparameters for Algorithm 7 and Algorithm 8 chosen via grid search (for NR-MDP setting with $\delta = 0.1$)

	Algorithm 7: (σ_0, η, N_k)	Algorithm 8 (with GAD): η	Algorithm 8 (with Extra-Adam): η
$\rho = 0.2$	$(10^{-5}, 10^{-3}, 1)$	10^{-4}	10^{-4}

Table 10: Additional hyperparameters for Algorithm 7 and Algorithm 8 chosen via grid search (for NR-MDP setting with $\delta = 0$)

	Algorithm 7: (σ_0, η, N_k)	Algorithm 8 (with GAD): η	Algorithm 8 (with Extra-Adam): η
$\rho = 0.2$	$(10^{-4}, 10^{-4}, 10)$	10^{-4}	10^{-3}

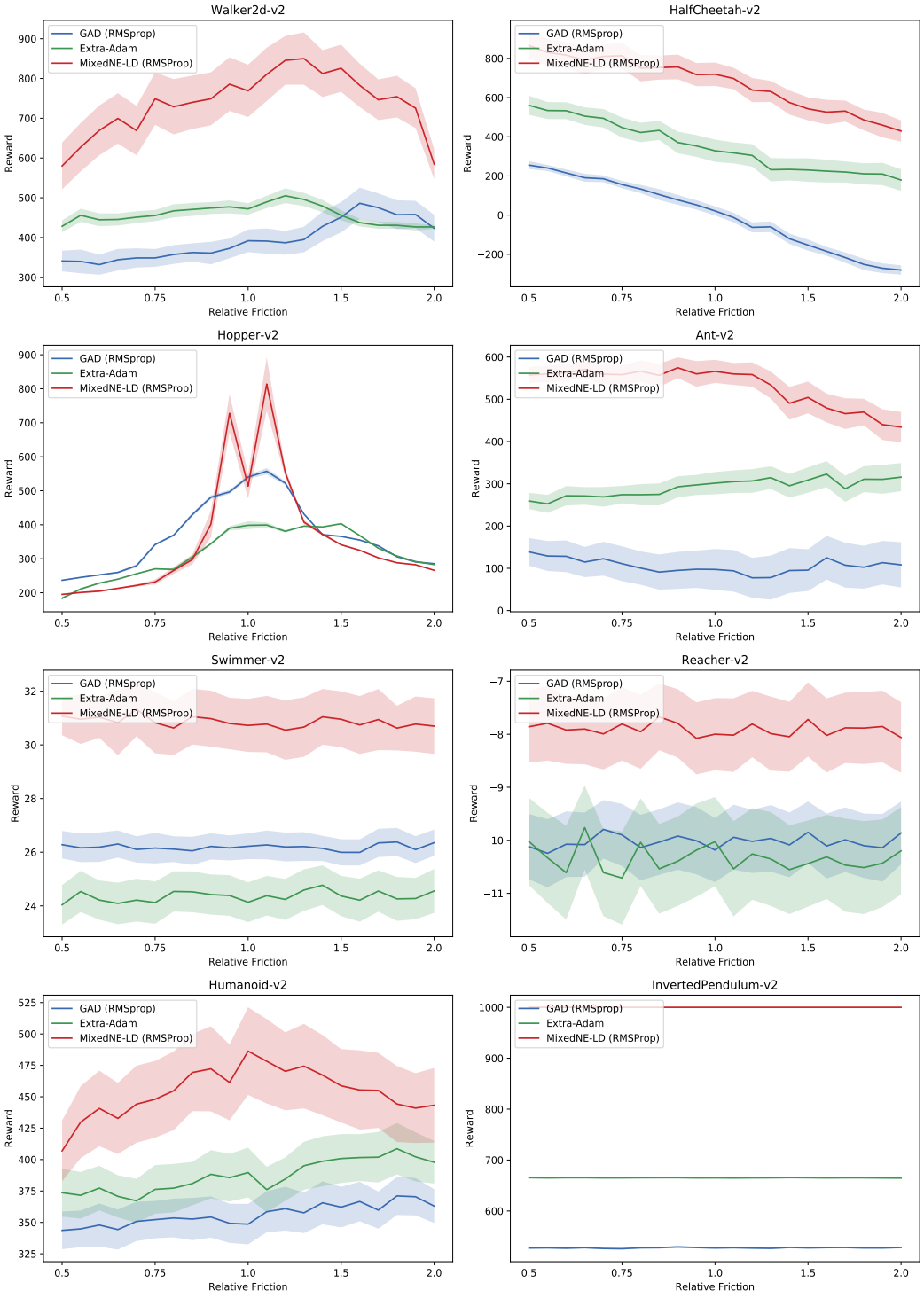


Figure 6: Average performance (over 5 seeds) of Algorithm 3 (DDPG with MixedNE-LD), and Algorithm 4 (DDPG with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training.

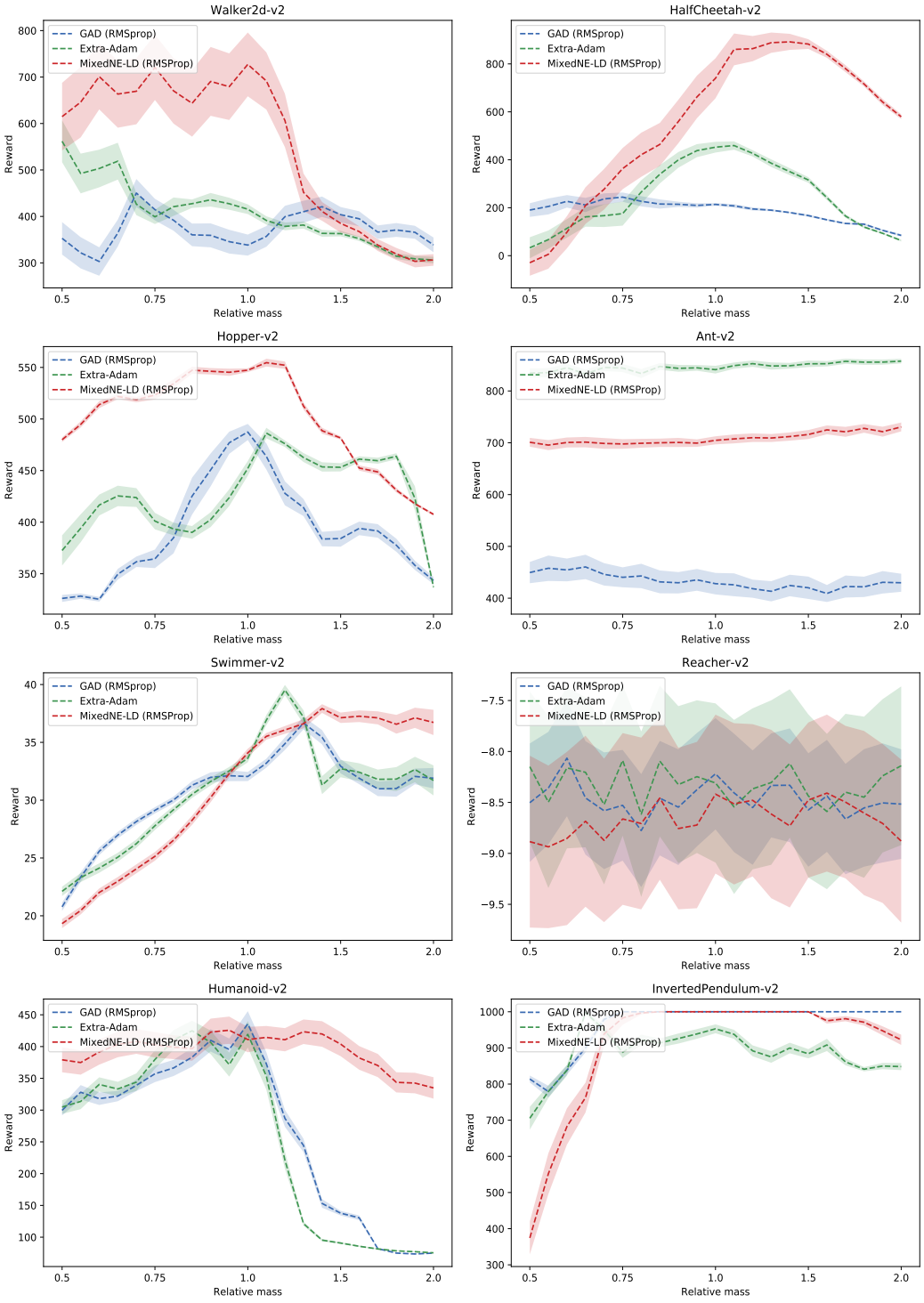


Figure 7: Average performance (over 5 seeds) of Algorithm 3 (DDPG with MixedNE-LD), and Algorithm 4 (DDPG with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training.

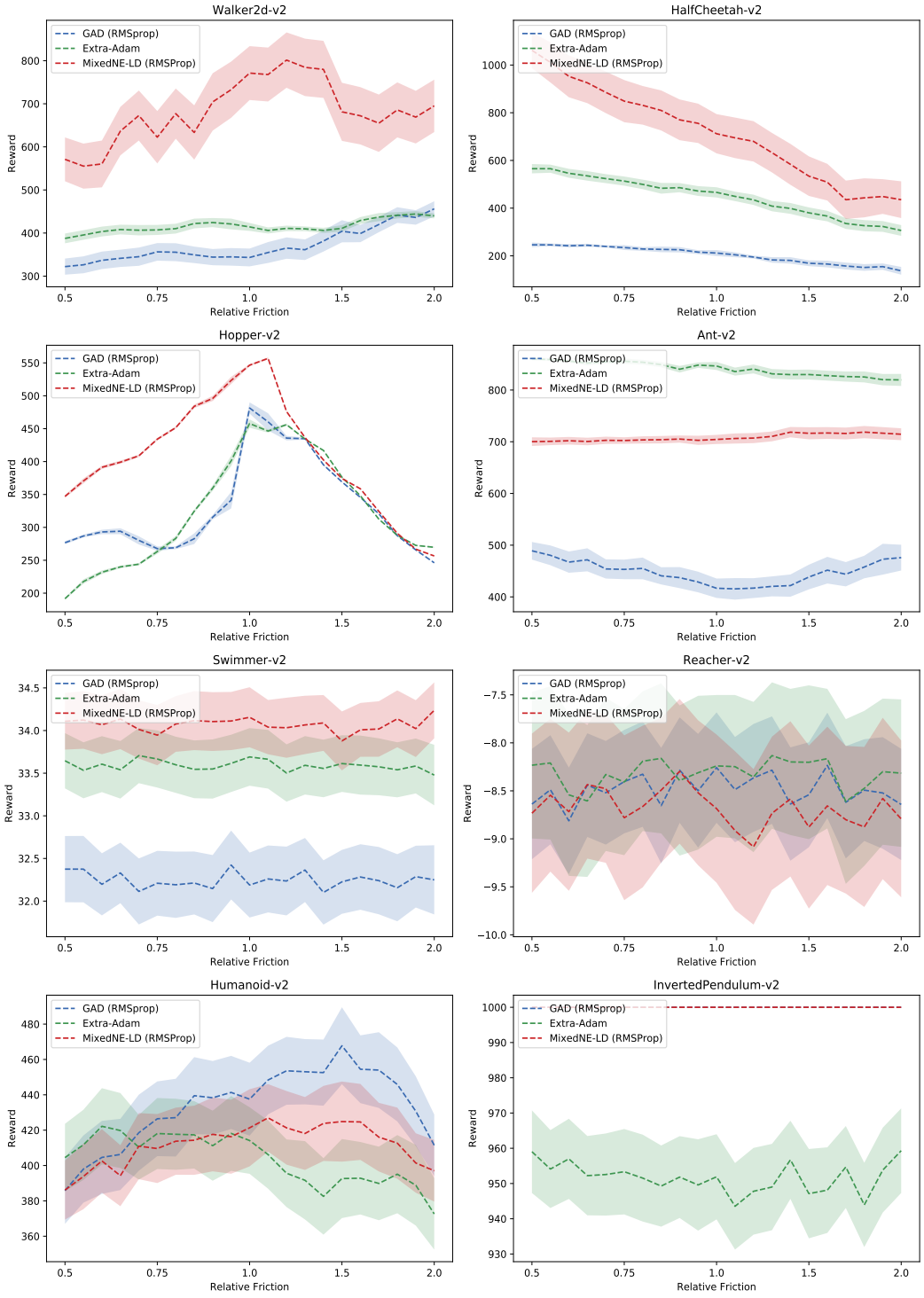


Figure 8: Average performance (over 5 seeds) of Algorithm 3 (DDPG with MixedNE-LD), and Algorithm 4 (DDPG with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training.

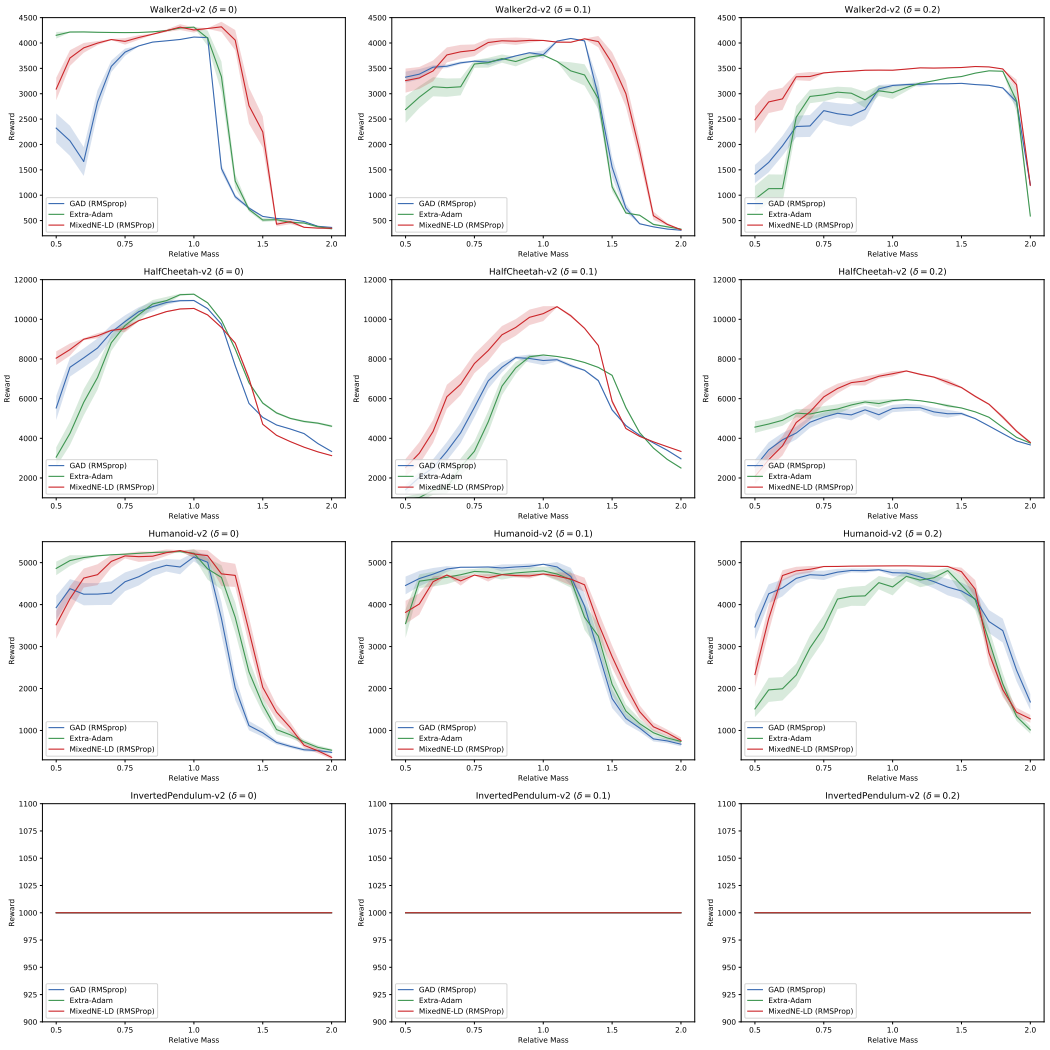


Figure 9: Average performance (over 5 seeds) of Algorithm 5 (TD3 with MixedNE-LD), and Algorithm 6 (TD3 with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0, 0.1, 0.2$. The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training.

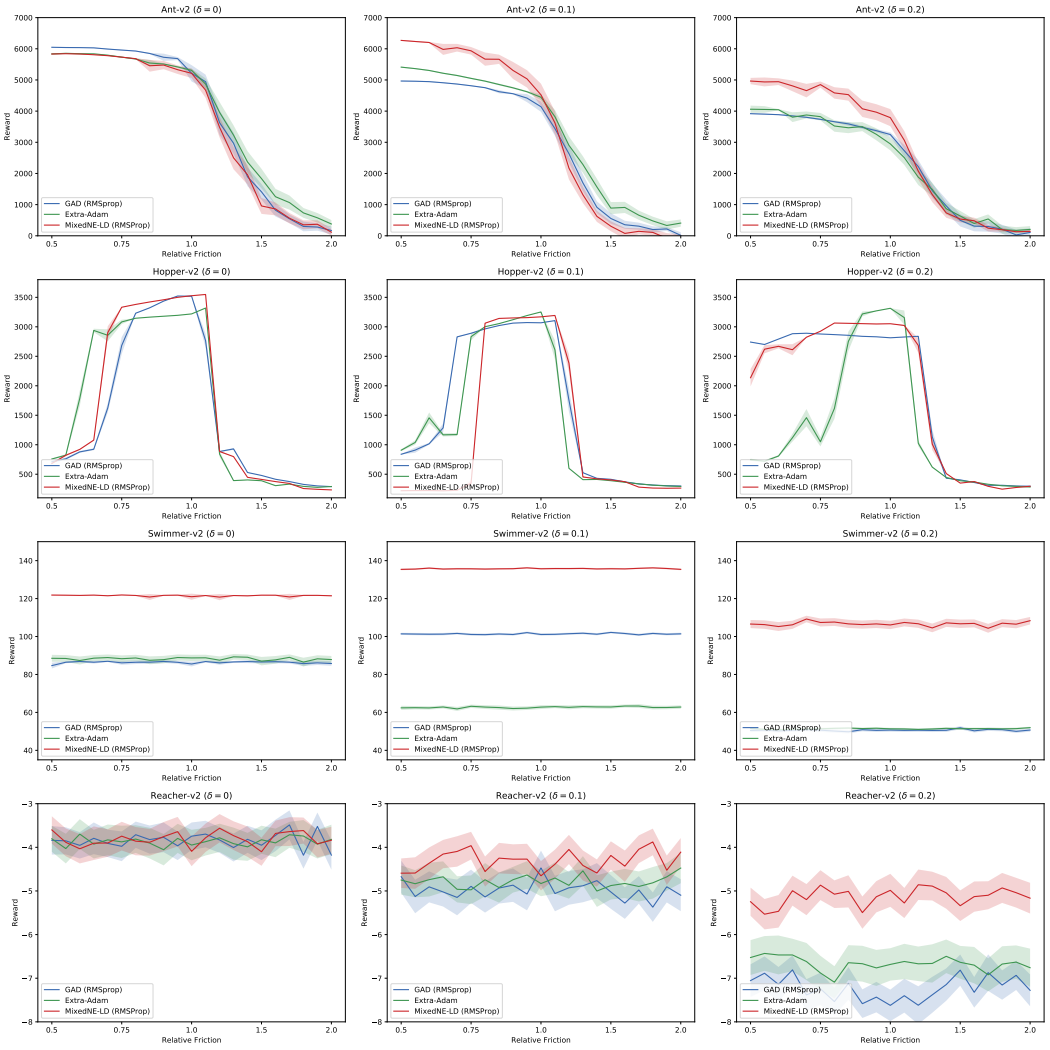


Figure 10: Average performance (over 5 seeds) of Algorithm 5 (TD3 with MixedNE-LD), and Algorithm 6 (TD3 with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0, 0.1, 0.2$. The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training.

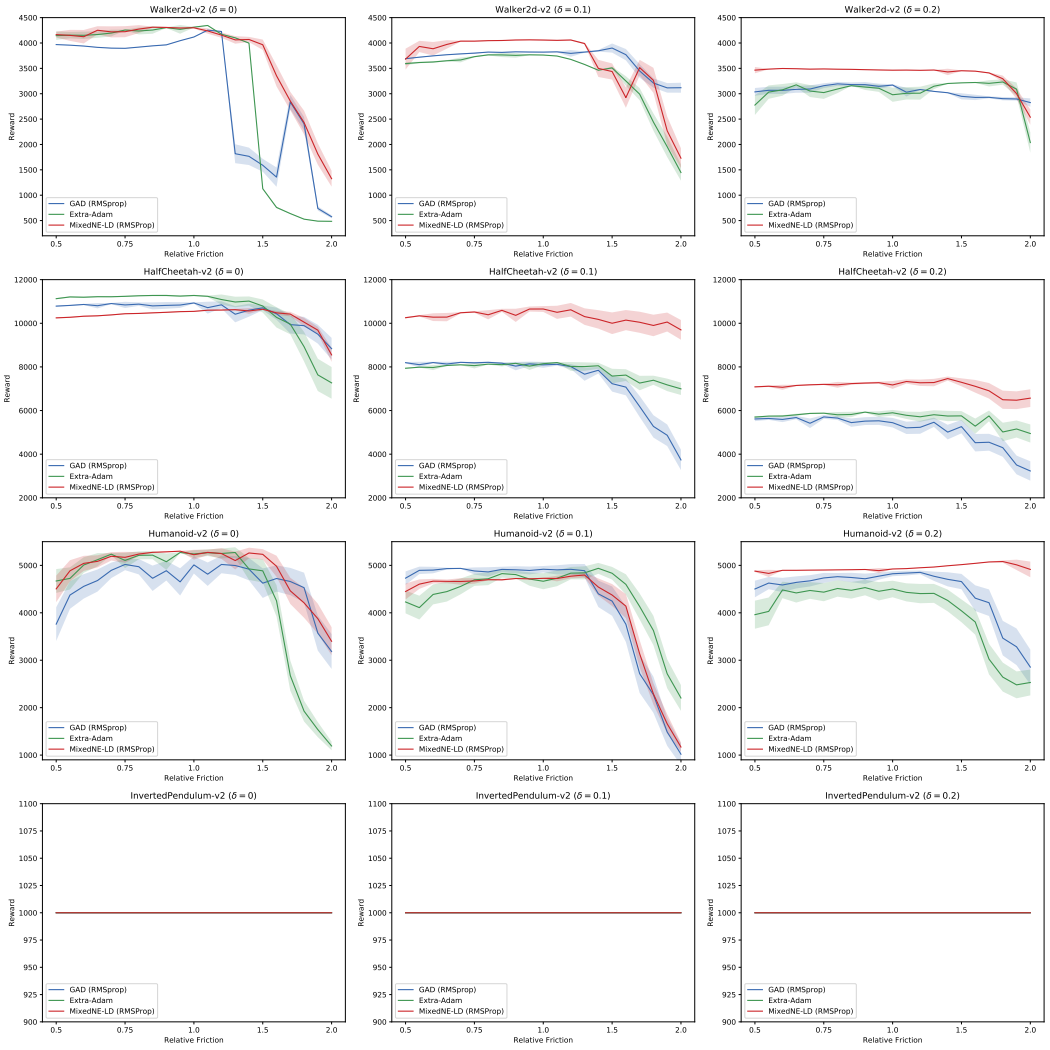


Figure 11: Average performance (over 5 seeds) of Algorithm 5 (TD3 with MixedNE-LD), and Algorithm 6 (TD3 with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0, 0.1, 0.2$. The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training.

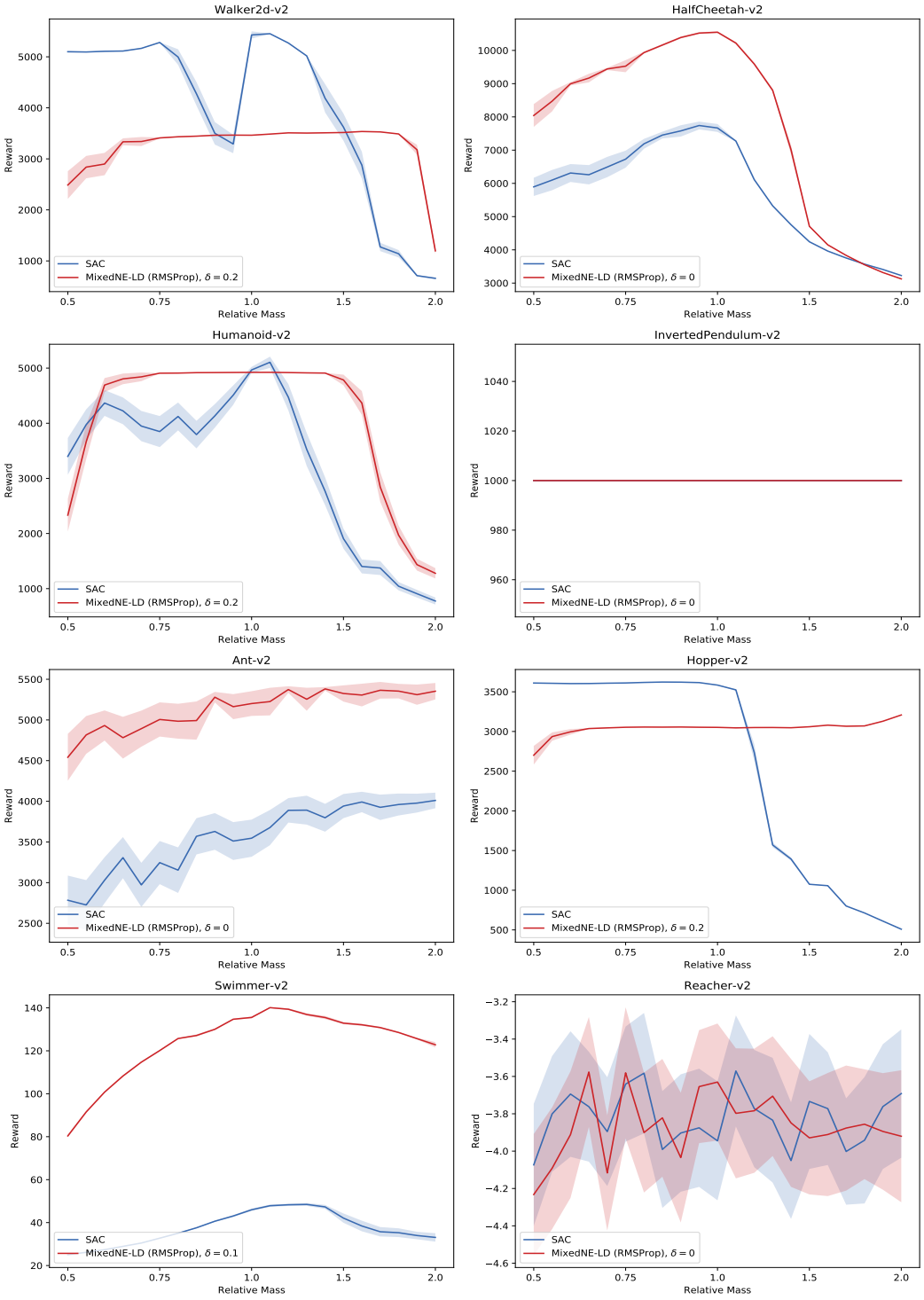


Figure 12: Average performance (over 5 seeds) of Algorithm 5 (TD3 with MixedNE-LD), and SAC, under the NR-MDP setting. The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training.

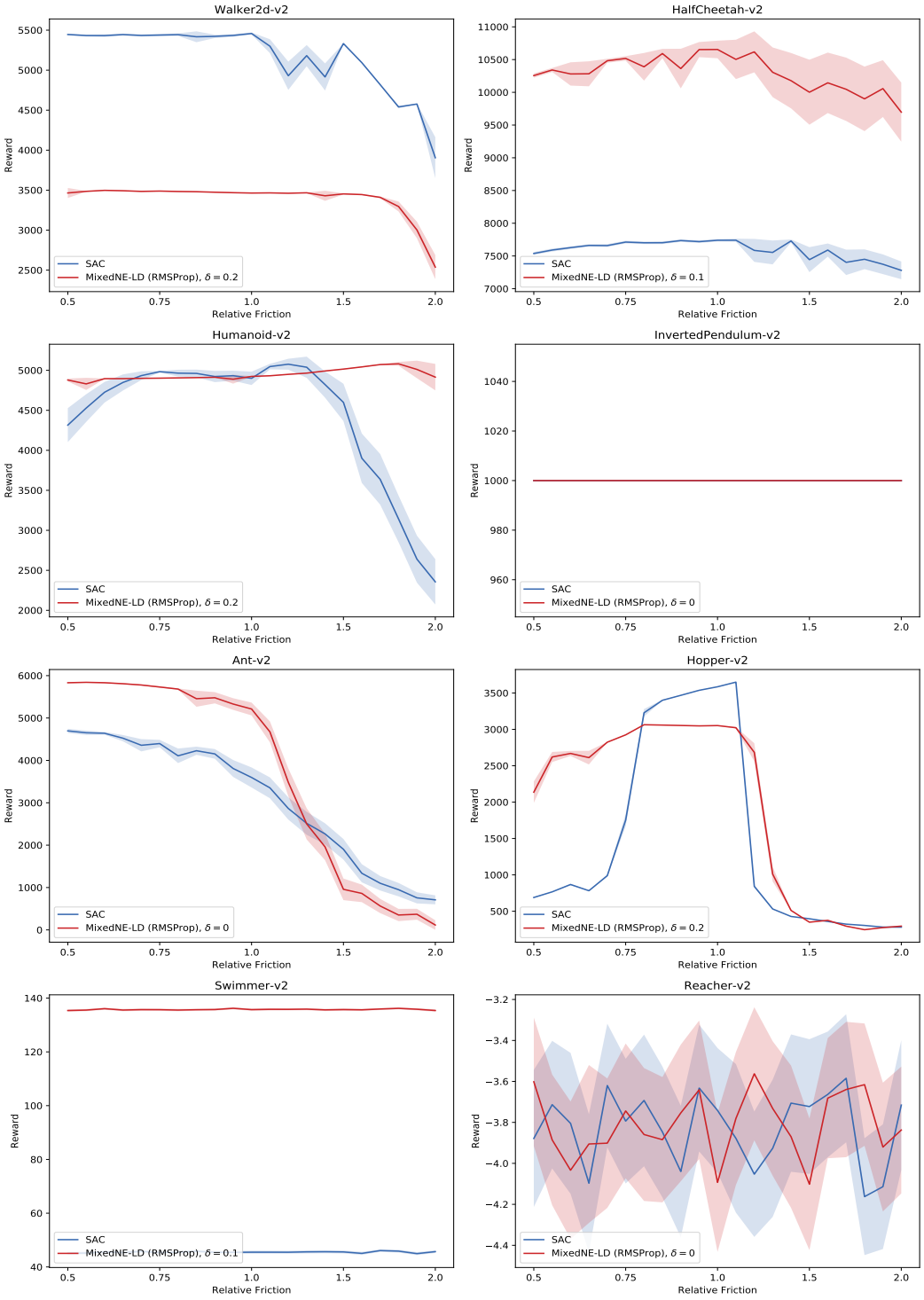
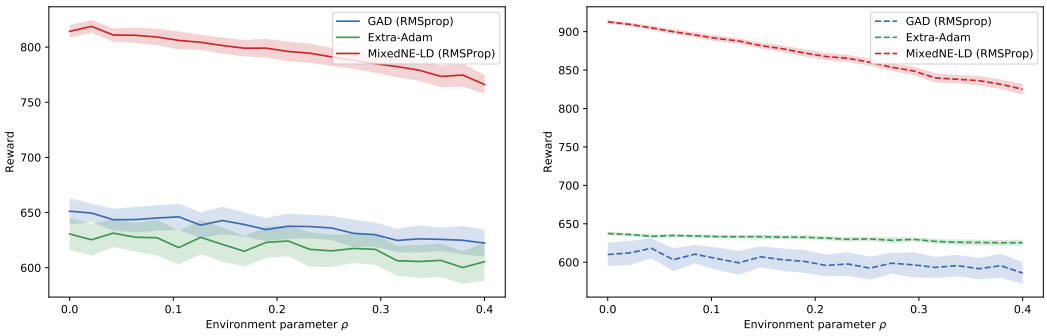


Figure 13: Average performance (over 5 seeds) of Algorithm 5 (TD3 with MixedNE-LD), and SAC, under the NR-MDP setting. The evaluation is performed without adversarial perturbations, on a range of friction values not encountered during training.



(a) $\delta = 0.1$ (b) $\delta = 0$

Figure 14: Average performance (over 5 seeds) of Algorithm 7, and Algorithm 8 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$ and 0 (training on nominal environment $\rho_0 = 0.2$). The evaluation is performed without adversarial perturbations, on a range of environment parameters not encountered during training.

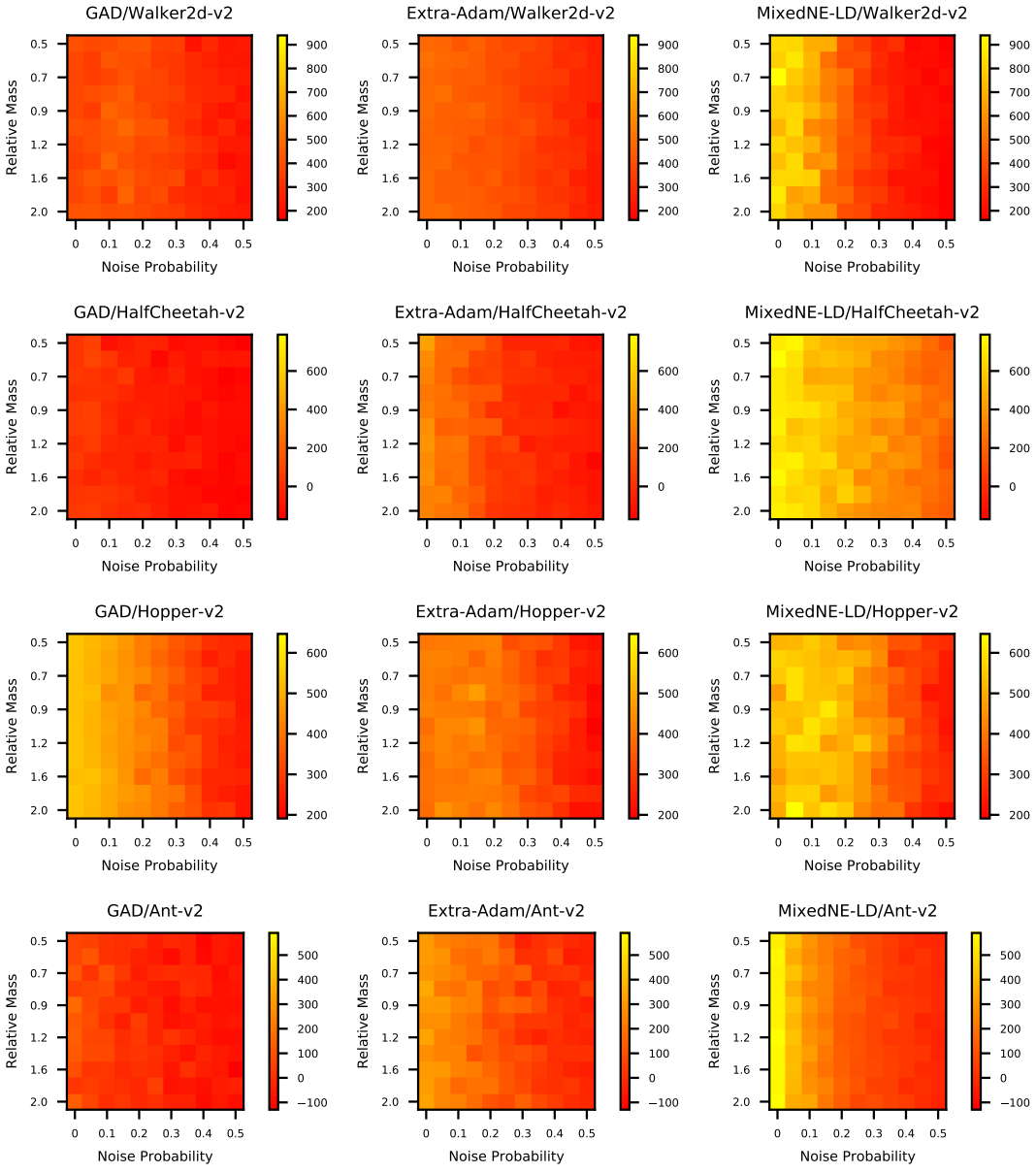


Figure 15: Average performance (over 5 seeds) of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of noise probability and mass values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant.

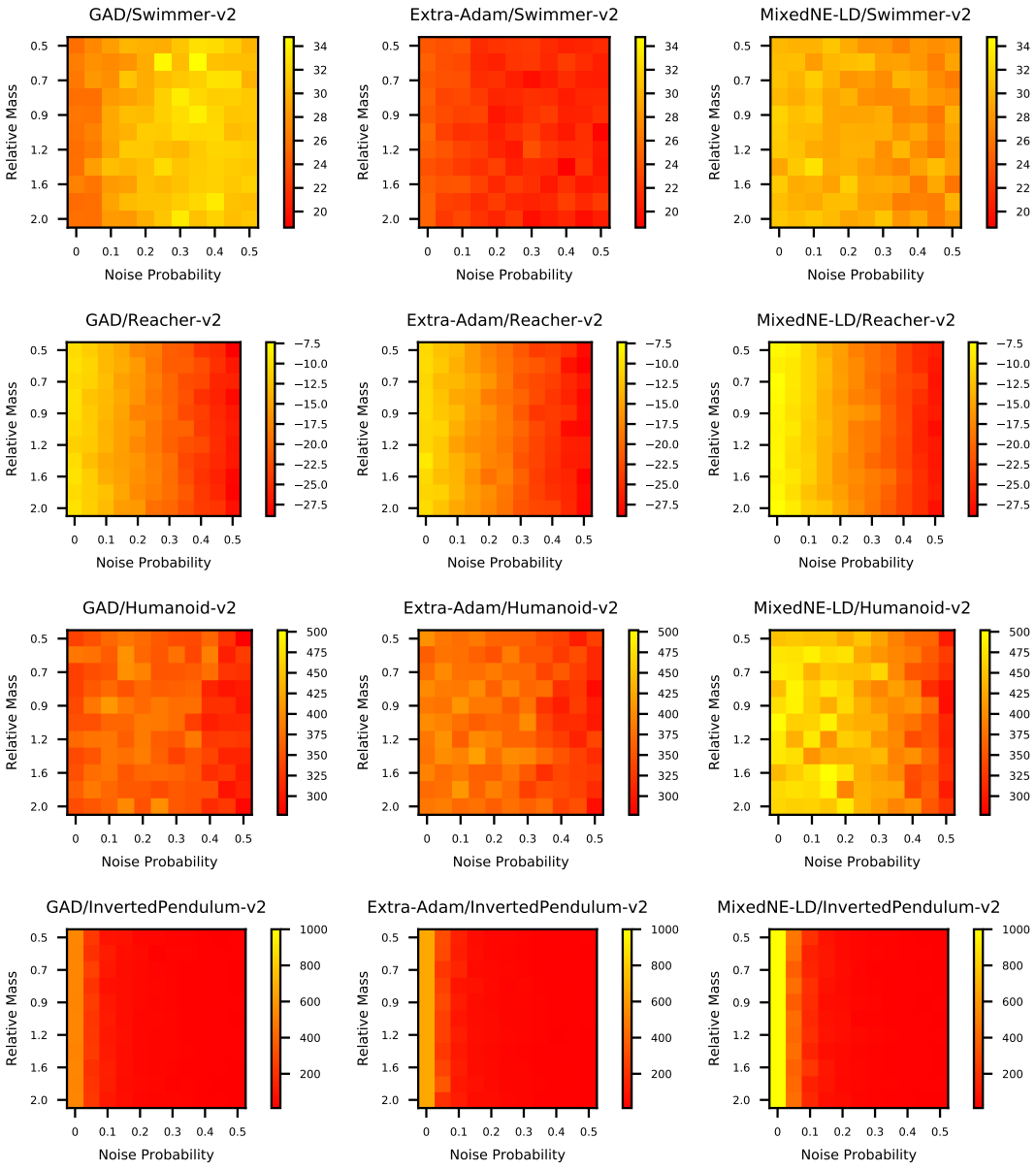


Figure 16: Average performance (over 5 seeds) of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of noise probability and mass values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum.

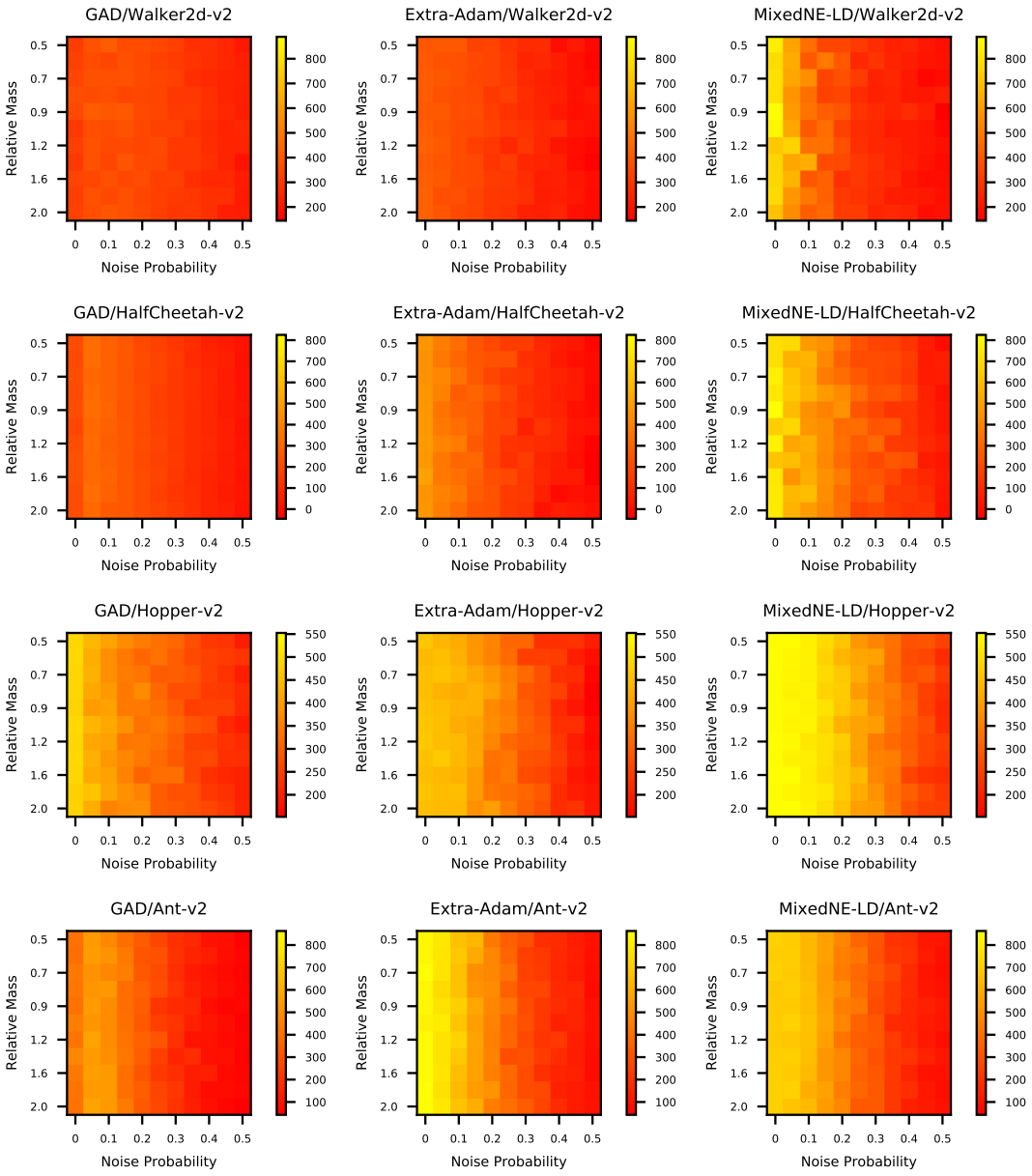


Figure 17: Average performance (over 5 seeds) of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of noise probability and mass values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant.

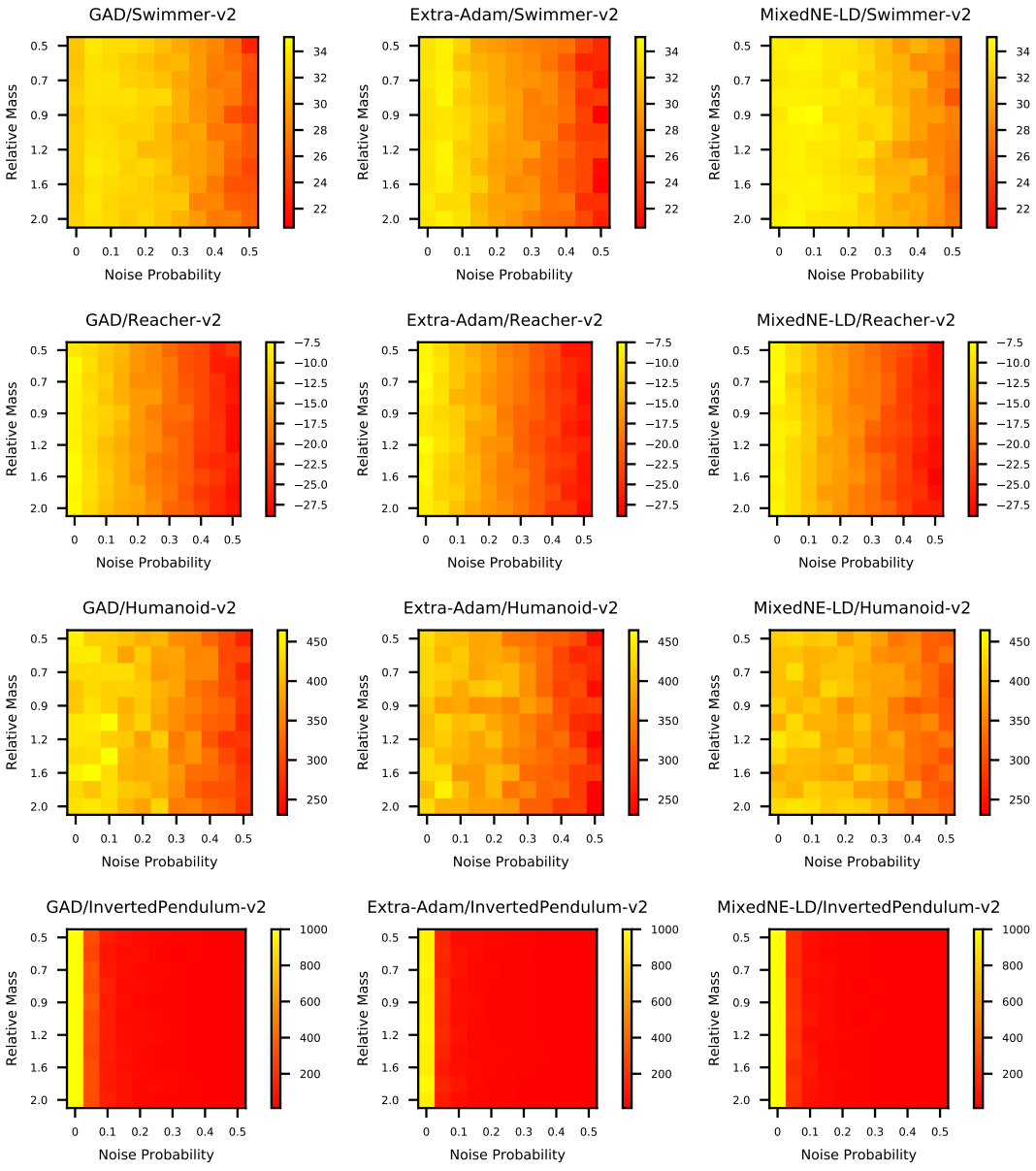


Figure 18: Average performance (over 5 seeds) of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of noise probability and mass values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum.

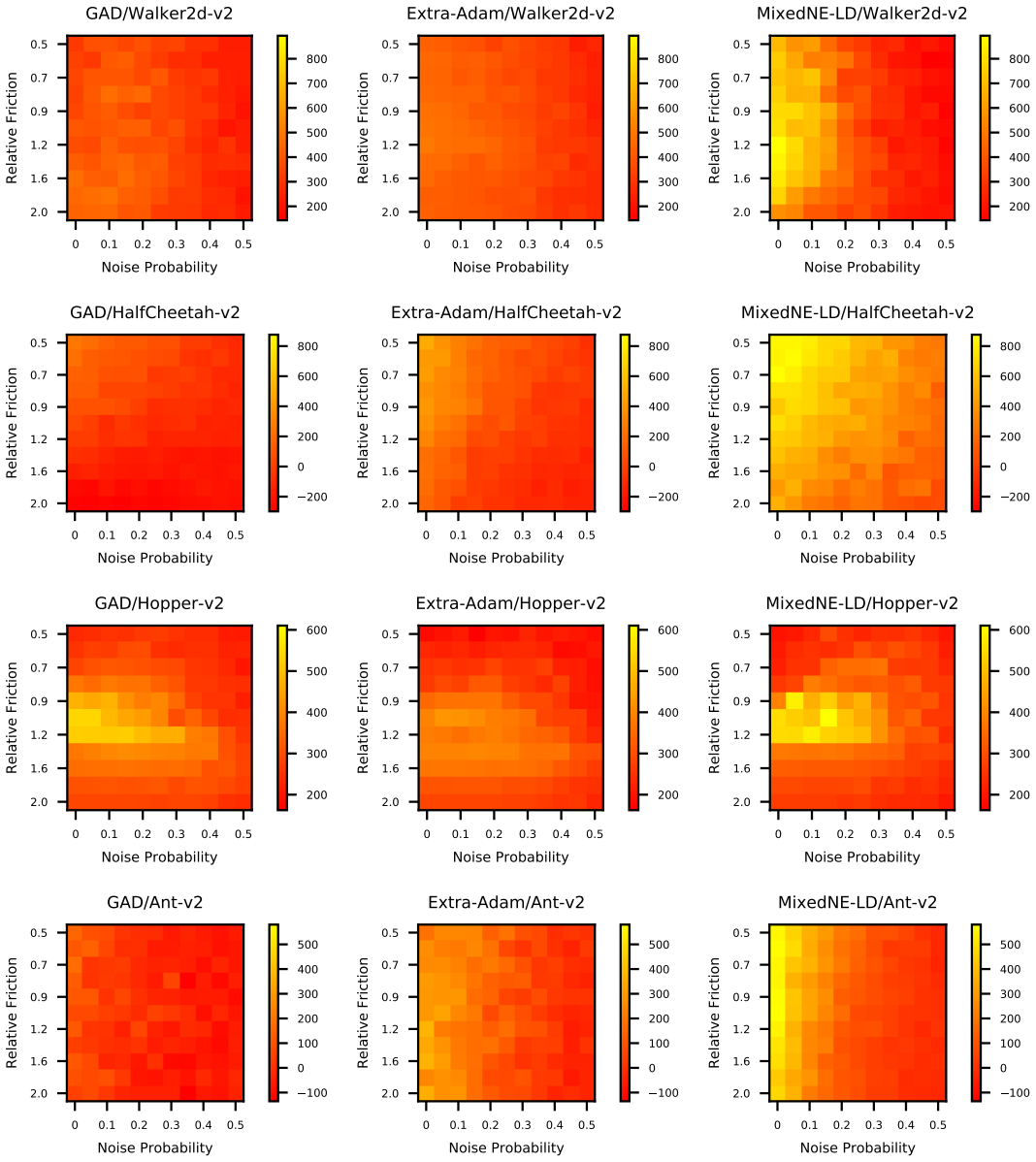


Figure 19: Average performance (over 5 seeds) of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of noise probability and friction values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant.

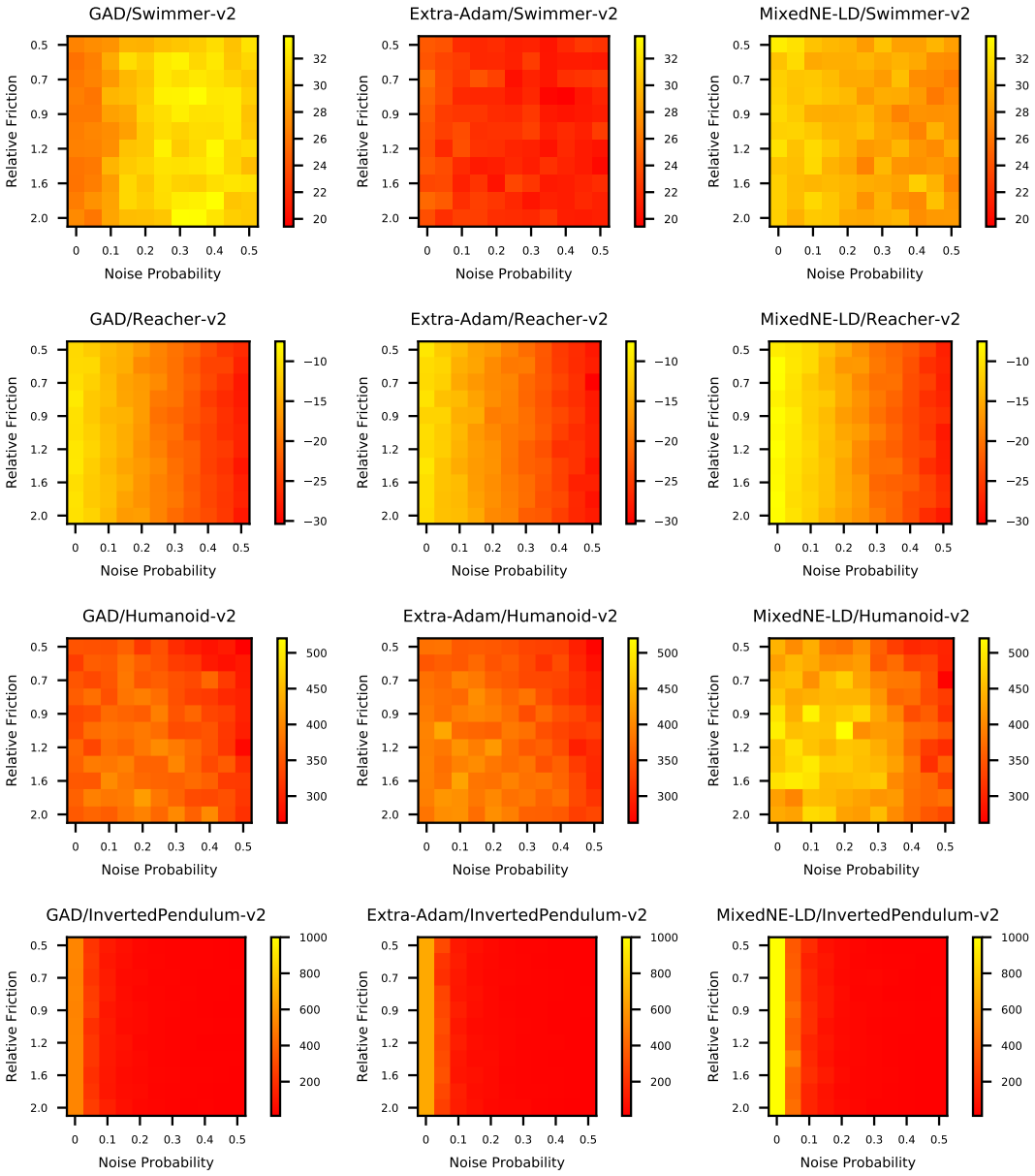


Figure 20: Average performance (over 5 seeds) of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of noise probability and friction values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum.

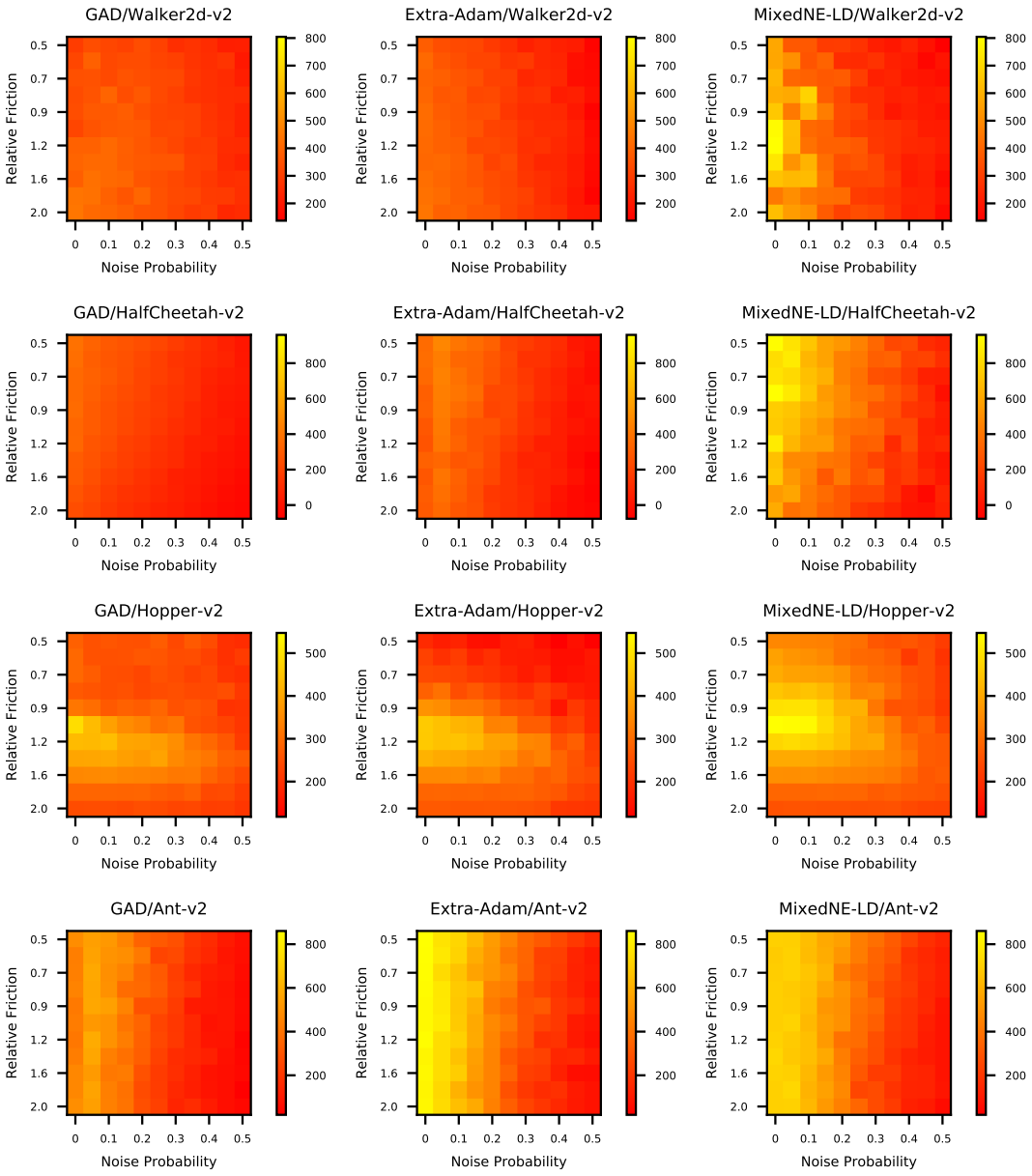


Figure 21: Average performance (over 5 seeds) of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of noise probability and friction values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant.

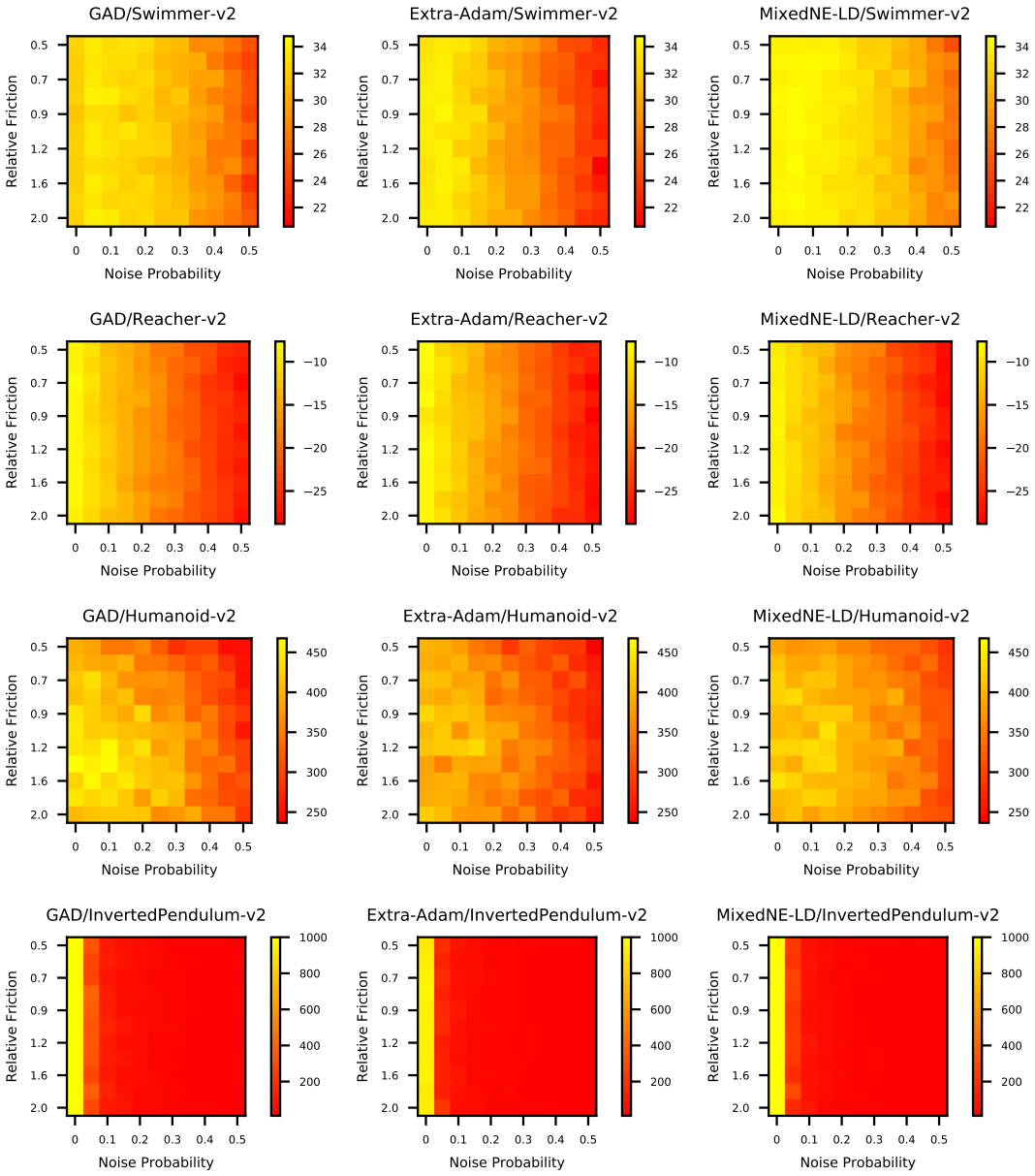


Figure 22: Average performance (over 5 seeds) of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of noise probability and friction values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum.

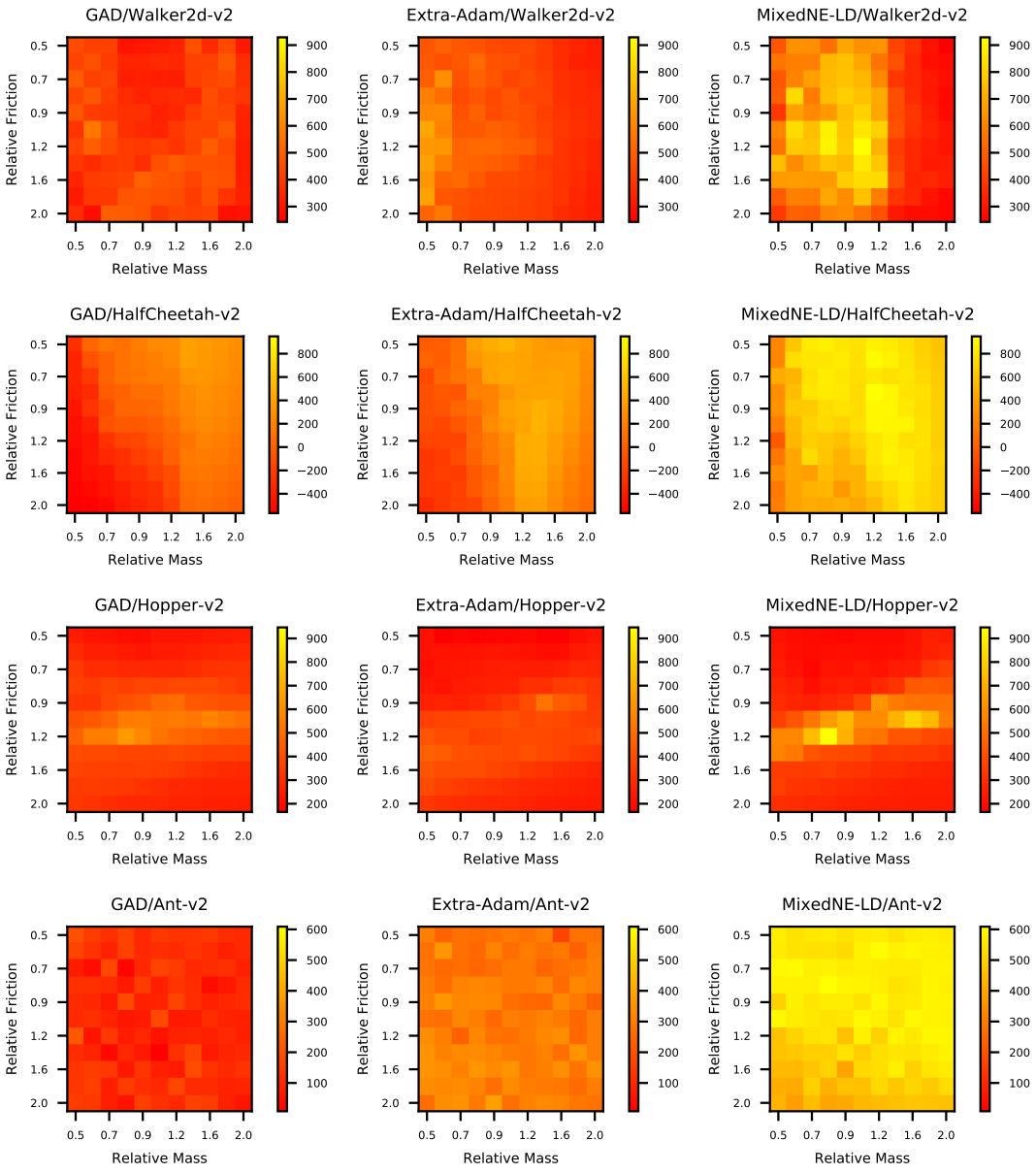


Figure 23: Average performance (over 5 seeds) of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of friction and mass values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant.

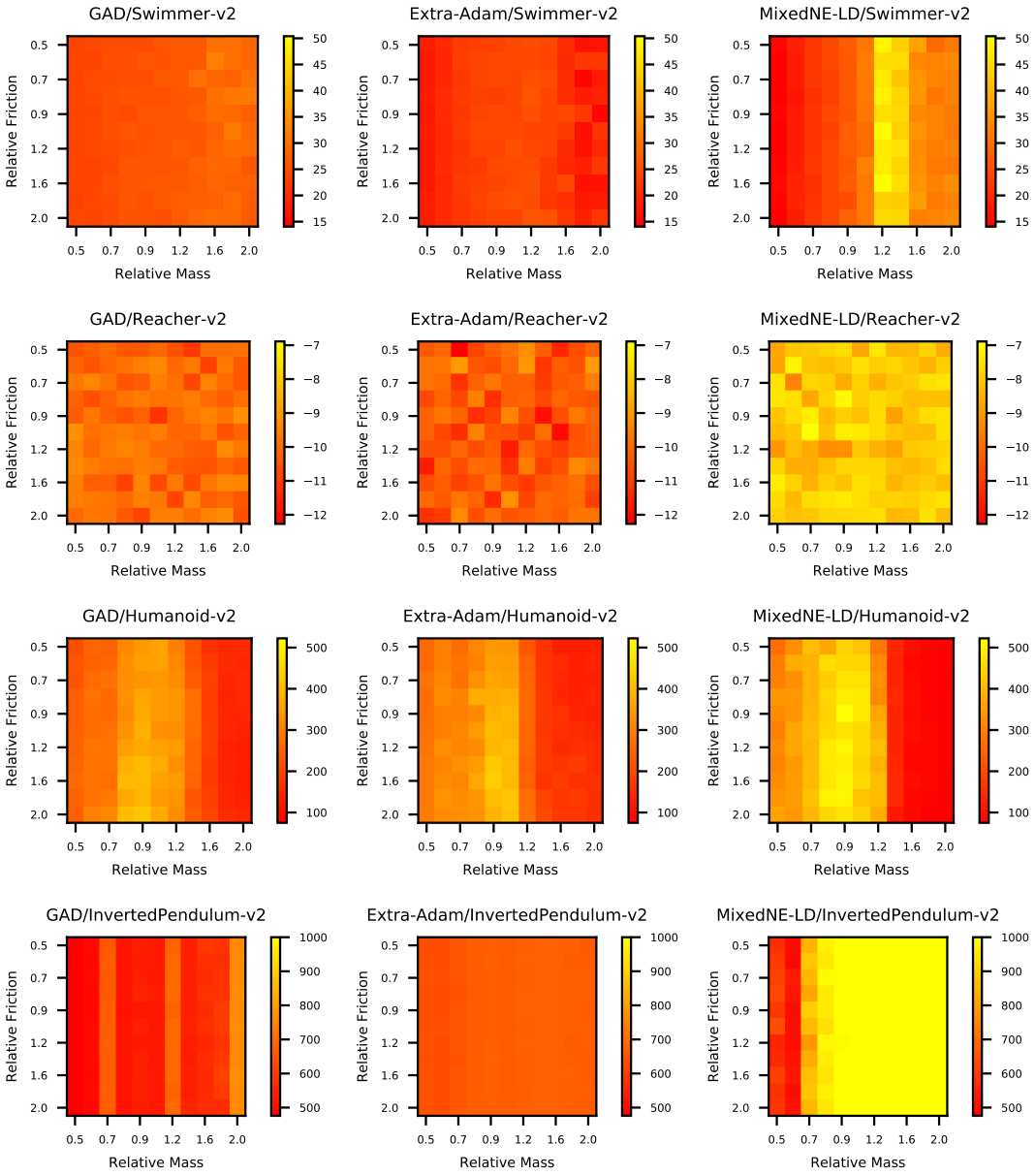


Figure 24: Average performance (over 5 seeds) of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0.1$. The evaluation is performed on a range of friction and mass values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum.

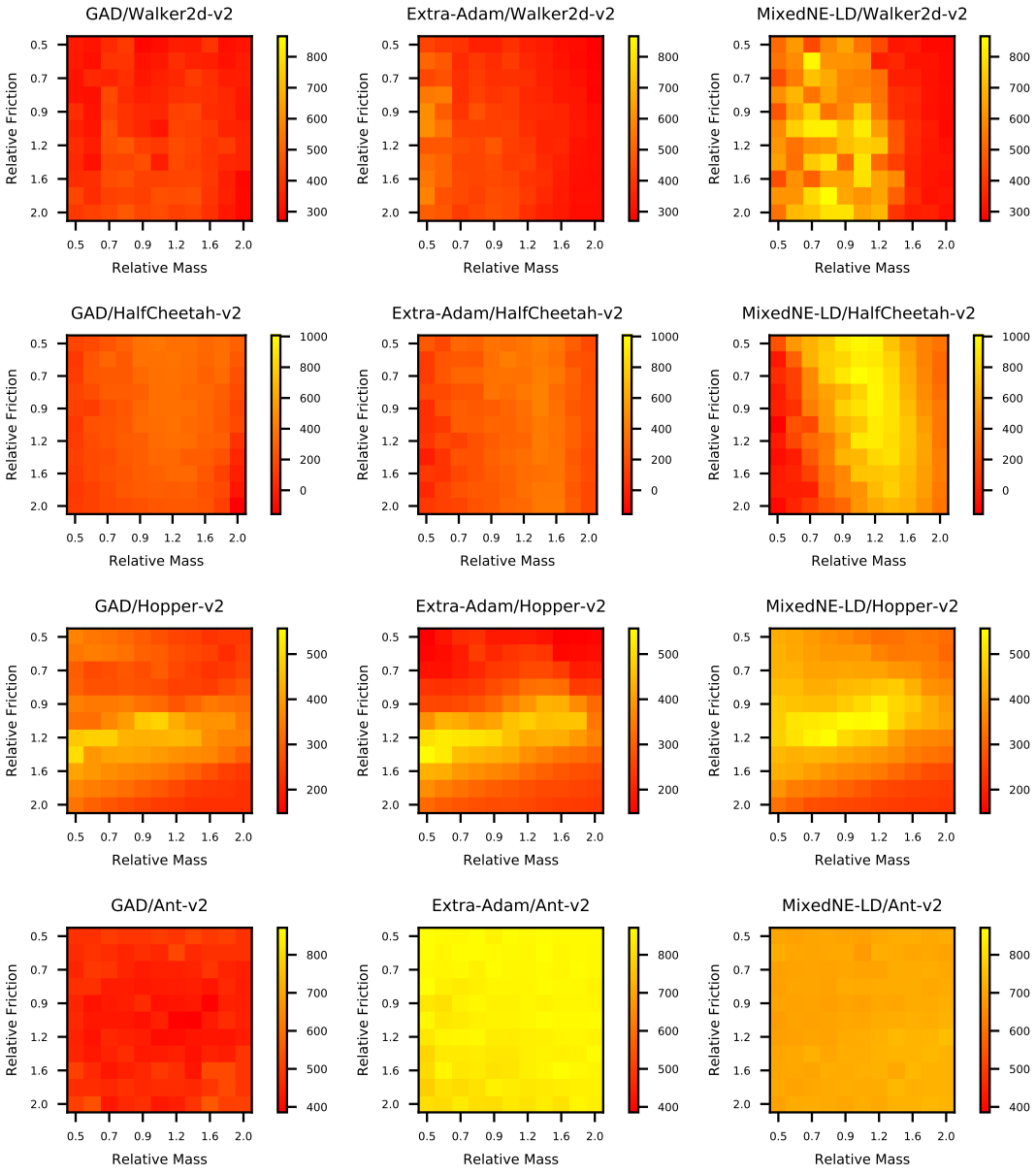


Figure 25: Average performance (over 5 seeds) of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of friction and mass values not encountered during training. Environments: Walker, HalfCheetah, Hopper, and Ant.

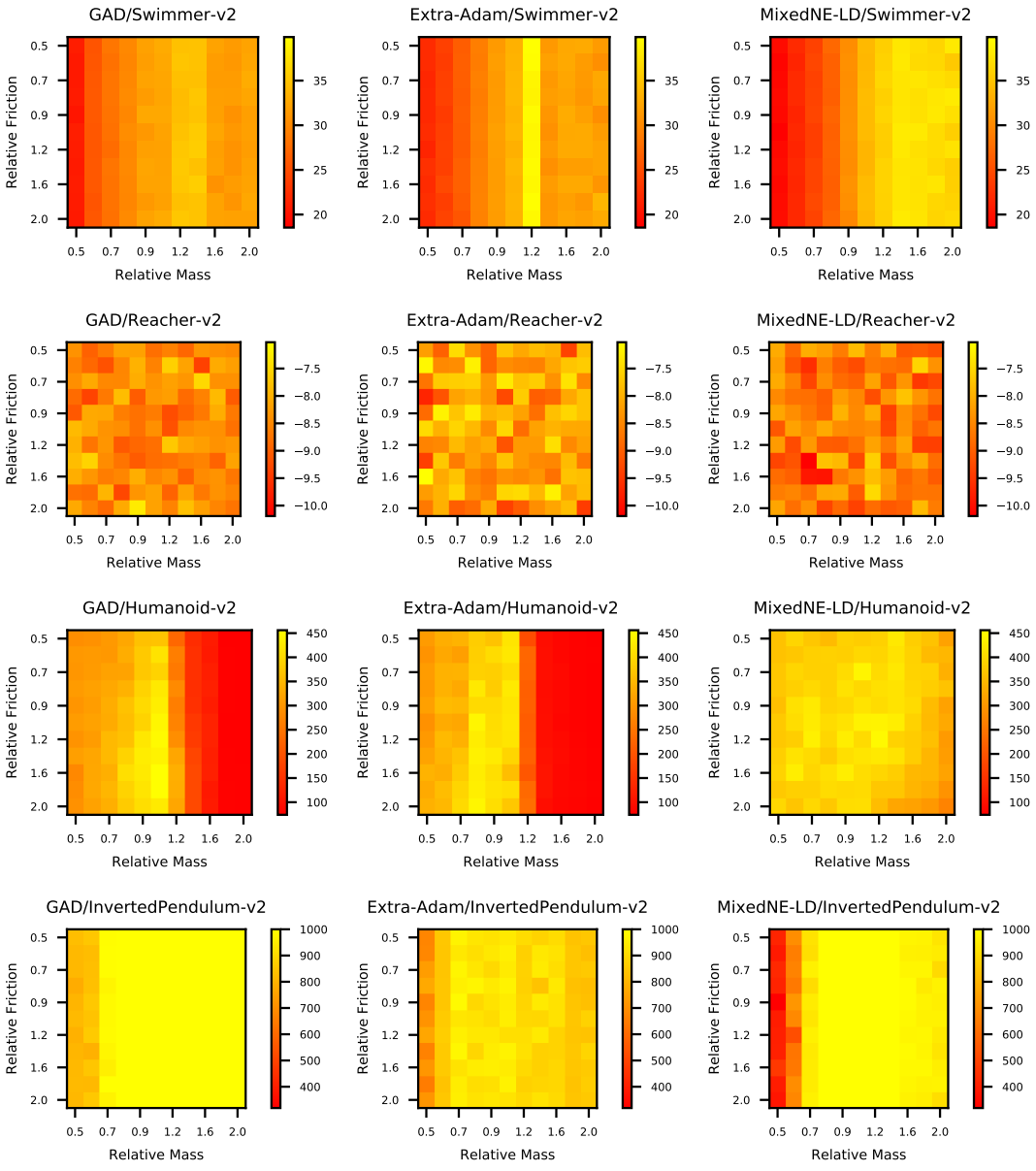


Figure 26: Average performance (over 5 seeds) of Algorithm 3, and Algorithm 4 (with GAD and Extra-Adam), under the NR-MDP setting with $\delta = 0$. The evaluation is performed on a range of friction and mass values not encountered during training. Environments: Swimmer, Reacher, Humanoid, and InvertedPendulum.