Appendix

A. Proofs

Proposition 3.1. *Let d be a pseudometric in $\mathbb{M}$, then $\mathcal{F}(d)$ is a pseudometric in $\mathbb{M}$.*

Proof. Let d be a pseudometric in $\mathbb{M}$, we show that $\mathcal{F}(d)$ respects all properties in Definition 1 and therefore is a pseudometric. Let $(s_1, a_1), (s_2, a_2), (s_3, a_3) \in \mathcal{S} \times \mathcal{A}$ and their associated rewards r_1, r_2, r_3 and next states s'_1, s'_2, s'_3:

- the pseudo-distance of a couple to itself is null:

$$\mathcal{F}(d)(s_1, a_1; s_1, a_1) = \underbrace{|r_1 - r_1|}_{=0} + \gamma \mathop{\mathbb{E}}_{u \in \mathcal{U}(\mathcal{A})} \underbrace{d(s'_1, u; s'_1, u)}_{=0 \text{ since } d \text{ is a pseudometric}} = 0;$$

- symmetry:

$$\mathcal{F}(d)(s_1, a_1; s_2, a_2) = \underbrace{|r_1 - r_2|}_{=|r_2 - r_1|} + \gamma \mathop{\mathbb{E}}_{u \in \mathcal{U}(\mathcal{A})} \underbrace{d(s'_1, u; s'_2, u)}_{=d(s'_2, u; s'_1, u) \text{ since } d \text{ is a pseudometric}} = \mathcal{F}(d)(s_2, a_2; s_1, a_1);$$

- triangular inequality:

$$\begin{aligned}
\mathcal{F}(d)(s_1, a_1; s_3, a_3) &= |r_1 - r_3| + \gamma \mathop{\mathbb{E}}_{u \in \mathcal{U}(\mathcal{A})} d(s'_1, u; s'_3, u) \\
&\leq |r_1 - r_2| + |r_2 - r_3| + \gamma \mathop{\mathbb{E}}_{u \in \mathcal{U}(\mathcal{A})} d(s'_1, u; s'_2, u) + d(s'_2, u; s'_3, u) \\
&\leq \mathcal{F}(d)(s_1, a_1; s_2, a_2) + \mathcal{F}(d)(s_2, a_2; s_3, a_3).
\end{aligned}$$

□

Proposition 3.2. *Let d be a pseudometric in $\mathbb{M}$. We note $\|d\|_\infty$ as $\max_{s,s' \in \mathcal{S}} \max_{a,a' \in \mathcal{A}} d(s, a; s', a')$. The operator $\mathcal{F}$ is a γ-contraction for $\|\cdot\|_\infty$.*

Proof. Let $d_1, d_2 \in \mathbb{M}$, let $(s_1, a_1), (s_2, a_2) \in \mathcal{S} \times \mathcal{A}$ and their associated rewards r_1, r_2 and next states s'_1, s'_2, we have:

$$\begin{aligned}
&\mathcal{F}(d_1)(s_1, a_1; s_2, a_2) - \mathcal{F}(d_2)(s_1, a_1; s_2, a_2) \\
&\qquad = |r_1 - r_2| - |r_1 - r_2| + \gamma \mathop{\mathbb{E}}_{u \in \mathcal{U}(\mathcal{A})} d_1(s'_1, u; s'_2, u) - \gamma \mathop{\mathbb{E}}_{u \in \mathcal{U}(\mathcal{A})} d_2(s'_1, u; s'_2, u) \\
&\qquad = \gamma \mathop{\mathbb{E}}_{u \in \mathcal{U}(\mathcal{A})} d_1(s'_1, u; s'_2, u) - d_2(s'_1, u; s'_2, u).
\end{aligned}$$

Therefore, we have:

$$\begin{aligned}
|\mathcal{F}(d_1)(s_1, a_1; s_2, a_2) - \mathcal{F}(d_2)(s_1, a_1; s_2, a_2)| &\leq \gamma \mathop{\mathbb{E}}_{u \in \mathcal{U}(\mathcal{A})} |d_1(s'_1, u; s'_2, u) - d_2(s'_1, u; s'_2, u)| \\
&\leq \gamma \max_{u \in \mathcal{A}} |d_1(s'_1, u; s'_2, u) - d_2(s'_1, u; s'_2, u)| \\
&\leq \gamma \max_{s,s' \in \mathcal{S}} \max_{u,u' \in \mathcal{A}} |d_1(s, u; s', u') - d_2(s, u; s', u')| \\
&\leq \gamma \|d_1 - d_2\|_\infty.
\end{aligned}$$

We thus have that $\|\mathcal{F}(d_1) - \mathcal{F}(d_2)\|_\infty \leq \gamma \|d_1 - d_2\|_\infty$, therefore $\mathcal{F}$ is a γ-contraction for $\|\cdot\|_\infty$. □

Proposition 3.3. *$\mathcal{F}$ has a unique fixed point d^* in $\mathbb{M}$. Suppose $d_0 \in \mathbb{M}$ then $\lim_{n \to \infty} \mathcal{F}^n(d_0) = d^*$.*

Proof. This is a direct application of the Banach theorem (Banach, 1922). $\mathcal{F}$ is a γ-contracting operator with $\gamma \in [0, 1)$, in the metric space $((\mathcal{S} \times \mathcal{A}) \times (\mathcal{S} \times \mathcal{A}), \|\cdot\|_\infty)$, therefore using the Banach theorem we have that $\mathcal{F}$ has a unique fixed point d^* and $\forall d_0 \in \mathbb{M}, \lim_{n\to\infty} \mathcal{F}^n(d_0) = d^*$. □

Proposition 3.4. *Suppose sufficient coverage of the state-action space: $\exists \epsilon > 0$ such that for any pairs of state-action pairs $(s, a), (\hat{s}, \hat{a}) \in (\mathcal{S} \times \mathcal{A}) \times (\mathcal{S} \times \mathcal{A})$, $(s, a), (\hat{s}, \hat{a})$ is sampled with at least probability ϵ, then the repeated application of $\hat{\mathcal{F}}$ converges to the fixed point d^* of $\mathcal{F}$.*

Proof. The repeated application of $\hat{\mathcal{F}}$ is an asynchronous fixed point iteration scheme. The convergence to d^* (almost surely) is a direct application of Proposition 3 from Bertsekas & Tsitsiklis (1991). Note that the state-action coverage assumption enables to apply this result since all pairs of state-action are visited an infinite number of times (almost surely).

□

B. Implementation

In this section, we provide a detailed description of the experimental study.

Offline datasets preprocessing. We use datasets from Fu et al. (2020). We scale the rewards by shifting them by $-\min_{r\sim\mathcal{D}} r$ and dividing them by $\max_{r\sim\mathcal{D}} r - \min_{r\sim\mathcal{D}} r$ for both pseudometric learning and policy learning. This enables to have comparable range of rewards between environments.

Pseudometric learning. The Siamese networks Φ et Ψ have the same architecture: a two-layer MLP of size $(1024, 32)$ with relu activation on top of the first layer. Note that Φ takes the concatenation of state and action as input, whereas Ψ only takes the state as an input. We use a discount factor $\gamma = 0.9$ (which is different than the discount factor from the agent) in the experiments (we noticed some instabilities on human datasets, which contains less transitions than the rest of the datasets, if the discount factor is larger).

We minimize the losses $\hat{\mathcal{L}}_\Phi$ and $\hat{\mathcal{L}}_\Psi$ using the Adam optimizer with a learning rate 10^{-3}. The batch size used to compute the losses is 256. The bootstrapped estimate in $\hat{\mathcal{L}}_\Phi$ is estimated with 256 actions sampled uniformly. We train the two networks by iteratively taking a gradient step on each loss for 2.10^6 gradient steps with parameters updated using exponential parameters averaging with a rate $\tau = 0.005$

Once the Ψ network is trained, we derive the k-nearest neighbors of each state in $\mathcal{D}$ for the distance induced by Ψ, with $k = 50$. The nearest neighbors are computed using the scikit-learn implementation of the kd-tree algorithm, taking advantage of multiprocessing (with 50 CPUs).

Agent training. We re-implemented the TD3 agent from Fujimoto et al. (2019) in JAX (Bradbury et al., 2018). We use the default hyperparameters (and did not perform HP search).

For the critic, we used a three-layer network with size $(256, 256, 1)$ with tanh activation on top of the first layer and elu activation on top of the second layer. For the policy, we used a three-layer network with size $(256, 256, |\mathcal{A}|)$ where $|\mathcal{A}|$ is the dimension of the action space, with tanh activation on top of the first layer, elu activation on top of the second layer and tanh activation on top of the last layer. We used the Adam optimizer for both the actor and the critic and used a learning rate of 3.10^{-4} (consistently with Fujimoto et al. (2019)) and trained them using batch of transitions of size 256 sampled uniformly in $\mathcal{D}$, for 500000 gradient steps.

We led experiments with the following bonuses $\bar{b}$ (and focused on the first one as it led to better empirical performance):

- $\bar{b}(s, a) = Q_{\bar{\omega}}(s, a) \exp(-\beta d_{\mathcal{D}}(s, a))$
- $\bar{b}(s, a) = \exp(-\beta d_{\mathcal{D}}(s, a))$
- $\bar{b}(s, a) = 1 - \exp(\beta d_{\mathcal{D}}(s, a))$

We led a hyperparameter search for both the locomotion environments and the hand manipulation environments on $\alpha_a, \alpha_c, \beta$. We selected α_a, α_c in $\{1, 5, 10\}$, $\beta \in \{0.1, 0.25, 0.5\}$ as the better combination on the average normalized performance

on the tasks (averaged over 3 seeds). We re-ran the best combination of hyperparameters for 10 seeds and report results averaged over the 10 seeds and 10 evaluation episodes per seed. We found that the best combination for hand manipulation tasks was $\alpha_a = 10, \alpha_c = 10, \beta = 0.5$ and for locomotion tasks was $\alpha_a = 5, \alpha_c = 1, \beta = 0.5$.

C. Metric Visualization

In this section, we show the state similarity learned by PLOFF (Ψ network) and visualize it for MuJoCo locomotion environments (we could not provide such visualizations on Adroit tasks since states cannot be retrieved from observations, which is the necessary condition to generate rendering).

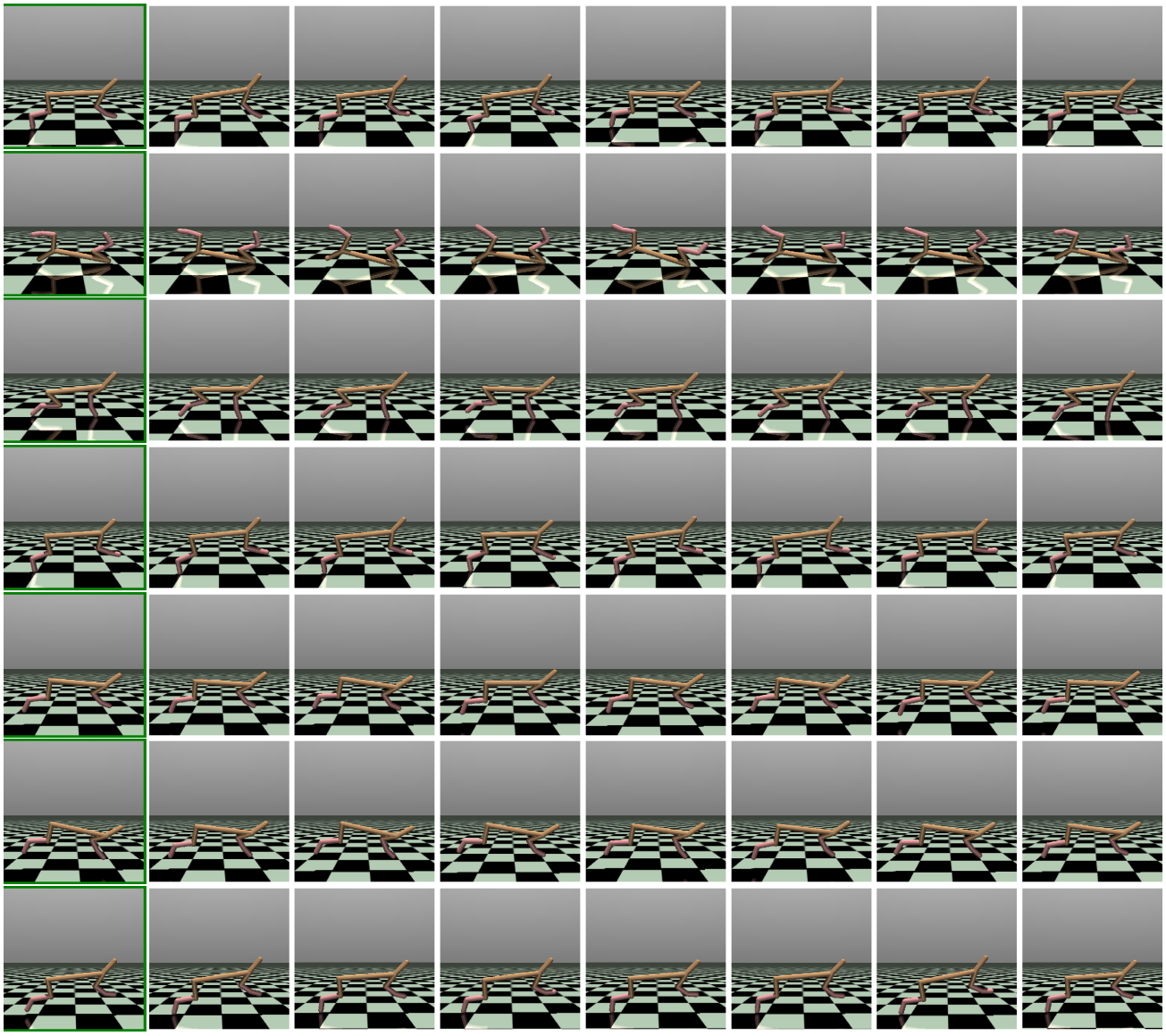

Figure 6. State similarity learned by PLOFF for HalfCheetah on the medium-replay dataset. For each row, the leftmost image is the state for which we compute nearest neighbors in the dataset $\mathcal{D}$ for the metric induced by Ψ (ranked by decreasing level of similarity).

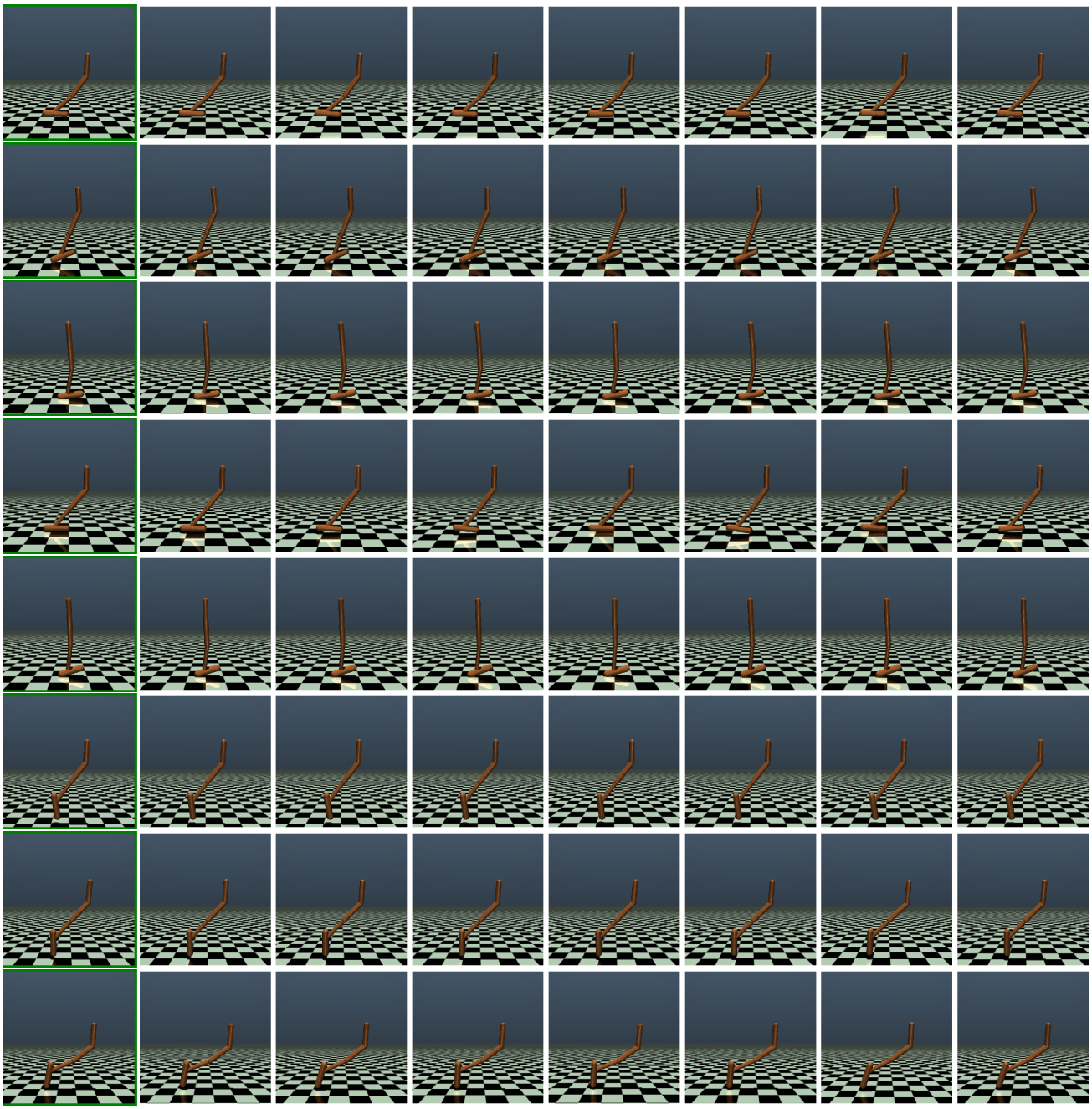

Figure 7. State similarity learned by PLOFF for Hopper on the medium-replay dataset. For each row, the leftmost image is the state for which we compute nearest neighbors in the dataset $\mathcal{D}$ for the metric induced by Ψ (ranked by decreasing level of similarity).

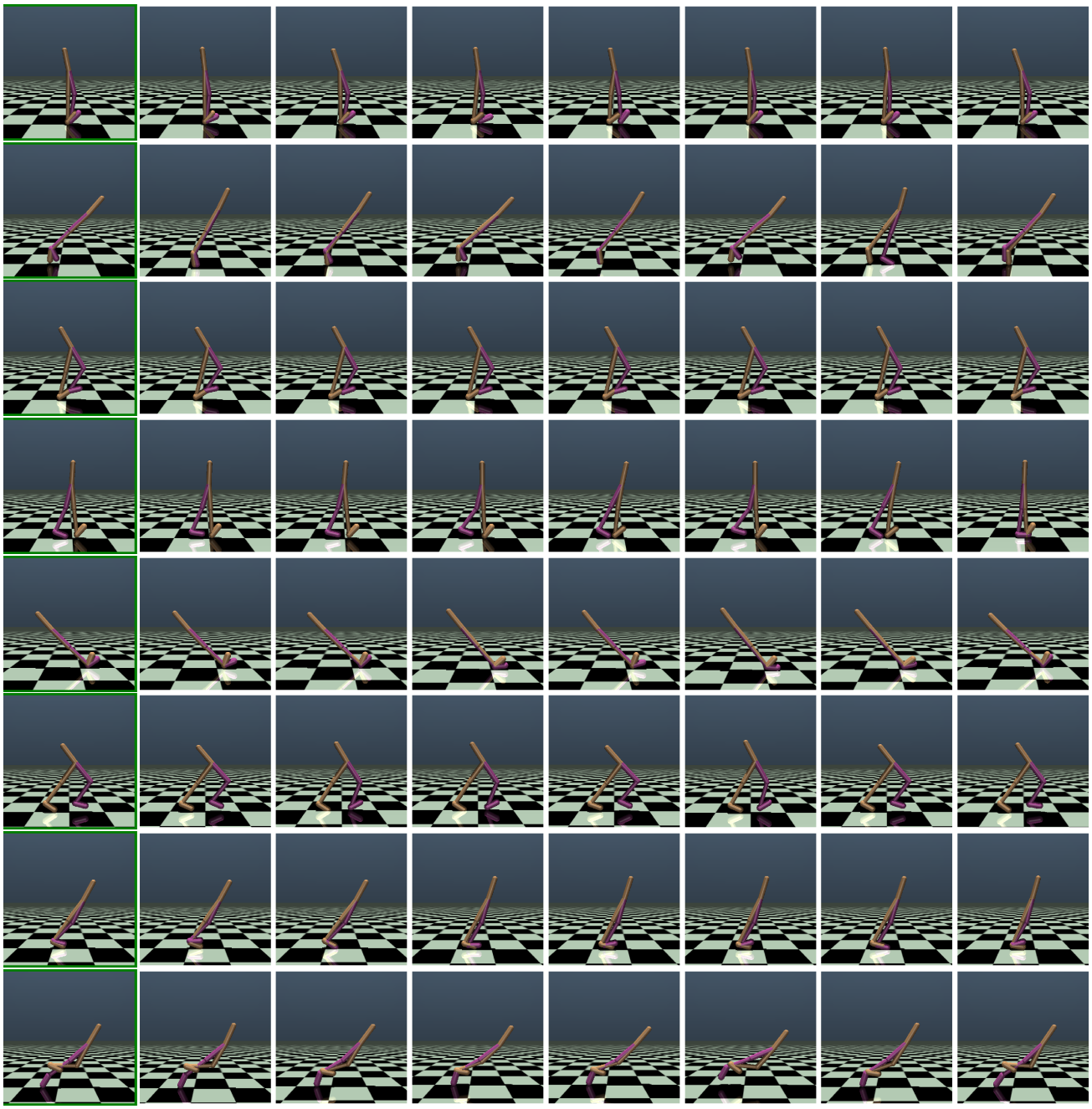

Figure 8. State similarity learned by PLOFF for Walker2d on the medium-replay dataset. For each row, the leftmost image is the state for which we compute nearest neighbors in the dataset $\mathcal{D}$ for the metric induced by Ψ (ranked by decreasing level of similarity).