

A Training Details

A.1 Computing Resources and Experiments

All experiments are done on a single RTX 4090 GPU and 4 CPU cores. Each state-based experiment takes 12 hours for all method, following METRA [1], which trains each method for 9-10 hours. It corresponds to the different environment steps used for different experiments, as described in Table 1.

We use 3 (5) random seeds for each experiment in the main paper (the supplementary material). For Humanoid-Run in the main paper, we reported the result with one seed, but the five-seed results can be found in Figure 9. For all experiments, we report the mean and standard deviation of the results.

Table 1: # of environment steps for experiments.

Environment	TLDR	METRA	PEG	LEXA	APT	RND	Disagreement
Ant	44.2M	44.3M	0.7M	-	2.4M	4.1M	4.8M
HalfCheetah	38.6M	40.7M	0.7M	-	2.5M	4.2M	5.0M
AntMaze-Large	42.6M	62.3M	0.7M	-	2.4M	6.4M	5.0M
AntMaze-Ultra	37.5M	54.7M	0.6M	-	2.4M	4.5M	3.4M
Quadruped-Escape	28.0M	33.3M	0.6M	-	2.2M	4.5M	4.4M
Humanoid-Run	40.8M	57.6M	0.6M	-	3.5M	4.7M	4.7M
Quadruped (Pixel)	2.9M	3.1M	-	2.1M	-	-	-
Kitchen (Pixel)	1.1M	1.7M	-	1.0M	-	-	-

A.2 Implementation Details

Our method, TLDR, is implemented on top of the official implementation of METRA. Similar to METRA, we use SAC [37] for learning the goal-reaching policy and exploration policy. We train our temporal distance-aware representation $\phi(\mathbf{s})$ by maximizing the following objective:

$$\mathbb{E}_{\mathbf{s} \sim p_{\mathbf{s}}, \mathbf{g} \sim p_{\mathbf{g}}} [f(\|\phi(\mathbf{s}) - \phi(\mathbf{g})\|) + \lambda \cdot \min(\epsilon, 1 - \|\phi(\mathbf{s}) - \phi(\mathbf{s}')\|)], \quad (5)$$

where we apply affine-transformed softplus f to Equation (1):

$$f(x) = -\text{softplus}(500 - x, \beta = 0.01), \quad (6)$$

which alleviates the effect of too long distances $\|\phi(\mathbf{s}) - \phi(\mathbf{g})\|$, following QRL [14].

For METRA, PEG, and LEXA, we use their official implementation. For random exploration approaches (APT, RND, Disagreement), we use the implementation from URLB [38].

A.3 Hyperparameters

The hyperparameters used in our experiments can be found in Table 2.

For METRA, we use 2-D continuous skills for Ant, 16-D discrete skills for Half-Cheetah, 24-D discrete skills for Kitchen (Pixel), and 4-D continuous skills for other environments. We use the default values for the remaining hyperparameters.

In PEG, we use the same hyperparameters used in their AntMaze experiments. Since PEG use the normalized goal space, we measure the ranges of the observations and normalize goal states according to them.

In LEXA, we follow their hyperparameters and opt for the temporal distance reward for training the Achiever policy.

A.4 Environment Details

Ant. We use the MuJoCo Ant environment in OpenAI gym [41]. The observation space is 29-D and the action space is 8-D. Following METRA, we normalize the observations for Ant with fixed

Table 2: List of hyperparameters.

Hyperparameter	Value
Learning rate	0.0001
Learning rate for ϕ	0.0005
Batch size	1024 (State), 256 (Pixel)
Replay buffer size	10^6 (State), 3×10^5 (Quadruped (Pixel)), 10^5 (Kitchen)
Frame stack (Pixel)	3
Optimizer	Adam [39]
Relaxation constant ϵ in Eq. (1)	10^{-3}
$\dim \phi(s)$	8 (Kitchen), 4 (Others)
k in Eq. (2)	12
Initial λ	3×10^3
SAC entropy coefficient	0.01 (Kitchen), target entropy as $-\dim \mathcal{A} $ (others)
Discount factor γ	0.97 (Goal-reaching policy), 0.99 (Exploration policy)
Normalization	LayerNorm [40] for the critics, None for ϕ
Encoder for image observations	CNN
MLP dimensions	1024
MLP depths	2
Goal relabelling	0.75 (sampled from future observations), 0.25 (no relabelling)
# of gradient steps per epoch	50 (Ant, HalfCheetah, Humanoid-Run), 75 (AntMaze-Large), 100 (Kitchen), 150 (AntMaze-Ultra), 200 (Quadruped (Pixel))
# of episode rollouts per epoch	8
τ for updating the target network	0.995

mean and standard deviation of observations computed from randomly generated trajectories. The episode length is 200.

HalfCheetah We use the MuJoCo HalfCheetah environment in OpenAI gym [41]. The observation space is 18-D and the action space is 6-D. Following METRA, we normalize the observations for HalfCheetah with fixed mean and standard deviation of observations from randomly generated trajectories. The episode length is 200.

AntMaze-Large. We use antmaze-large-play-v2 in D4RL [32]. The observation and action spaces are the same with the Ant environment. The episode length is 300. To make exploration more challenging, we fix the initial location of the agent to be the bottom right corner of the maze, as shown in Figure 3c.

AntMaze-Ultra. We use antmaze-ultra-play-v0 proposed by Jiang et al. [33]. The observation and action spaces are the same with the Ant environment. The episode length is 600, since the maze is two times larger than that of AntMaze-Large. Similar to AntMaze-Large, we fix the initial location of the agent to be the bottom right corner of the maze, as shown in Figure 3d.

Quadruped-Escape. Quadruped-Escape is included in DeepMind Control Suite [31]. The quadruped robot is initialized in a basin surrounded by complex terrains, as described in Figure 3e. Due to the complex terrains, moving further away from the initial position is challenging. Similar to the AntMaze environments, we fix the terrain shape. Quadruped has 101-D observation space with 12-D action space. Additionally, we add the global x, y coordinates of the agent to the observation. The episode length is 200.

Humanoid-Run. We use the Humanoid-Run task from DeepMind Control Suite [31]. Humanoid has 55-D observation space with 21-D action space. Additionally, the global x, y coordinates of the agent is added to observation. The episode length is 200.

Quadruped (Pixel). We use the pixel-based version of the Quadruped environment [31] used in METRA [1]. Specifically, we use the image size of $64 \times 64 \times 3$ with 200 episode length.

402 **Kitchen (Pixel).** We use the pixel-based version of the Kitchen environment [42] used in ME-
 403 TRA [1] and LEXA [10]. Specifically, we use the image size of $64 \times 64 \times 3$ with 50 episode
 404 length.

405 A.5 Evaluation Protocol

406 For Ant, Humanoid, and Quadruped (Pixel), we sample goals with (x, y) -coordinates from $[-50, 50]^2$,
 407 $[-40, 40]^2$, and $[-15, 15]^2$, respectively. For the rest of the goal state (e.g. joint poses), we use the
 408 initial robot configuration following Park et al. [1].

409 For HalfCheetah, we sample goals with x -coordinates from $[-100, 100]$.

410 For AntMaze-Large and AntMaze-Ultra, we use the pre-defined goals in HARDEST_MAZE_EVAL and
 411 ULTRA_MAZE_EVAL from the official code of D4RL and AntMaze-Ultra. Goal locations are shown in
 412 Figure 7. A goal is deemed to be reached when an ant gets closer than 0.5 to the goal.

413 For Kitchen, we use the same 6 single-task goal images used in LEXA [10], which has the interaction
 414 of Kettle, Microwave, Light switch, Hinge cabinet, Slide cabinet, and Bottom burner. We report the
 415 total number of achieved goals during evaluation as *goal success*.

416 For all environments, we use a full state as a goal. Specifically, for state-base observations, we use the
 417 observation upon reset as the base observation, and switch the x, y coordinates (or x for HalfCheetah)
 418 to the right dimensions. For Quadruped (Pixel), we render the image of the state where the agent is at
 419 the goal position, and use it as the goal.

420 B Main Results with 5 Seeds

421 In Figure 9 and Figure 10, we report the state coverage and goal-reaching metrics with 5 seeds.
 422 Figure 9 shows the consistent results with 3-seed results in Figure 4 of the main paper. For Humanoid-
 423 Run, we can see a much clearer difference between METRA and ours with 5 seeds. We can also
 424 observe the consistent results in *goal distance* and *goal success* (i.e. # goal achieved), as can be seen
 425 in Figure 10. Overall, we show that our method, TLDR, can achieve better state coverages and goal
 426 reaching metrics in most of the environments.

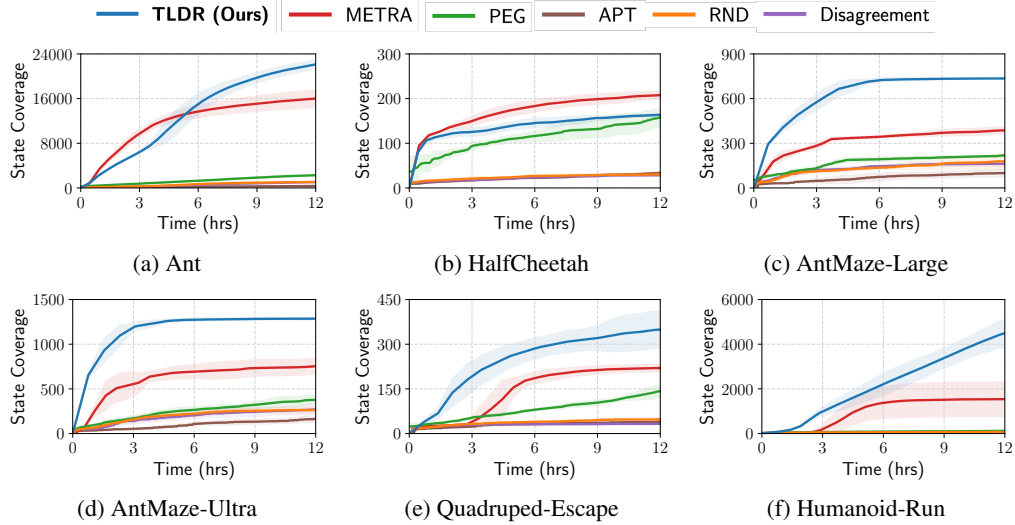


Figure 9: **State coverage on state-based environments with 5 seeds.** We compare the state coverage of unsupervised exploration methods. Our method shows better state coverage than other methods, except in HalfCheetah against METRA.

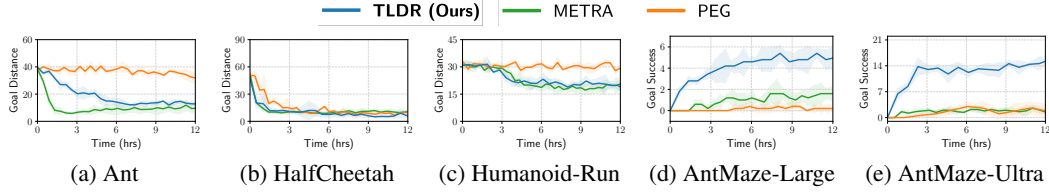


Figure 10: **Goal distance and goal success of a goal-conditioned policy with 5 seeds.** We first report the mean distance between goals and the last states of the evaluation episodes in Ant, HalfCheetah, and Humanoid-Run. TLDR achieves comparable average **goal distance (lower is better)** to METRA in (a-c). For AntMaze environments, we report the number of pre-defined goals reached by a goal-reaching policy (7 for AntMaze-Large and 21 for AntMaze-Ultra). TLDR significantly outperforms prior works on **goal success (higher is better)** in (d-e).

427 C Pixel-based Environment Results

428 Figure 11 shows the results of pixel-based experiments with 5 seeds. In Quadruped (Pixel), TLDR
 429 can explore diverse regions, but learns slower compared to LEXA and METRA. For Kitchen (Pixel),
 430 we additionally report the Queue State Coverage, which is computed as the total number of objects
 431 interacted at least once during the last 100000 environment steps. TLDR is more stable in maintaining
 432 the interactions for each object during training, but achieves lower success rates when the pre-defined
 433 goals are given. We suspect that the temporal abstraction is harder in pixel observations than state
 434 inputs. This may make ϕ not generalize well to the pre-defined goals when they are out-of-distribution,
 435 resulting in the inferior learning of the goal-reaching policy.

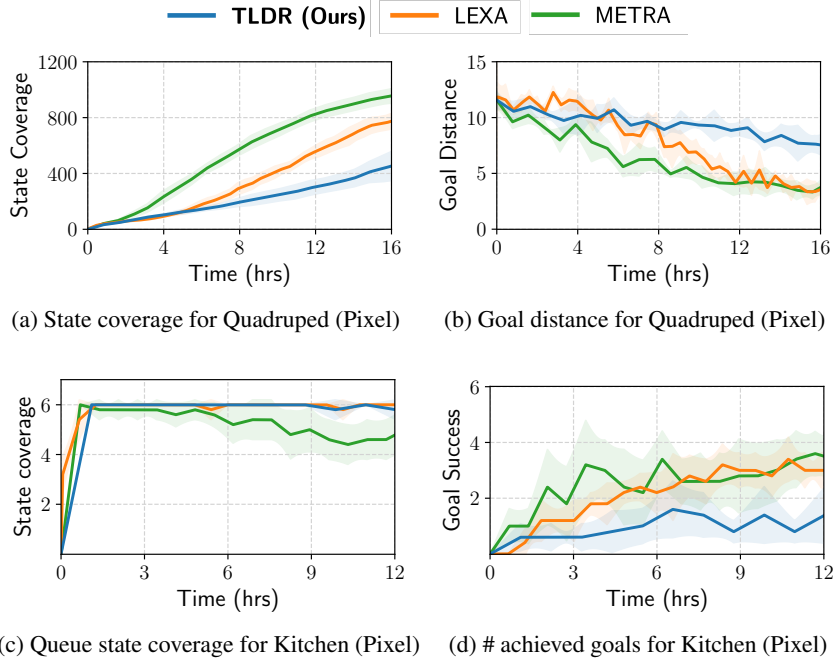


Figure 11: **Experimenting TLDR in pixel-based environments.** We compare TLDR with METRA in Quadruped-Escape and Kitchen with pixel observations. TLDR can explore in those tasks, but the learning speed is much slower compared to METRA.

436 D More Ablation Studies

437 We conduct the ablation studies on the number of nearest neighbors k (Figure 12), $\dim \phi(s)$ (Fig-
 438 ure 13) used in Equation (2), different exploration schemes (Figure 14), and GCRL algorithms
 439 (Figure 15). Also, we compare our GCRL algorithm with dense TLDR-based rewards (Eq. (4)) with
 440 HER and QRL in AntMaze environments.

441 Figure 12 and Figure 13 show that our method is having nearly the same performance across
 442 different values of number of nearest neighbors k in Eq. (1) and $\dim \phi(s)$. This indicates that TLDR
 443 is relatively robust to specific choice of the hyperparameters and does not require sophisticated
 444 hyperparameter tuning. Figure 14 and Figure 15 show that our TLDR representations are important
 445 for both exploration and GCRL.

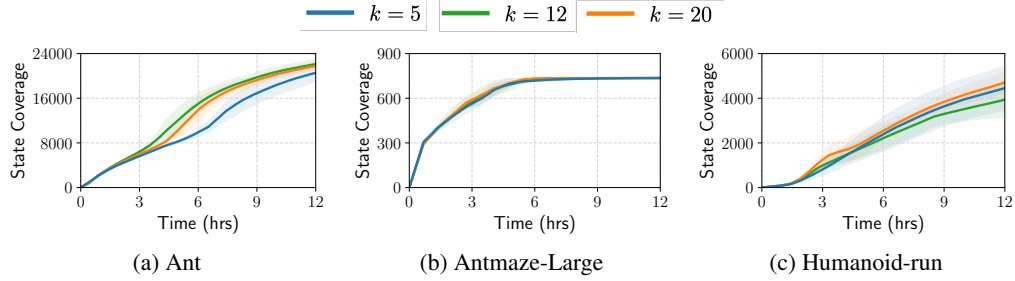


Figure 12: **State coverage on state-based environments with different k .** We measure the state coverage of our method with $k = 5, 12$ (ours), 20, which is used to calculate the TLDR reward for the exploration policy. The results show that k does not have a critical impact on the performance.

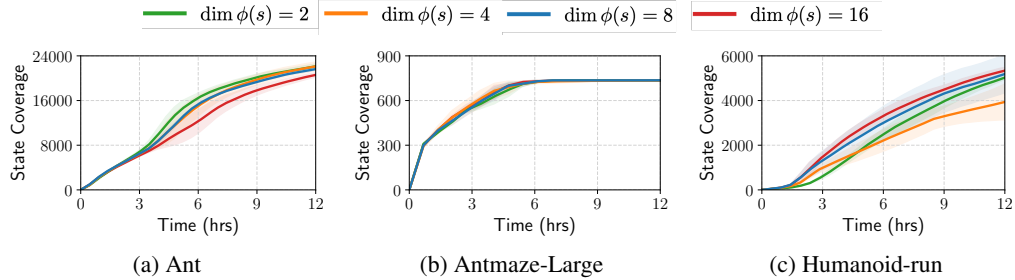


Figure 13: **State coverage on state-based environments with different $\dim \phi(s)$.** We measure the state coverage of our method with $\dim \phi(s) \in \{2, 4, 8, 16\}$, where $\dim \phi(s)$ is the dimension of the TLDR representations. The results show that $\dim \phi(s)$ does not have a critical impact on the performance.

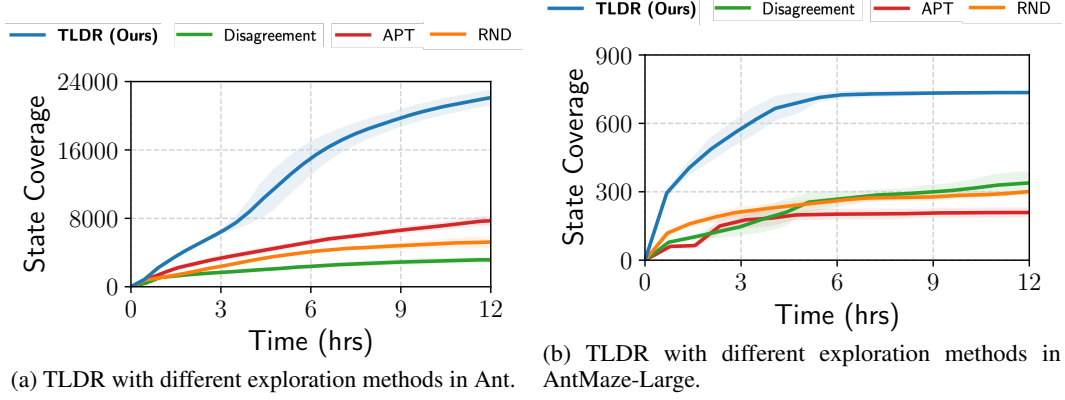


Figure 14: **Impact of temporal distance-aware representations for exploration.** We evaluate our method with different design choices for exploration methods on Ant and AntMaze-Large. We compare with RND, APT with ICM [43] representations, and Disagreement for selecting a goal and training an exploration policy. TLDR shows significantly better state coverage than its ablated versions.

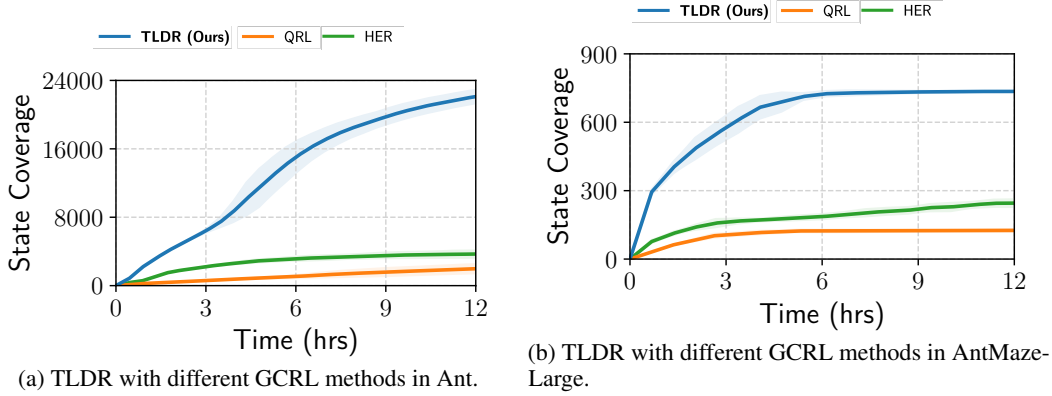


Figure 15: **Impact of temporal distance-aware representations for GCRL.** We evaluate our method with different design choices for GCRL rewards on Ant and AntMaze-Large. For goal-conditioned policy learning, we compare our method with the sparse HER reward and QRL. TLDR significantly improves the performance for prior GCRL approaches, indicating the importance of using temporal distance for GCRL.

446 E More Qualitative Results

447 We include more qualitative results in Figure 16, Figure 17, Figure 18, and Figure 19. For the
 448 qualitative results in Quadruped-Escape (Figure 19), we evenly select 48 states satisfying $x^2 + y^2 =$
 449 10^2 , where x, y is the global x and y coordinates of the agent. z coordinates is selected as the
 450 minimum possible height that the agent do not collide with the terrain. For all environments, TLDR
 451 achieves the best goal-reaching behaviors compared to the other unsupervised GCRL methods,
 452 covering the goals in more diverse regions.

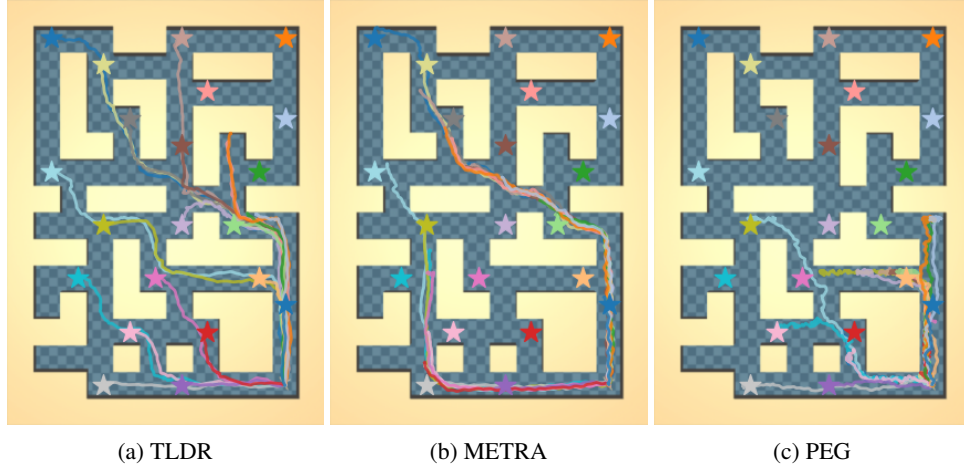


Figure 16: **Goal-reaching ability in AntMaze-Ultra.** TLDR can cover the most number of goals in AntMaze-Ultra compared to other GCRL methods.

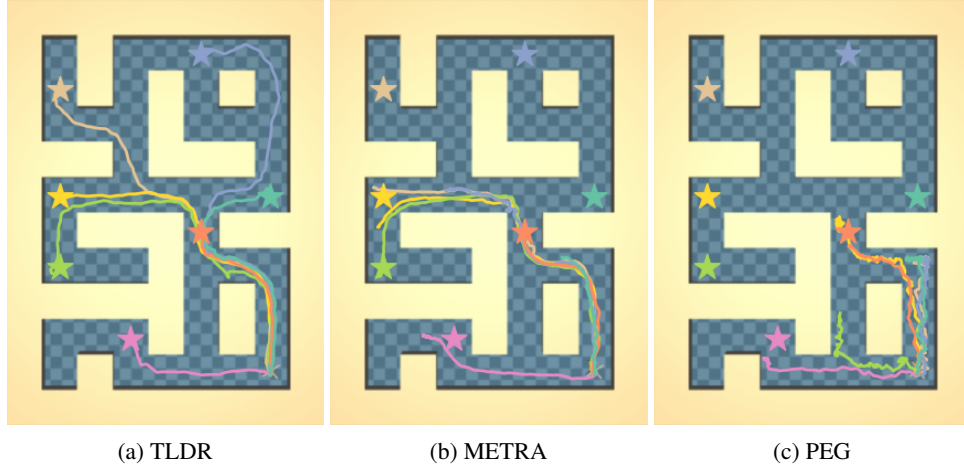


Figure 17: **Goal-reaching ability in AntMaze-Large.** TLDR can cover the most number of goals in AntMaze-Large compared to other GCRL methods.

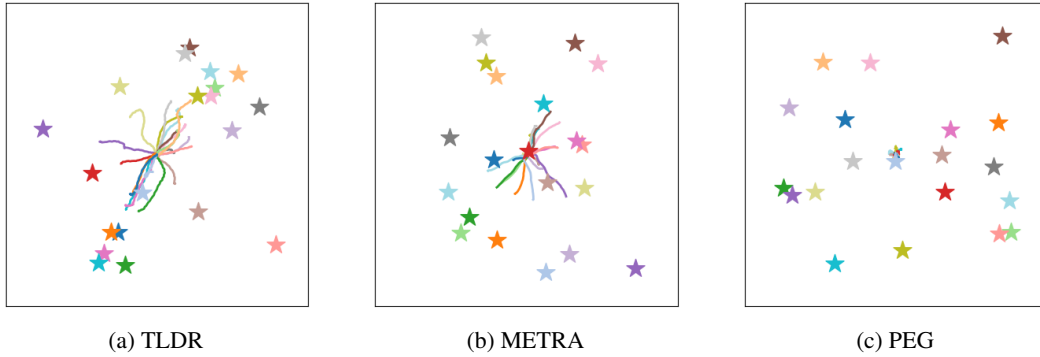


Figure 18: **Goal-reaching ability in Humanoid-Run.** We evaluate each method with the goals sampled according to (Appendix A.5). TLDR tries to move further towards the goal of diverse directions, compared to other methods.

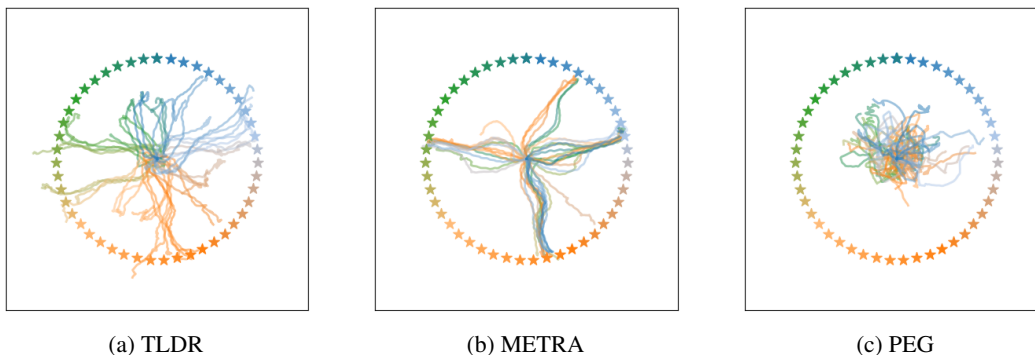


Figure 19: **Goal-reaching ability in Quadrupe-Escape.** When given the goal from the origin, TLDR can not only cover more regions but also have a better goal following ability, compared to METRA.