

A DETAILED DERIVATION OF THE GRADIENTS OF DISCRIMINATOR'S OBJECTIVE FUNCTION

We present the detailed derivation of gradients of the discriminator objective in Eq. (10). Note that we exclude the regularisation terms introduced in Section 6, whose gradients are straightforward to calculate. Since the reward parameter θ is only involved in the first and second terms of $\mathcal{L}_{\text{dis}}(\theta, \phi)$, its gradient w.r.t. θ is calculated by:

$$\begin{aligned}
 \frac{\partial}{\partial \theta} \mathcal{L}_{\text{dis}}(\theta, \phi) &= \frac{\partial}{\partial \theta} \mathbb{E}_{\tau \sim \pi_E^c} [\log D_\theta(\tau)] + \frac{\partial}{\partial \theta} \mathbb{E}_{\tau \sim q_{\omega, \phi}} [\log(1 - \log D_\theta(\tau))] \\
 &= \frac{\partial}{\partial \theta} \mathbb{E}_{\tau \sim \mathcal{D}_E} \left[\log \frac{\exp(f_\theta(\tau))}{\exp(f_\theta(\tau)) + \pi_\omega(\tau)} \right] + \frac{\partial}{\partial \theta} \mathbb{E}_{\tau \sim \mathcal{D}_S} \left[\log \frac{\pi_\omega(\tau)}{\exp(f_\theta(\tau)) + \pi_\omega(\tau)} \right] \\
 &= \mathbb{E}_{\tau \sim \mathcal{D}_E} \left[\frac{\partial}{\partial \theta} f_\theta(\tau) - \frac{\partial}{\partial \theta} \log(\exp(f_\theta(\tau)) + \pi_\omega(\tau)) \right] - \\
 &\quad \mathbb{E}_{\tau \sim \mathcal{D}_S} \left[\frac{\partial}{\partial \theta} \log(\exp(f_\theta(\tau)) + \pi_\omega(\tau)) \right] \\
 &= \mathbb{E}_{\tau \sim \mathcal{D}_E} \left[\left(1 - \frac{\exp(f_\theta(\tau))}{\exp(f_\theta(\tau)) + \pi_\omega(\tau)} \right) \frac{\partial}{\partial \theta} f_\theta(\tau) \right] - \\
 &\quad \mathbb{E}_{\tau \sim \mathcal{D}_S} \left[\frac{\exp(f_\theta(\tau))}{\exp(f_\theta(\tau)) + \pi_\omega(\tau)} \frac{\partial}{\partial \theta} f_\theta(\tau) \right].
 \end{aligned} \tag{14}$$

The feasibility function parameter ϕ is only involved in the second and third terms of $\mathcal{L}_{\text{dis}}(\theta, \phi)$, and thus its gradient w.r.t. ϕ is calculated by:

$$\begin{aligned}
 \frac{\partial}{\partial \phi} \mathcal{L}_{\text{dis}}(\theta, \phi) &= \frac{\partial}{\partial \phi} \mathbb{E}_{\tau \sim q_{\omega, \phi}} [\log(1 - D_\theta(\tau))] - \frac{\partial}{\partial \phi} \mathbb{E}_{\tau \sim \pi_\omega} [\bar{\delta}_\phi(\tau) - \alpha] \\
 &= \frac{1}{|\mathcal{D}_S|} \sum_{i=1}^{|\mathcal{D}_S|} \left[\log(1 - D_\theta(\tau_i)) \frac{\partial}{\partial \phi} q_{\omega, \phi}(\tau_i) \right] - \mathbb{E}_{\tau \sim \mathcal{D}_P} \left[\frac{\partial \bar{\delta}_\phi(\tau)}{\partial \phi} \right] \\
 &= \frac{1}{|\mathcal{D}_S|} \sum_{i=1}^{|\mathcal{D}_S|} \left[\frac{\pi_\omega(\tau_i)}{\exp(f_\theta(\tau_i)) + \pi_\omega(\tau_i)} \frac{\partial}{\partial \phi} \pi_\omega(\tau_i) \bar{\delta}_\phi(\tau_i) \right] - \mathbb{E}_{\tau \sim \mathcal{D}_P} \left[\frac{\partial \bar{\delta}_\phi(\tau_i)}{\partial \phi} \right] \\
 &= \mathbb{E}_{\tau \sim \mathcal{D}_S} \left[\frac{\pi_\omega(\tau)}{1 + \exp(f_\theta(\tau))/\pi_\omega(\tau)} \frac{\partial}{\partial \phi} \bar{\delta}_\phi(\tau) \right] - \mathbb{E}_{\tau \sim \mathcal{D}_P} \left[\frac{\partial}{\partial \phi} \bar{\delta}_\phi(\tau) \right].
 \end{aligned} \tag{15}$$

B MEERKAT DATA PROCESSING

To obtain the meerkat behaviour, two GoPro Max cameras are set on the back wall of the enclosure, one focusing on the replica termite mound in the centre of the enclosure and the other overlooking the foraging area and entrance to the enclosure, which are hubs of activity (Figure 5). For example, the mound is a popular area for guarding behaviour, and the foraging area is popular when meerkats are looking for food. The cameras are set to automatically record videos every 12 minutes, and the contents recorded are filtered, which exclude the fragments that include visitors. Videos with many individuals, social interactions, and other interesting behaviours were selected for the annotation (Figure 6). During the annotation process, the computer vision annotation tool CVAT version 2.3 is utilised to sign the behaviour in the videos. Besides, masking techniques are used to protect the privacy of visitors and maintain the vision information of human activities at the same time. The adult and baby meerkat are annotated specifically in the dataset, with annotators using a small bounding box to note the baby meerkat's positions relative to the adults. Through multiple checks as well as using scripts to automatically detect the error, the accuracy and the consistency of the annotations are ensured (Rogers et al., 2023).

In our research, the dataset is organised according to the unique identifiers of individual meerkats, and every meerkat's behaviour is recorded over different timestamps. Specifically, the information of each

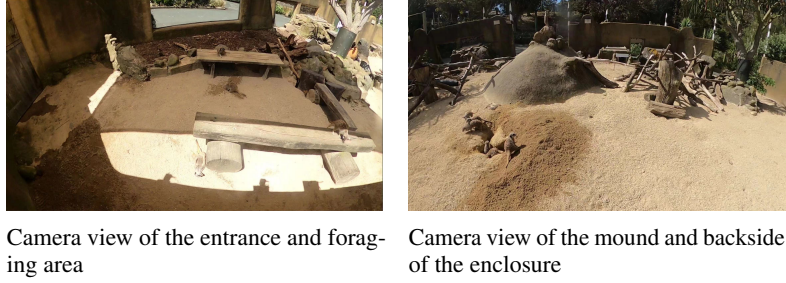


Figure 5: Example images of the camera views.



Figure 6: Examples of the meerkat behaviours.

timestamp includes four different parts: the identifier, the scene the meerkat is located in, the action and the three-dimensional coordinate point. In order to uniform the length of time series for analysis, we process the dataset, retaining only complete sequences of every 30 timestamps as independent trajectories, and delete those with fewer than 30 timestamps. This method can not only simplify the structure of data but also facilitate further analysis. Through this data processing approach, we construct a meerkat dataset that includes both state and action information in each timestamp.

We divide each area based on meerkat’s activity range and labelled each area with a unique colour to distinguish its scope, as shown in Figure 8. After obtaining the Meerkat’s behavioural dataset, we analyse the transition frequency of each area and observe that in certain areas, the activity frequency is particularly high (Figure 7). We are inspired by this to explore whether meerkat’s various behaviours are driven by certain causal constraints.

C CAUSAL STRUCTURE DISCOVERY IN MEERKAT BEHAVIOUR

PCMCI Algorithm is designed to detect and quantify causal relationships in large-scale nonlinear time series datasets (Runge et al., 2019). Combined with the linear or non-linear conditional independence tests and causal discovery algorithm, PCMCI can effectively improve the ability to recognise ground truth causal relationships. For example, in the meerkat behaviour dataset characterised by time series

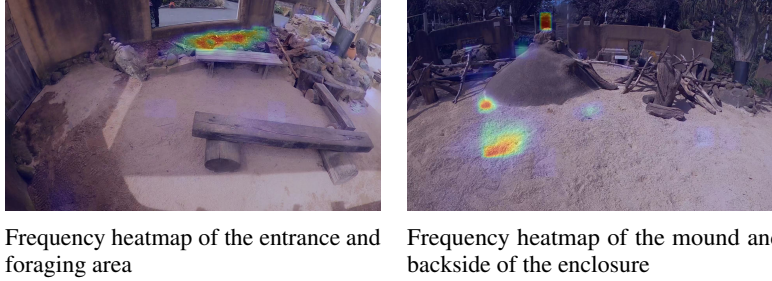


Figure 7: The frequency of meerkat activity in various regions corresponds to the heatmap from the camera perspective. The areas where meerkat is frequently active are highlighted.

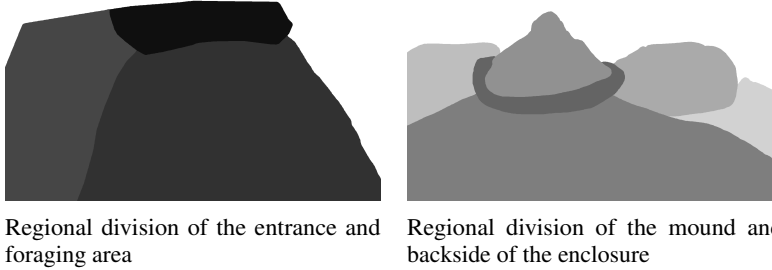


Figure 8: Referring to Figure 2 in the main text, we have labelled blocks of different colours for each area to visually illustrate the division of meerkat activity zones.

data, PCMCI can be used to analyse the causal effects between state transitions, in case to reflect the interaction between states. In this context, behaviour transitions with higher causality might have lower constraints, while those with lower causality could show stronger constraints. This indicates that even if some state transitions offer high rewards, there may be a large cost to take the action.

In the application of PCMCI to analyse the meerkat behaviour dataset, the output is a directed graph of all states, where the colour of each edge represents the causal strength between the starting state and ending state. Considering that there are a total of 25 states, using the directed graph may cause visual confusion and make it difficult to clearly display the relationships between states. Therefore, we select heatmap to present the result of the PCMCI algorithm, and colour variations are used to display the causal strength between different states, therefore allowing a clearer display of causal differences (Figure 9).

D EXPERIMENT SETTINGS

We utilise the open-source library from Gleave et al. (2022), which provides high-quality, reliable, and modular implementations of various reinforcement learning and imitation learning algorithms. Built on Stable Baseline 3 (Raffin et al., 2021), the imitation library offers accurate experimental baselines, allowing us to easily train and compare a range of algorithms. We extend the library by incorporating our algorithm and modifying specific methods related to generative adversarial algorithms to support the implementation of a trajectory-based discriminator as our design.

In addition, we refer to the constrained environments and benchmarking methods designed by Liu et al. (2022) to evaluate our algorithm and baselines based on metrics of discriminator accuracy and constraint violation rate. Each constraint is customly designed to ensure that the agent performs safe and controlled actions within the defined parameters.

Furthermore, we set unique hyperparameters for each environment, optimising the algorithm’s efficiency while avoiding overfitting. All important hyperparameters are listed in Table 2.

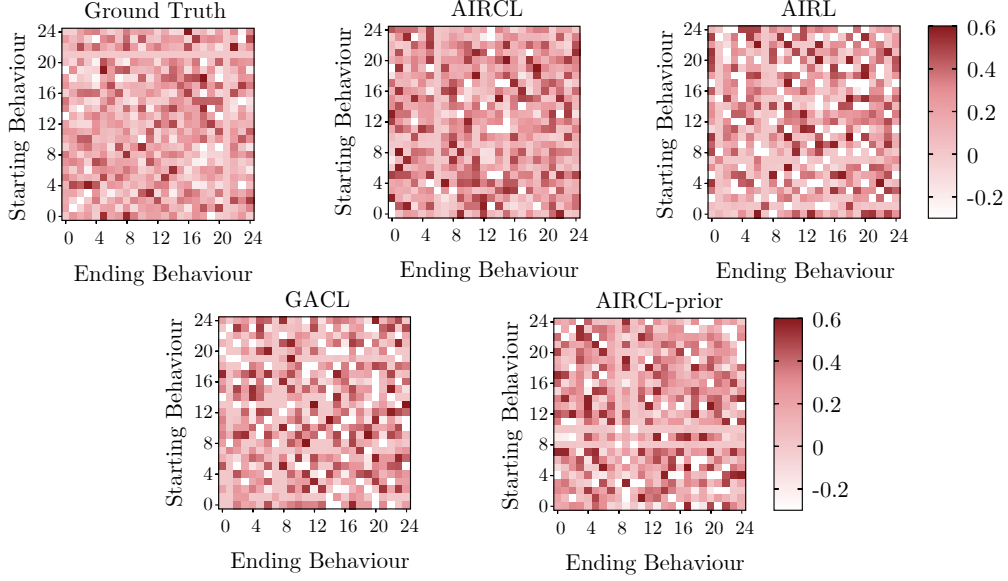


Figure 9: Causal strength for each state transition, as the ground truth constraints. Please note that in our experiments, we recorded the differences between the causal constraints of trajectories generated by each algorithm and the truth constraints.

Table 2: The hyperparameters of each environment, note that hidden units in each layer are reported for network architecture.

	GRIDWORLD	SWIMMER	WALKER	INVERSEPENDULUM	MEERKAT
EXPERT TRAJECTORY	70	50	50	50	2182
SAMPLED TRAJECTORY	70	50	50	50	2182
HORIZON	10	500	500	100	30
REWARD NETWORK	32, 32	32, 32	32, 32	32, 32	32, 32
FEASIBILITY NETWORK	32, 32	32, 32	32, 32	32, 32	32, 32
BATCH SIZE	700	2500	2500	1000	500
LEARNING RATE	0.0005	0.0005	0.0005	0.0005	0.0005
PPO CLIP RANGE	0.1	0.1	0.1	0.1	0.1
COEFFICIENT (φ, κ)	0.001, 0.001	0.001, 0.001	0.001, 0.001	0.001, 0.001	0.001, 0.001