Semantically Grounded Skill Discovery via Vision-Language Models

Ben McDowell

Bachelor of Science in Computer Science and Artificial Intelligence
The University of Bath
2025-2026

Semantically Grounded Skill Discovery via Vision-Language Models

Submitted by: Ben McDowell

Copyright

Attention is drawn to the fact that copyright of this dissertation rests with its author. The Intellectual Property Rights of the products produced as part of the project belong to the author unless otherwise specified below, in accordance with the University of Bath's policy on intellectual property (see `https://www.bath.ac.uk/publications/university-ordinances/attachments/Ordinances_1_October_2020.pdf`).
This copy of the dissertation has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the dissertation and no information derived from it may be published without the prior written consent of the author.

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

Unsupervised skill discovery offers a promising path toward generalist reinforcement learning agents capable of learning useful skills without a reward function. However, existing approaches operate in uninformed representation spaces whose geometry reflects pixel-level statistics rather than semantically meaningful behavioural structure, causing agents to discover skills that are statistically distinguishable but behaviourally trivial. We propose grounding skill discovery in the semantic embedding space of a vision-language model, replacing task-trained convolutional encoders with a frozen CLIP ViT-B/32 encoder pretrained on internet-scale image-text pairs. Because CLIP's contrastive training organises visual observations by semantic similarity, the diversity objective, discriminator, and downstream policy all operate in a space already structured by human-interpretable behavioural content, without any environment-specific annotation or reward engineering. We integrate this representation into two established skill discovery algorithms, DIAYN and APS, and evaluate across twelve downstream locomotion tasks spanning three domains in photorealistic textured environments. APS-CLIP achieves 4.52$\times$ the CNN baseline score on Walker and 1.22$\times$ on Quadruped, while the frozen encoder produces a representation 96% more compact than the CNN baseline and eliminates encoder gradient computation entirely. Our results establish that internet-scale vision-language pretraining provides a useful and computationally efficient semantic prior for unsupervised skill discovery, reducing the representational burden on the skill discovery objective and enabling more semantically grounded behaviour without any task-specific supervision.

Contents

List of Abbreviations **7**

1 Introduction **9**

2 Literature and Technology Survey **12**
2.1 Reinforcement Learning . . . 12
2.1.1 The Challenge of Reward Engineering . . . 13
2.1.2 Unsupervised Reinforcement Learning . . . 13
2.1.3 Temporal Abstraction and Skill Discovery . . . 14
2.1.4 Information-Theoretic Skill Discovery . . . 15
2.1.5 Metric-Based Skill Discovery . . . 16
2.1.6 Aligning Skill Discovery with Human Intent . . . 17
2.2 Vision-Language Models and Semantic Representations . . . 17
2.2.1 CLIP (Contrastive Language-Image Pre-Training) . . . 18
2.3 CLIP in Reinforcement Learning . . . 19
2.3.1 CLIP as a Goal-Reaching Reward Signal . . . 19
2.3.2 CLIP for Exploration . . . 19
2.3.3 Foundation Models as Skill Discovery Guides . . . 19
2.3.4 Language-Grounded Skill Discovery . . . 20
2.3.5 Positioning This Work . . . 20

3 Methodology **21**
3.1 CLIP Integration . . . 21
3.1.1 Encoder Selection . . . 21
3.1.2 Semantic Normalisation and Dimensionality . . . 21
3.1.3 Temporal Context via Frame Stacking . . . 22
3.2 Algorithm Selection, Justification, and Modification . . . 22
3.2.1 DIAYN: Mutual Information and Discriminability . . . 23
3.2.2 APS: Entropy Maximisation and Successor Features . . . 23

4 Experimental Setup **25**
4.1 Experimental Conditions . . . 25
4.2 Environments . . . 25
4.3 Implementation . . . 26
4.4 Evaluation Protocol . . . 26

5 Results and Analysis **28**
5.1 Preliminary Sanity Check: CLIP Encodes Behavioural Structure . . . 28

5.2 Pretraining Results . . . 29
5.3 Finetuning Results . . . 33
5.3.1 Textured Walker . . . 33
5.3.2 Textured Quadruped . . . 33
5.3.3 Textured Cheetah . . . 34
5.3.4 Aggregate Comparison . . . 34
5.4 Qualitative Analysis: Textured Walker . . . 34
5.5 Ablations . . . 36
5.5.1 Pretraining Scalability . . . 37
5.5.2 Effect of Textured Environments . . . 37
5.5.3 Effect of CLIP Model Scale . . . 38
5.5.4 Effect of Frame Stacking . . . 39
5.5.5 Effect of Skill Space Size . . . 40

6 Discussion **42**
6.1 When Does CLIP Improve Over CNN? . . . 42
6.2 Static Skills and the Limits of a Fixed Prior . . . 43
6.3 Limitations of Experimental Setup . . . 44

7 Conclusion and Future Work **46**
7.1 Conclusion . . . 46
7.2 Future Work . . . 46
7.2.1 Beyond Static Skill Discovery: Trajectory-Level Diversity . . . 46
7.2.2 Richer Representations and More Realistic Environments . . . 47
7.2.3 Motion-Aware Encoders . . . 48
7.2.4 Zero-Shot Skill Selection . . . 48

Bibliography **49**

A Derivation of the DIAYN Variational Lower Bound **55**

B Derivation of the APS Exploitation Reward **56**

C Implementation Details and Hyperparameters **58**

D Finetuning Approach for DIAYN and APS **61**

E Reward Function Specifications **63**

F Individual Numerical Results **65**

G Extended Qualitative Analysis **66**

List of Figures

1.1 Pixel L2 distance fails as a semantically meaningful similarity measure for skill discovery, while CLIP cosine distance correctly reflects behavioural equivalence. **Top row:** the same standing quadruped rendered under daytime and near-darkness lighting produces a pixel L2 distance of 21,814 despite being an identical behavioural state; CLIP cosine distance correctly identifies the poses as near-equivalent (0.0356). **Bottom row:** the same pose at different positions in the frame yields a pixel L2 distance of 16,240, while CLIP cosine distance remains low (0.0127). CLIP cosine distance ranges from 0 (identical) to 2 (maximally dissimilar). 10

2.1 **The Reinforcement Learning Interaction Loop.** At each time step i, the agent observes state s_i and reward r_i, executes an action a_i, and receives a next state s_{i+1} and reward signal r_{i+1} from the environment. Figure from Nguyen et al. [2024]. 12

4.1 Comparison of the original DMC environment and the textured variant introduced in this work. 26

5.1 t-SNE projection of CLIP ViT-B/32 embeddings for 1,400 rendered frames across seven manually specified Walker poses, with randomised lighting and agent position. Each colour corresponds to a distinct pose class. The clusters are well-separated, achieving a k-NN classification accuracy of 99%, demonstrating that CLIP organises pixel observations by behavioural semantics rather than low-level visual features. 29

5.2 DIAYN discriminator accuracy during pretraining across three domains. The dashed horizontal line indicates chance performance ($1/16 = 6.25\%$). 31

5.3 Incidental extrinsic reward during pretraining across all four methods and three domains. Solid lines denote DIAYN variants; dashed lines denote APS variants. 32

5.4 APS intrinsic reward components during pretraining. Top row: entropy reward r^{ent}; bottom row: SF reward r^{sf}. 32

5.5 Discovered skill repertoires on Textured Walker after 2M pretraining frames. DIAYN-CNN is shown as a motion composite of three frames (start, midpoint, terminal) with earlier frames faded, revealing trajectory dynamics. All other methods are shown as terminal frames, reflecting the static nature of their discovered skills. CNN skills are evaluated at 500k pretraining frames due to memory constraints. Skills are 0-indexed from top-left to bottom-right. . . . 35

5.6 Downstream performance on Textured Quadruped Stand as a function of pretraining duration for APS-CLIP and APS-CNN. Shaded regions indicate ± one standard deviation across three seeds. 37
5.7 Incidental extrinsic reward during APS-CLIP pretraining on plain versus textured Quadruped. Shaded regions indicate the range across seeds. 38
5.8 Pretraining curves for APS-CLIP on Textured Quadruped comparing ViT-B/32 and ViT-g-14 encoders. Top row: extrinsic reward and total intrinsic reward. Bottom row: entropy reward r^{ent} and SF reward r^{sf}. 39
5.9 APS-CLIP pretraining curves on Textured Quadruped comparing single-frame (frame stack 1) and three-frame (frame stack 3) encodings. From left to right: extrinsic reward, entropy reward r^{ent}, and successor feature reward r^{sf}. 40
5.10 DIAYN discriminator accuracy during pretraining on Textured Quadruped for skill space sizes of 8, 16, and 32. Dashed lines indicate chance performance for each configuration. 41
5.11 Incidental extrinsic reward during pretraining on Textured Quadruped for skill space sizes of 8, 16, and 32. 41

G.1 Final-frame collages of discovered skills on Textured Cheetah after 2M pretraining frames. Each cell shows the terminal state of a single skill rollout. CNN skills are shown at 1M pretraining frames due to memory constraints. Skills are 0-indexed from top-left to bottom-right. 67
G.2 Final-frame collages of discovered skills on Textured Quadruped after 2M pretraining frames. Each cell shows the terminal state of a single skill rollout. CNN skills are shown at 500k pretraining frames due to memory constraints. Skills are 0-indexed from top-left to bottom-right. 68

List of Tables

5.1 Pretraining metrics at final checkpoint. CLIP methods are reported at 2M frames; CNN methods at 500k frames (Walker, Quadruped) and 1M frames (Cheetah) due to memory constraints. All values averaged over the final 10k training steps. Discriminator accuracy reported for DIAYN (chance = 6.25%). Intrinsic reward components and extrinsic reward reported for APS. †CNN Walker and Quadruped pretraining was not run for downstream finetuning; metrics are indicative of representation quality only. 30

5.2 Finetuning performance across three textured locomotion domains. Results report mean episode reward ± standard deviation across three seeds. CNN baselines for Textured Walker and Textured Quadruped are taken from Laskin et al. [2021] using plain, untextured environments; CNN baselines for Textured Cheetah are from our own runs. †CNN Cheetah baselines are reported at 1×10^6 pretraining steps due to memory constraints (Section 4). The highest score within each DIAYN and APS pair is in **bold**. Aggregate improvement ratios are computed as the mean of per-task score ratios APS-CLIP$_i$/APS-CNN$_i$ across the four downstream tasks within each domain; APS-CLIP achieves 4.52× on Textured Walker and 1.22× on Textured Quadruped. CLIP-based methods underperform CNN on all Textured Cheetah tasks. 33

5.3 Summary of ablation studies. All finetuning runs are averaged across 3 seeds unless otherwise stated. 36

C.1 Hyperparameters used in all experiments (excluding ablations). CNN values follow the URLB pixel-based configuration [Laskin et al., 2021]. CLIP values match the CNN configuration except where explicitly justified; differences are in bold. 58

C.2 Algorithm-specific hyperparameters for DIAYN and APS, shared across CNN and CLIP variants unless otherwise stated. 59

F.1 Per-seed fine-tuning performance after 2×10^6 pre-training steps. S1, S2, S3 denote individual random seeds. The highest mean score for each task is shown in bold. CNN baselines for Textured Walker and Textured Quadruped are taken from Laskin et al. [2021] and per-seed results are therefore unavailable (—). CNN baselines for Textured Cheetah are from our own runs. Aggregate mean and standard deviation are reported in Table 5.2. 65

List of Abbreviations

Abbreviation	Definition
APS	Active Pre-training with Successor Features
CNN	Convolutional Neural Network
DDPG	Deep Deterministic Policy Gradient
DIAYN	Diversity is All You Need
DLSD	Discovery Skills With Language
DMC	Deep Mind Control Suite
DoF	Degrees of Freedom
LSD	Lipschitz-constrained Skill Discovery
LGSD	Language Guided Skill Discovery
MDP	Markov Decision Process
MI	Mutual Information
MLP	Multi-Layer Perceptron
RL	Reinforcement Learning
SF	Successor Feature
URL	Unsupervised Reinforcement Learning
URLB	Unsupervised Reinforcement Learning Benchmark
VLM	Vision-Language Model

Acknowledgements

I would like to express my sincere thanks to my supervisor, Callum Pitceathly, and my second marker, Andreas Theophilou, for their guidance and support throughout this project.

I am grateful to the University of Bath for access to the Hex computing cluster, which provided the GPU resources necessary to conduct the experiments in this dissertation, and to Weights and Biases for their academic plan, which provided 200GB of free storage for experiment tracking, video evaluations and model checkpointing.

I am also grateful to my friends, from Bath and beyond, for their constant support during the more challenging moments, and for celebrating the rewarding ones with me.

Above all, I thank my family for their unwavering support, patience, and belief in me. None of this would have been possible without them.

This journey would not have been the same without all of you, and I am deeply grateful to have shared it with such wonderful people.

Chapter 1

Introduction

Unsupervised pretraining has emerged as one of the key ingredients behind recent breakthroughs in computer vision and language modelling [Brown et al., 2020, Devlin et al., 2019, Caron et al., 2021]. By training on large datasets of unlabelled data, models acquire generalisable representations that transfer efficiently to a broad range of downstream tasks. Transferring this paradigm to reinforcement learning (RL), producing agents that autonomously explore their environment and acquire reusable behaviours without task-specific reward, remains an open question of central importance to the field [Laskin et al., 2021].

Unsupervised skill discovery (USD) [Eysenbach et al., 2018, Liu and Abbeel, 2021a, Park, Rybkin and Levine, 2024] offers a structured approach to this problem. In the pretraining paradigm explored in this work, the agent autonomously acquires a repertoire of diverse, reusable behaviours in a reward-free phase, each associated with a latent skill vector; a brief finetuning phase then adapts this repertoire to a specific downstream task using extrinsic reward [Laskin et al., 2021]. The quality of the pretrained skill repertoire directly determines downstream adaptation efficiency, making the diversity and semantic meaningfulness of discovered skills a central concern.

A fundamental limitation of existing skill discovery methods is their reliance on uninformed state representations. In pixel-based settings, task-trained convolutional encoders are shaped entirely by the skill discovery objective with no prior knowledge of what constitutes a meaningful distinction between states. Skills may therefore partition the state space by lighting conditions or background texture rather than by locomotion mode, appearing diverse in feature space while remaining behaviourally trivial. For USD to serve as a viable pretraining backbone for generalist agents, skills must be semantically grounded by construction rather than by accident. An agent whose repertoire reflects irrelevant visual variation will adapt poorly to downstream tasks regardless of how efficiently finetuning operates, undermining the broader promise of unsupervised RL.

Recent advances in vision-language pretraining [Bommasani et al., 2022] offer a principled solution. CLIP [Radford et al., 2021] is a vision-language model (VLM) trained on internet-scale image-text pairs via a contrastive objective, producing embeddings structured by semantic similarity. States that differ only in background appearance are mapped to nearby embeddings, while behaviourally distinct states (e.g., a standing agent versus a fallen one) are separated in a way that reflects human-interpretable semantics, thus making CLIP cosine distance a principled proxy for behavioural dissimilarity, as illustrated in Figure 1.1.

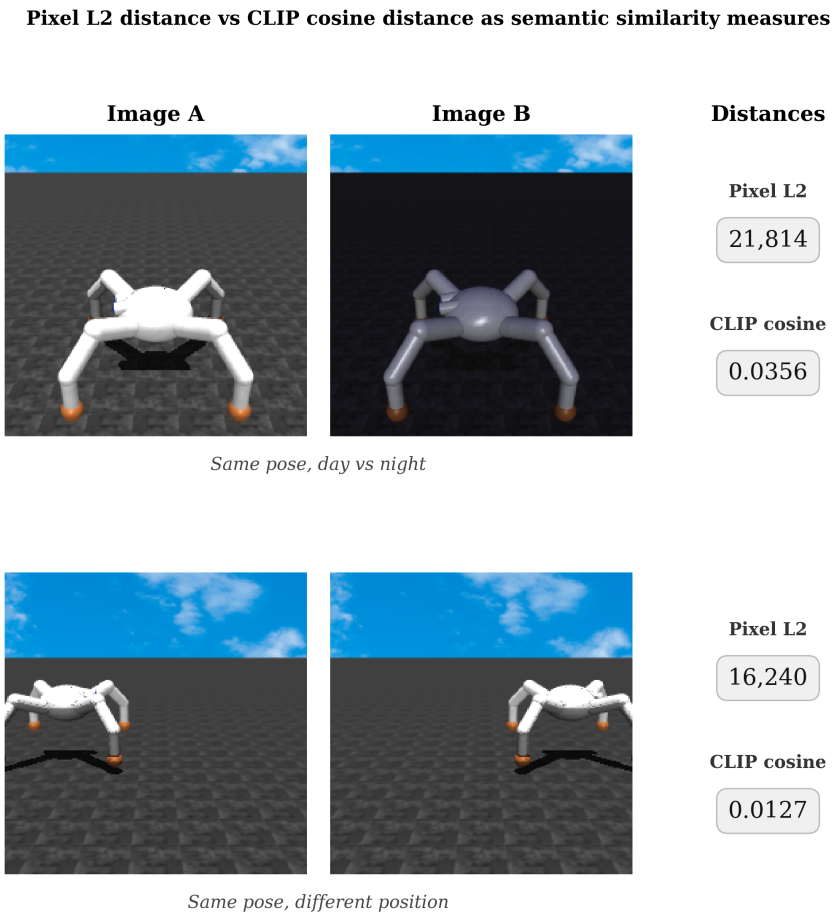


Figure 1.1: Pixel L2 distance fails as a semantically meaningful similarity measure for skill discovery, while CLIP cosine distance correctly reflects behavioural equivalence. **Top row:** the same standing quadruped rendered under daytime and near-darkness lighting produces a pixel L2 distance of 21,814 despite being an identical behavioural state; CLIP cosine distance correctly identifies the poses as near-equivalent (0.0356). **Bottom row:** the same pose at different positions in the frame yields a pixel L2 distance of 16,240, while CLIP cosine distance remains low (0.0127). CLIP cosine distance ranges from 0 (identical) to 2 (maximally dissimilar).

This dissertation investigates whether replacing task-trained CNN encoders with a frozen CLIP encoder improves the semantic diversity of skills discovered by USD methods in pixel-based settings. We extend two established algorithms from the URLB benchmark [Laskin et al., 2021], DIAYN [Eysenbach et al., 2018] and APS [Liu and Abbeel, 2021a], by substituting their visual encoders with frozen CLIP and evaluating the resulting skill repertoires on twelve downstream locomotion tasks across three domains. The central hypothesis is that CLIP's semantic structure provides a knowledge-grounded prior over the latent space, enabling skill diversity objectives to discover human-aligned behaviours from pixel observations without annotation, reward engineering, or environment-specific prompt design.

The primary contributions of this work are as follows.

- We propose and evaluate a unified framework for integrating frozen CLIP encoders into the skill discovery loop, applicable across mutual information (MI) and successor feature (SF) objectives. To our knowledge this is the first work to integrate a frozen vision-

language encoder directly as the primary state representation in the skill discovery loop, requiring no symbolic state access, human supervision, or generative model inference.

- We provide an empirical evaluation on URLB comparing CLIP-based variants of DIAYN and APS against CNN baselines across three locomotion domains. Our experiments show that CLIP features are beneficial when the relevant variation between skills is postural and visually salient, with APS-CLIP substantially outperforming its CNN counterpart on Walker and Quadruped locomotion tasks, but provide little advantage on temporally dynamic tasks where motion rather than posture is the discriminating feature.
- We present a comprehensive ablation study examining the effect of background texture, encoder scale, skill space size, frame stacking, and pretraining duration, identifying the conditions under which semantic grounding via CLIP is and is not beneficial.

The remainder of this dissertation is structured as follows. Chapter 2 reviews the relevant background in unsupervised RL, skill discovery, and VLMs, building the theoretical case for semantic grounding and positioning this work with respect to prior methods. Chapter 3 details the CLIP integration and its design decisions. Chapter 4 describes the experimental setup. Chapter 5 reports and analyses results across pretraining and finetuning phases. Chapter 6 discusses the implications and limitations of the findings. Chapter 7 concludes and identifies the most promising directions for future work.

Chapter 2

Literature and Technology Survey

2.1 Reinforcement Learning

Reinforcement Learning (RL) [Sutton and Barto, 2018] is a subset of Artificial Intelligence concerned with how agents ought to take actions in an environment to maximise a cumulative reward. At its core, the reward signal is a scalar feedback value provided by the environment which indicates the immediate desirability of the agent's state. The agent's objective is not just to maximise this immediate signal but to maximise the expected sum of future rewards (the return).

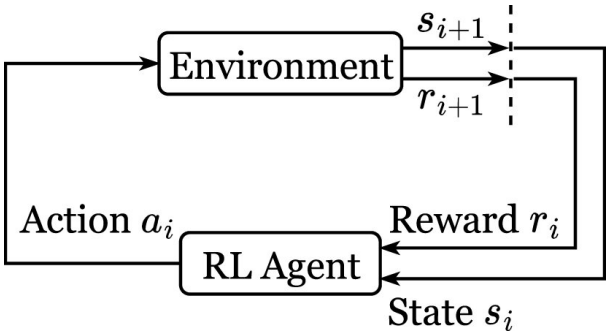


Figure 2.1: **The Reinforcement Learning Interaction Loop.** At each time step i, the agent observes state s_i and reward r_i, executes an action a_i, and receives a next state s_{i+1} and reward signal r_{i+1} from the environment. Figure from Nguyen et al. [2024].

As illustrated in Figure 2.1, the standard formalism for this interaction is the Markov Decision Process (MDP), defined as a tuple $\mathcal{M} = (S, A, P, R, \gamma, \rho_0)$, where:

- S is the set of states,
- A is the set of actions,
- $P(s'|s, a)$ is the transition probability distribution,
- $R(s, a)$ is the reward function,
- $\gamma \in [0, 1)$ is the discount factor,
- $\rho_0(s)$ is the initial state distribution.

The goal of the agent is to learn a policy $\pi(a|s)$ that maximises the expected return $E_{s_0 \sim \rho_0}[\sum_{t=0}^{\infty} \gamma^t r_t]$.

2.1.1 The Challenge of Reward Engineering

RL has made significant strides in solving complex tasks where reward signals arise naturally from the environment, such as strategic games [Schrittwieser et al., 2020, Badia et al., 2020] where the score or win condition serves as a direct training signal. However, in domains lacking such natural feedback, such as robotic manipulation [Gu et al., 2016], practitioners must resort to manual engineering of *dense extrinsic reward functions* that provide frequent feedback to guide the agent incrementally toward the goal. For example, rather than rewarding a robot only when it successfully opens a door, a shaped function might reward the robot for approaching the handle, grasping it, and turning it.

However, manually defining these functions is impractical for several reasons: (i) they are often brittle and domain-specific, failing to generalise to new agents or environments [Shah et al., 2019, di Langosco et al., 2021]; (ii) accurate specification of human preferences is notoriously difficult, often resulting in "reward hacking" where the agent exploits loopholes to maximise the metric rather than the intended objective [Amodei et al., 2016, Shah et al., 2019, Turner, Ratzlaff and Tadepalli, 2020]; and (iii) in many real-world settings, such shaped rewards are simply unavailable [Christiano et al., 2023] (e.g., a humanoid robot attempting to stand up from a fall, where defining intermediate progress towards "upright" is ambiguous). Consequently, agents must rely on *sparse rewards*, where feedback is only received once countless decisions have been made. This creates a challenging "needle-in-a-haystack" exploration problem that significantly hinders learning.

This difficulty is well-captured by Polanyi's Paradox [Polanyi, 2009], which states, "We can know more than we can tell", implying that while humans can recognise a clean table or a scrambled egg, codifying this tacit knowledge into a mathematical reward function based on raw sensor data is often intractable. This motivates *unsupervised RL*, where an agent must learn meaningful behaviours by interacting with the environment *without* a manually crafted reward function.

2.1.2 Unsupervised Reinforcement Learning

The aforementioned supervised approaches stand in contrast to the way intelligent creatures quickly adapt to new tasks by leveraging previously acquired behaviours. *Intrinsic* motivation [Oudeyer and Kaplan, 2008, Stout, Konidaris and Barto, 2005], internally generated rewards that drive exploration independently of any downstream task objective, offers a more naturalistic alternative.

Attempts to define intrinsic motivation objectives can generally be divided into two broad categories [Aubret, Matignon and Hassas, 2019]. The first is *knowledge acquisition*, where the agent strives to find new knowledge about its environment through novelty-seeking, prediction error, or information gain [Liu and Abbeel, 2021b, Pathak et al., 2017, Burda et al., 2018]. The second is *skill learning*, where the agent constructs task-independent and reusable behaviours that can be repurposed for downstream tasks [Eysenbach et al., 2018, Liu and Abbeel, 2021a, Park et al., 2022]. This dissertation is concerned with the second category.

Within the RL literature, methods of this type are most commonly referred to as *unsupervised*

skill discovery (USD) algorithms. URLB [Laskin et al., 2021], whose codebase this work extends, refers to the same class of methods as *competence-based* algorithms, a term drawn from the broader intrinsic motivation taxonomy of Oudeyer and Kaplan [2008]. These terms are treated as synonymous throughout this dissertation, with *unsupervised skill discovery* preferred as it is more descriptive of the objective.

This dissertation adopts the two-phase evaluation protocol formalised by URLB: a long reward-free pretraining phase in which the agent acquires a skill repertoire through intrinsic motivation alone, followed by a brief finetuning phase in which those skills are adapted to a specific downstream task using extrinsic reward.

More concretely, USD builds on the idea of *empowerment* [Klyubin, Polani and Nehaniv, 2005], where the agent ignores the downstream task during pretraining and instead optimises a diversity objective to differentiate behaviours, creating a repertoire of skills available for future use. The latent variable that indexes these skills, typically a discrete code or continuous vector sampled from a fixed prior, must be made to correspond to behaviourally distinct outcomes in the environment, either by learning this correspondence implicitly through a training objective, or by encoding it directly through the variable's position in a structured space. The choice of how to formalise this correspondence is what distinguishes the major families of USD methods, surveyed in the following sections. Across all of them, a shared requirement holds: to learn meaningful skills in complex environments, agents cannot operate solely at the level of primitive motor commands. They require *temporal abstraction*.

2.1.3 Temporal Abstraction and Skill Discovery

To solve complex, long-horizon tasks without dense reward signals, agents need the ability to reason, plan, and act over multiple time scales.

The Options Framework

The canonical formalism for this abstraction is the *options framework* [Sutton and Barto, 2018]. While a primitive action executes for a single time step, an option ω corresponds to a closed-loop policy that governs behaviour over a variable duration.

Formally, an option ω is defined as a tuple $(I_\omega, \pi_\omega, \beta_\omega)$ consisting of:

- **Initiation set** $I_\omega \subseteq S$, specifying the states in which the option may be initiated;
- **Intra-option policy** $\pi_\omega(a \mid s)$, which governs action selection while the option is active;
- **Termination condition** $\beta_\omega(s) \in [0, 1]$, defining the probability of termination upon reaching state s.

Learning Options via Latent Variables

To automate the discovery of options, modern approaches condition the policy on a latent skill variable $z \sim p(z)$ [Haarnoja et al., 2018, Shi, Lim and Lee, 2022]. Instead of manually defining initiation sets and termination conditions, the agent learns a skill-conditioned policy $\pi(a \mid s, z)$, where different values of z should produce clearly distinct trajectories in the environment. This connects the *formalism* of options with the *objective* of skill discovery: the latent variable z acts as the index for an option, and the diversity objective ensures each option performs a

distinct, temporally extended behaviour. How "distinct" is formalised mathematically is what separates the two major families of USD methods:

- **Information-Theoretic Approaches:** distinguishability is defined via MI between z and the resulting states, operationalised through predictability objectives.
- **Metric-Based Approaches:** distinguishability is defined via geometric distance between skill trajectories in a representation space, such that z encodes a direction or displacement rather than merely labelling a behaviour.

2.1.4 Information-Theoretic Skill Discovery

Information-theoretic approaches to USD operate on the principle that for a set of skills to be useful, they must be distinct. This distinctness is mathematically formalised as maximising the MI between the latent skill variable Z and the agent's observation S. Two decompositions of this objective have been explored in the literature, leading to structurally different methods.

Reverse MI. The reverse decomposition maximises $H(Z)$ while minimising $H(Z|S)$. A variational approximation $q_\phi(z|s)$ is introduced for the intractable true posterior, yielding the tractable intrinsic reward $r(s, z) = \log q_\phi(z \mid s) - \log p(z)$ (see Appendix A for the derivation). The agent is rewarded for reaching states where the discriminator can confidently identify the current skill.

VIC [Gregor, Rezende and Wierstra, 2016] applies this objective to terminal states; DIAYN [Eysenbach et al., 2018] extends it to every visited state, removing dependence on fixed initial conditions. Building on this, VALOR [Achiam et al., 2018] decodes the skill from complete trajectories rather than individual states to encourage learning continuous dynamic modes, while VISR [Hansen et al., 2020] structures the discriminator to output controllable features that are linear with respect to a task vector, enabling fast task inference via SF.

Despite these extensions, reverse MI objectives inherently favour skills that are *instantaneously distinguishable*. Because $H(Z)$ is fixed by the prior $p(z)$, the optimisation is dominated by the discriminator's ease of classification, causing agents to converge to static or quasi-static configurations that are easy to classify but lack dynamic complexity [Hansen et al., 2020].

Forward MI. To mitigate the static bias, an alternative decomposition explicitly optimises state entropy:

$$I(S; Z) = H(S) - H(S \mid Z) \tag{2.1}$$

Unlike the reverse form where $H(Z)$ is constant, $H(S)$ here depends on the agent's behaviour and is actively maximised, encouraging broad state-space coverage. $H(S \mid Z)$ encourages task-conditioned predictability. EDL [Campos et al., 2020] applies this to the state distribution to discover skills that cover the state space uniformly. DADS [Sharma et al., 2020] instantiates it over transition dynamics by maximising $I(S_{t+1}; Z \mid S_t)$, encouraging temporally coherent skills, though this requires a skill-conditioned dynamics model that is difficult to stabilise in high-dimensional environments. CIC [Laskin et al., 2022] uses a contrastive estimator for the $H(S|Z)$ term.

APS [Liu and Abbeel, 2021a] adopts the same forward decomposition and jointly optimises

both terms via particle-based entropy estimation from APT [Liu and Abbeel, 2021b]:

$$r_{\text{APS}}^{\text{exploration}}(s, a, s') = \log\left(1 + \frac{1}{k}\sum_{h^{(j)}\in\mathcal{N}_k(h)} \|h - h^{(j)}\|_{n_h}^{n_h}\right), \quad h = f(s') \tag{2.2}$$

where $\mathcal{N}_k(h)$ denotes the k nearest neighbours of the encoded state h in the replay buffer. The conditional entropy $-H(S \mid Z)$ is lower-bounded variationally via SF [Hansen et al., 2020] (see Appendix B), yielding an exploitation reward $r_{\text{APS}}^{\text{exploitation}}(s, a, s', w) = \phi(s')^\top w$, where w is a randomly sampled task vector and $\phi(s')$ is an ℓ_2-normalised state feature.[1] The full intrinsic reward combines both terms:

$$r_{\text{APS}}(s, a, s', w) = \underbrace{\phi(s')^\top w}_{\text{exploitation}} + \underbrace{\log\left(1 + \frac{1}{k}\sum_{h^{(j)}\in\mathcal{N}_k(h)} \|h - h^{(j)}\|_{n_h}^{n_h}\right)}_{\text{exploration}} \tag{2.3}$$

APS also addresses the rigidity of discrete skill representations. Methods that index skills with a categorical variable are constrained to whichever of their K pretrained skills best matches the downstream task, with no ability to interpolate between them. APS instead conditions the policy on a task vector w drawn continuously from the d_{SF}-dimensional unit sphere, meaning the pretrained policy covers a dense, continuous family of behaviours rather than K fixed ones. The probe exploits this by solving a least-squares regression between SF vectors and extrinsic rewards across probe transitions, identifying the direction w^* most correlated with the downstream reward, which can be any linear combination of the pretrained behavioural axes, making the probe considerably more expressive than a discrete argmax over a fixed skill set.

However, exploration bonuses address only the coverage problem, not the underlying metric-agnosticism. MI-based objectives remain indifferent to the distances travelled between states regardless of decomposition, which is why both reverse and forward MI methods frequently converge to static behaviours [Park et al., 2022, Laskin et al., 2022]. This motivates a qualitatively different family of approaches.

2.1.5 Metric-Based Skill Discovery

Rather than augmenting MI objectives with exploration bonuses, *metric-based* methods promote diversity by driving skills toward geometrically distant regions of the state space. METRA [Park, Rybkin and Levine, 2024] formalises this family under the Wasserstein Dependency Measure (WDM):

$$I_{\mathcal{W}}(S; Z) = \mathcal{W}_1(p(s, z),\ p(s)p(z)) \tag{2.4}$$

where $\mathcal{W}_1$ is the 1-Wasserstein distance on the metric space $(\mathcal{S} \times \mathcal{Z}, d)$. Unlike MI, which is agnostic to the underlying distance metric, WDM is metric-aware: maximising it not only discovers diverse skills but actively maximises distances d between skill trajectories. The choice of d is therefore critical, and distinguishes the main methods in this family. LSD [Park et al., 2022] uses Euclidean distance in the raw state space, producing dynamic skills but treating all state dimensions equally. CSD [Park et al., 2023] learns a controllability-aware distance that assigns larger values to harder-to-achieve transitions, progressively directing the agent toward more complex behaviours. METRA [Park, Rybkin and Levine, 2024] defines d via

[1] The original paper defines the exploitation reward using the current state $\phi(s)^\top w$; we follow the URLB codebase implementation [Laskin et al., 2021], which applies it to the next state $\phi(s')^\top w$.

temporal reachability, separating skills by the minimum number of environment steps required to transition between them rather than by visual or physical distance.

Despite these advances, all three methods measure diversity at the level of terminal states rather than trajectories, meaning cyclic behaviours such as a backflip that returns to the starting pose receive little reward regardless of the semantic richness of the intermediate trajectory [Rho et al., 2025b].

More broadly, whether maximising MI or geometric distance, existing methods operate in uninformed representation spaces in which diversity is rewarded without regard to semantic meaning, yielding skills that are mathematically distinct yet behaviourally trivial [Kim et al., 2022]. Addressing this gap requires introducing an explicit notion of semantic preference into the skill discovery process.

2.1.6 Aligning Skill Discovery with Human Intent

A natural approach to ensuring discovered skills are meaningful is to incorporate human feedback into training, either through explicit skill preferences [Wang et al., 2021, Hussonnois, Karimpanal and Rana, 2025], safety constraints [Kim et al., 2022], or expert demonstrations [Rho et al., 2025b, Kim et al., 2025]. However, all such approaches require a human to remain actively involved during training, which is expensive, sample-inefficient, and difficult to scale. More fundamentally, any form of human supervision contradicts the unsupervised assumption underlying skill discovery and risks restricting behavioural diversity to the subset of behaviours the human anticipated.

Semantic Structure as a Medium for Skill Preferences. Language offers a more scalable route to the same goal, encoding human semantic knowledge without requiring active involvement during training. RL works that directly incorporate language offer evidence that this semantic structure improves skill discovery. Tam et al. [2022] showed that language abstractions improve intrinsic exploration by providing a more efficient and interpretable space than raw state representations. Mu et al. [2022] demonstrated that language-conditioned agents maintain exploration efficiency as environment scale grows by collapsing visually distinct but semantically identical states, a property that purely state-based methods lose. Yu, Goodman and Mu [2023] further found that language generalises more effectively than expert demonstrations as task complexity increases. Taken together, these results suggest that the semantic structure language provides, a space in which behaviourally equivalent states are close and behaviourally distinct states are far apart, is the key ingredient for meaningful skill discovery. With the emergence of VLMs pretrained on internet-scale image-text pairs [Bommasani et al., 2022], the question arises whether this structure can be accessed directly from visual observations, inheriting the benefits of language-scale training without requiring language generation at inference time. This motivates the approach taken in this work.

2.2 Vision-Language Models and Semantic Representations

The gap identified at the end of Section 2.1.5 is fundamentally a representation problem. Skill discovery objectives are only as semantically meaningful as the space in which they operate: an agent rewarded for diversity in an uninformed pixel space will discover statistically distinct

behaviours that may be semantically trivial. What is needed is a state representation whose geometry reflects human-interpretable semantic similarity, one in which a standing walker and a fallen one are far apart not because their pixel arrays differ, but because they constitute meaningfully different physical configurations. This section introduces CLIP as a model whose pretraining objective produces exactly this structure.

2.2.1 CLIP (Contrastive Language-Image Pre-Training)

CLIP [Radford et al., 2021] is a VLM trained on 400 million image-text pairs collected from the internet. Rather than optimising for a specific downstream task, CLIP is trained via a contrastive objective that pulls together the embeddings of matching image-text pairs while pushing apart non-matching ones, across a shared latent space. The result is an encoder that maps images and text into a common geometry organised by semantic content: images of similar concepts cluster together not because their pixels are alike, but because they are described by similar language. Originally developed for zero-shot image classification and cross-modal retrieval, CLIP's semantically structured embedding space has since found application well beyond its original purpose, including as a reward signal and representation backbone in RL (Section 2.3).

Cosine similarity between unit-normalised CLIP embeddings lies in $[-1, 1]$, where 1 indicates identical direction in embedding space and 0 indicates orthogonality. For use as a distance metric in skill discovery, we convert to cosine distance:

$$d_{\cos}(e_i, e_j) = 1 - \frac{e_i \cdot e_j}{\|e_i\|\|e_j\|} \tag{2.5}$$

which lies in $[0, 2]$, where 0 indicates identical embeddings and 2 indicates maximally dissimilar ones. In practice, semantically similar images typically achieve cosine distances below 0.1 (as shown in Figure 1.1), while semantically unrelated images fall in the range 0.2–0.5.

CLIP maps pixel observations directly into dense, deterministic numerical vectors, allowing semantic similarity to be measured via cosine distance without intermediate text generation. The CLIP latent space exhibits several properties that motivate its suitability for skill discovery:

- **Semantic topology.** CLIP induces a metric geometry in which distance reflects categorical behavioural similarity rather than pixel-level difference, making cosine distance a principled proxy for behavioural dissimilarity in skill discovery.
- **Human-aligned semantics.** Because CLIP was trained on human-curated captions, its embedding space is organised around the concepts humans use to describe scenes (e.g., posture, motion, and object interaction), meaning the categories that naturally emerge in CLIP space coincide with the behavioural categories meaningful for downstream locomotion tasks without any task-specific supervision.
- **Empirical robustness to distribution shift.** Fang et al. [2022] demonstrate that CLIP exhibits substantially higher robustness to natural distribution shifts than standard vision models trained on comparable data, meaning discovered skills are more likely to transfer across visual conditions.

There have been a range of attempts to introduce foundation models into RL [Xiao et al., 2025, Firoozi et al., 2025], including LLM-generated reward functions [Ma et al., 2024, Xie et al., 2024] and VLM-based trajectory preference labelling [Wang et al., 2024b, Singh et al., 2025],

demonstrating that generative models carry rich semantic knowledge that can meaningfully shape agent behaviour. However, they share a fundamental incompatibility with the dense, high-frequency reward signals that skill learning requires. Generative VLMs produce stochastic token sequences, meaning the same observation queried twice may yield different evaluations, introducing variance that destabilises training. More practically, the inference latency of models at this scale is incompatible with continuous control environments running at tens of steps per second.

CLIP resolves both problems simultaneously. Used as a contrastive encoder rather than a generative oracle, it maps observations to dense, deterministic numerical vectors in a single frozen forward pass, with inference latency orders of magnitude lower than autoregressive VLMs and compatible with continuous control environments running at tens of steps per second. Semantic similarity can then be computed as cosine distance between embedding vectors, providing a stable, low-latency signal that can be evaluated at every environment step without stochastic sampling. Freezing the encoder removes it from the RL gradient graph entirely, allowing its outputs to be cached at collection time and reducing computational overhead relative to a jointly trained convolutional encoder.

2.3 CLIP in Reinforcement Learning

With the properties of CLIP embeddings established, we survey their application in RL, proceeding from goal-conditioned reward shaping through exploration to skill discovery.

2.3.1 CLIP as a Goal-Reaching Reward Signal

CLIP cosine similarity provides a zero-shot measure of how well the current state matches a language description, and several works exploit this for reward shaping. VLM-RM [Rocamonde et al., 2024] demonstrated this on humanoid locomotion; Baumli et al. [2023] fine-tuned CLIP to output goal-achievement probability directly; and RoboCLIP [Sontakke et al., 2023] extended the paradigm to manipulation by comparing agent trajectories against CLIP-embedded video demonstrations. Collectively, these works confirm that CLIP embeddings constitute a semantically meaningful state space for reward shaping, but all require a human to specify the target behaviour at training time.

2.3.2 CLIP for Exploration

CLIP embeddings have also been used as intrinsic motivation signals, measuring novelty in semantic rather than pixel space to discount visually salient but semantically irrelevant variation such as lighting changes [Tam et al., 2022, Gupta et al., 2022, Andres and Del Ser, 2024]. As noted in Section 2.1.2, these fall within the knowledge-acquisition family of URL methods rather than skill discovery: no latent skill variable is conditioned on and no structured repertoire of reusable behaviours is produced for downstream selection.

2.3.3 Foundation Models as Skill Discovery Guides

FoG [Yang et al., 2025] is the most direct antecedent of this work among foundation model approaches. Rather than using CLIP for goal-reaching or state novelty, FoG explicitly targets skill discovery, using CLIP to score states against human-specified text descriptions of desirable

and undesirable behaviours. For example, an agent receives positive reward for states matching "running" and negative reward for states matching "standing still", steering skill discovery toward semantically meaningful locomotion behaviours.

However, two aspects of its design limit its applicability. First, FoG requires human-specified text descriptions of desirable and undesirable behaviours at training time, a form of supervision that contradicts the fully unsupervised assumption underlying skill discovery. Second, FoG uses CLIP only as an auxiliary scoring mechanism layered over a pixel-space policy. The actor, critic, and skill discovery objective all operate on representations produced by a task-trained CNN, with CLIP consulted only to shape the reward signal.

2.3.4 Language-Grounded Skill Discovery

DLSD [Rajaraman, Myers and Dragan, 2024] and LGSD [Rho et al., 2025a] are the closest prior works to our contribution. Both replace uninformed pixel-space representations with semantically structured language embeddings, demonstrating substantially improved skill diversity and interpretability. DLSD replaces DIAYN's discriminator with one operating on natural language descriptions of states, while LGSD grounds the Lipschitz constraint of a WDM in cosine distance between LLM-generated state descriptions. Both approaches share a critical limitation: they require step-wise language annotation of states, either via a hand-crafted captioner or an LLM oracle, restricting them to environments where states can be faithfully transcribed to language and incurring inference latency incompatible with high-frequency control. Both works identify VLMs applied directly to visual inputs as a natural extension, which is precisely the direction this work pursues.

2.3.5 Positioning This Work

The literature surveyed in this section reveals a consistent pattern: methods that introduce semantic structure into skill discovery incur either human supervision, symbolic state access, or step-wise generative inference as the price of that structure, while methods that use CLIP in RL do so for goal-reaching or exploration rather than USD. To the best of our knowledge, no existing work integrates a frozen vision-language encoder directly into the skill discovery loop as the primary state representation.

This work fills that gap. By replacing the task-trained CNN encoder with a frozen CLIP visual encoder in DIAYN and APS, we eliminate all three requirements while preserving the semantic grounding that prior work has shown to be beneficial for discovering behaviourally meaningful skills. The central hypothesis is that CLIP's contrastive pretraining has already organised the visual embedding space such that semantic skill discrimination is tractable from pixels alone.

Chapter 3

Methodology

3.1 CLIP Integration

This section details how the frozen CLIP encoder replaces the task-trained CNN in the standard URLB architecture, covering encoder selection, semantic normalisation, and temporal context. This pipeline produces the state representation e_t consumed by the downstream skill discovery algorithms.

3.1.1 Encoder Selection

The standard URLB architecture [Laskin et al., 2021] uses a shared convolutional encoder that processes raw pixel observations and feeds the resulting feature maps into the actor, critic, and intrinsic reward modules. In our framework, this CNN backbone is replaced entirely with a frozen CLIP image encoder f_{CLIP} [Radford et al., 2021]. We adopt the ViT-B/32 architecture, which produces 512-dimensional image embeddings and represents the smallest standard CLIP variant, chosen for its balance between semantic representation quality and computational feasibility within the URLB pretraining budget of 2×10^6 environment frames. We additionally ablate against ViT-g-14, a substantially larger encoder pretrained on 2 billion image-text pairs, to assess whether representational capacity translates to improved skill discovery within this budget (Section 5.5.3).

The CLIP encoder is kept strictly frozen throughout all phases of pretraining and finetuning for two reasons. First, updating the encoder via sparse, noisy RL gradients would distort the pretrained semantic manifold, destroying the structured geometry that motivates the substitution and replacing it with a task-specific representation no richer than a CNN trained from scratch. Second, freezing the encoder allows the CLIP forward pass to be removed from the RL gradient graph entirely, enabling the computational optimisations described in Appendix C.

3.1.2 Semantic Normalisation and Dimensionality

To extract features from an individual frame o_t, we compute its ℓ_2-normalised embedding:

$$\widehat{e}_t = \frac{f_{\text{CLIP}}(o_t)}{\|f_{\text{CLIP}}(o_t)\|_2}. \tag{3.1}$$

Projecting the embedding onto the unit hypersphere ensures that downstream distance computations operate in the same geometric space in which the representations were learned. Because CLIP's contrastive training objective is defined in terms of cosine similarity [Radford et al., 2021], the model produces embeddings in which magnitude carries no semantic information; normalising at inference time is therefore required for modules that use raw dot products or Euclidean distances (e.g., APS's [Liu and Abbeel, 2021a] particle-based entropy estimator) to behave consistently with this geometry. This is consistent with standard practice in CLIP-based RL [Rocamonde et al., 2024].

3.1.3 Temporal Context via Frame Stacking

Following the standard URLB protocol for pixel-based environments, observations consist of three consecutive frames stacked to provide temporal context. In the CNN baseline, all three frames are processed simultaneously across the channel dimension. A naive application of CLIP to this setting would process only the most recent frame, discarding the two earlier frames since CLIP's ViT-B/32 architecture expects a standard three-channel RGB image. This would introduce an unfair asymmetry between the CNN and CLIP conditions regarding velocity and motion data.

To eliminate this, we encode each of the three stacked frames independently through the frozen CLIP encoder. The resulting normalised embeddings are concatenated along the feature dimension to form the final state representation e_t:

$$e_t = \left[\widehat{e}_t^{(1)} \parallel \widehat{e}_t^{(2)} \parallel \widehat{e}_t^{(3)}\right] \in R^{1536}, \tag{3.2}$$

where $\widehat{e}_t^{(k)}$ is the normalised embedding of the k-th frame and $[\cdot \parallel \cdot]$ denotes concatenation. This produces a representation 96% more compact than the CNN baseline, whose convolutional encoder outputs a 39,200-dimensional feature map, reducing replay buffer storage and eliminating encoder gradient computation entirely. Temporal ordering is strictly preserved by the fixed concatenation sequence, allowing downstream MLPs to learn direction-sensitive features (i.e., the difference $\widehat{e}_t^{(3)} - \widehat{e}_t^{(1)}$ implicitly encodes the direction of semantic change). Averaging the three embeddings was rejected, as it is order-invariant and would conflate opposing motions, which is detrimental to locomotion tasks. We ablate our frame stacking implementation in Section 5.5.4.

3.2 Algorithm Selection, Justification, and Modification

This work evaluates the CLIP representation e_t within two skill discovery algorithms drawn from the URLB codebase: DIAYN [Eysenbach et al., 2018] and APS [Liu and Abbeel, 2021a]. DIAYN is selected as the canonical reverse-MI baseline, where the binding mechanism is discriminator capacity. APS is selected as the strongest competence-based method in pixel-based URLB [Laskin et al., 2021], where the binding mechanisms are the SF projection and the particle-based entropy estimator; the original authors additionally identify novel state representations as a promising direction for future work [Liu and Abbeel, 2021a]. Both mechanisms operate directly on encoder representations and should benefit from a space already organised along semantically meaningful axes. Distance-maximisation approaches (see Section 2.1.5) are not included in URLB and are therefore excluded from our evaluation; we discuss their extension to CLIP in Section 7.2.1.

3.2.1 DIAYN: Mutual Information and Discriminability

DIAYN [Eysenbach et al., 2018] was originally designed for low-dimensional state spaces where the discriminator operates directly on quantities such as x-y coordinates, and skill-separating boundaries are easy to learn. In the pixel-based setting, the CNN encoder is initialised randomly and shaped entirely by the discriminability objective, meaning the variational approximation $q_\phi(z \mid s)$ is initially uninformative. The discriminator cannot distinguish skills because the encoder has not yet learned to extract features that separate them. Skill-separating structure must therefore emerge gradually from pixel-level differences alone, exhausting much of the pretraining budget before the intrinsic reward produces a useful signal. The CLIP embedding sits at an intermediate dimensionality between the raw state space and the 39,200-dimensional CNN output, raising the question of whether this compressed, semantically structured space is better suited to DIAYN's discriminator than either alternative.

Replacing the CNN with a frozen CLIP encoder, the discriminator is now conditioned on the concatenated semantic embedding e_t, which is already organised into semantically distinct clusters by contrastive pretraining:

$$r^{\text{DIAYN-CLIP}}(e_t, z) = \log q_\phi(z \mid e_t) - \log p(z). \tag{3.3}$$

The variational approximation $q_\phi(z \mid e_t)$ therefore faces a much easier classification problem from the first update: rather than waiting for the encoder to develop discriminative structure through gradient feedback, the discriminator operates on a fixed space in which semantically distinct states are already separated, and need only learn the boundaries between pre-existing clusters rather than constructing non-linear decision boundaries from scratch to parse pixel noise.

3.2.2 APS: Entropy Maximisation and Successor Features

The CLIP substitution has distinct effects on each component of APS's intrinsic reward, which we address in turn.

Exploration term. The particle-based entropy reward measures novelty via k-nearest-neighbour distances computed in the output space of ϕ:

$$r^{\text{ent}}(e_{t+1}) = \log\left(1 + \frac{1}{k}\sum_{j \in \mathcal{N}_k} \left\|\phi(e_{t+1}) - \phi(e^{(j)})\right\|_2^2\right), \tag{3.4}$$

where $\mathcal{N}_k$ denotes the k nearest neighbours of $\phi(e_t)$ in the replay buffer and ϕ is applied without output normalisation. When ϕ operates on CNN features, these distances reflect pixel-level variation, meaning the entropy reward can be consumed by semantically trivial novelty such as lighting changes or background shifts. With CLIP, distances are computed in a space already invariant to such perturbations, so the entropy reward measures *semantic* novelty instead, driving the agent toward states that are behaviourally distinct rather than merely visually different.

Exploitation term. The SF reward uses the ℓ_2-normalised output of the same network:

$$r^{\text{sf}}(e_{t+1}) = \frac{\phi(e_{t+1})}{\|\phi(e_{t+1})\|_2}^\top w, \tag{3.5}$$

where w is a randomly sampled task vector on the unit sphere. This reward is linear in the state features, requiring that task-relevant variation falls along linear axes of ϕ. With a CNN encoder, ϕ must simultaneously extract meaningful features from pixels and organise them to support this linear structure. With CLIP, ϕ receives a semantically structured input in which behaviourally distinct states are already separated along interpretable directions, so it need only learn a shallow projection identifying which pre-existing semantic dimensions are task-relevant. The full intrinsic reward combines both terms:

$$r^{\text{APS}}(e_t, a, e_{t+1}) = r^{\text{ent}}(e_{t+1}) + r^{\text{sf}}(e_{t+1}). \tag{3.6}$$

Probe phase. This same property carries through to finetuning, where the optimal task vector is identified by least-squares regression over probe transitions:

$$w^* = \frac{\Phi^\dagger r_{\text{ext}}}{\|\Phi^\dagger r_{\text{ext}}\|_2} \tag{3.7}$$

where $\Phi \in R^{N \times d_{\text{SF}}}$ is the matrix of normalised state feature vectors collected during the probe and r_{ext} is the extrinsic reward vector. In the CNN baseline, the axes of ϕ are shaped by the skill discovery objective alone and may not align with quantities meaningful for downstream tasks. Because ϕ here projects from a semantically structured CLIP space, its axes are more likely to correspond to behaviourally meaningful quantities such as forward velocity or torso height, meaning w^* more accurately captures the task-relevant direction from the same 4,096 probe transitions and produces a better-initialised finetuning phase.

Chapter 4

Experimental Setup

4.1 Experimental Conditions

To isolate the contribution of the CLIP encoder from the underlying skill discovery objective, we evaluate a 2×2 matrix of conditions: two skill discovery algorithms (DIAYN and APS) each evaluated under two visual encoder architectures, yielding four experimental conditions per domain. We denote each agent as a hyphenated tuple of its USD method and visual encoder (e.g. DIAYN-CNN).

Skill discovery algorithms.

1. **DIAYN** [Eysenbach et al., 2018]. Following URLB, the discriminator learns 16 skills across all environments. We additionally ablate skill space size with 8 and 32 skills in Section 5.5.
2. **APS** [Liu and Abbeel, 2021a]. Following URLB, a continuous skill vector of dimension 10 is used across all environments.

Visual encoder conditions.

- **CNN (baseline).** The standard URLB four-layer convolutional encoder producing a 39,200-dimensional representation.
- **CLIP (ours).** A frozen ViT-B/32 encoder with frame stacking producing a 1,536-dimensional representation (Section 3.1.3). CLIP weights are never updated during training.

CNN baselines for Textured Walker and Textured Quadruped are taken from Laskin et al. [2021]. Textured Cheetah replaces the Jaco manipulation domain from the original benchmark; as no published CNN baseline exists, we run our own CNN experiments on this domain.

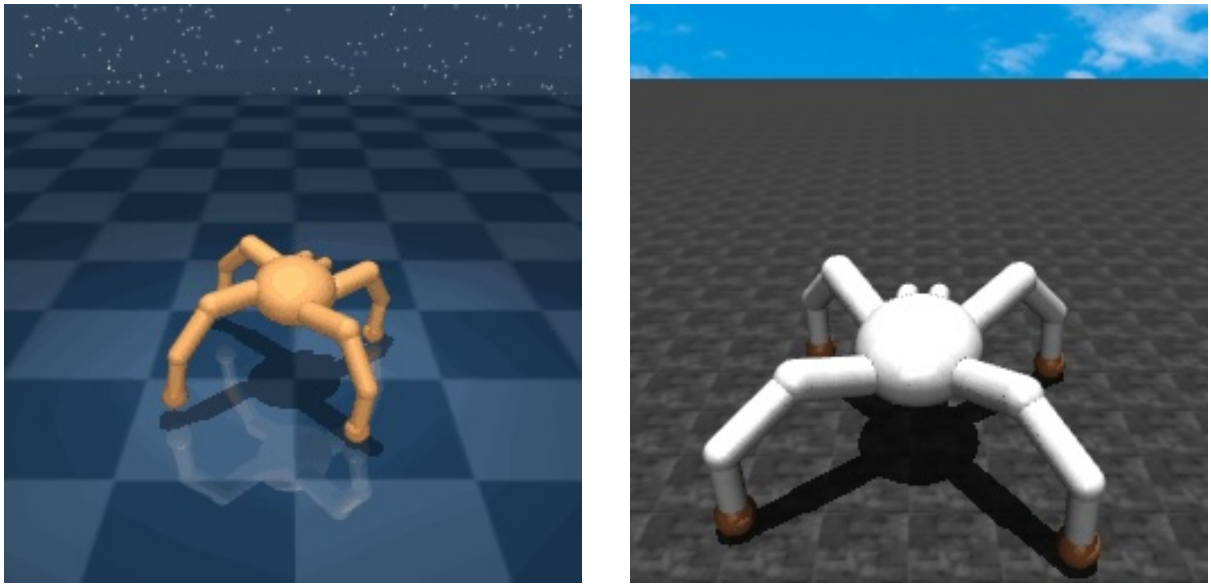
4.2 Environments

We evaluate on twelve downstream tasks from URLB [Laskin et al., 2021], spanning three continuous control domains in ascending order of morphological complexity: **Cheetah**, a planar biped with six degrees of freedom (DoF); **Walker**, a planar biped requiring precise balance

and gait control; and **Quadruped**, a four-legged robot operating in three-dimensional space. Each domain comprises four downstream tasks whose reward specifications are provided in Appendix E.

All domains are implemented on DMC [Tassa et al., 2018] to match the URLB evaluation protocol, with fixed-length episodes of 1000 steps regardless of agent behaviour. We discuss the implications of this for skill discovery in Section 6.3.

Textured variants. All domains are evaluated in textured variants replacing the default uniform floor and plain background with realistic tile textures and sky imagery,[1] with simulated shadows and reduced surface reflectance to narrow the domain gap between DMC observations and CLIP's natural image training distribution. A visual comparison is provided in Figure 4.1. We confirm the impact of these backgrounds on downstream performance in Section 5.5.2.



(a) Original DMC environment (b) Textured environment

Figure 4.1: Comparison of the original DMC environment and the textured variant introduced in this work.

4.3 Implementation

All experiments are implemented on top of the URLB codebase [Laskin et al., 2021]. Both algorithms use DDPG [Lillicrap et al., 2019] as the underlying RL backbone, following the DrQ-v2 implementation of Yarats et al. [2021]. Full implementation details and hyperparameters are provided in Appendix C.

4.4 Evaluation Protocol

We follow the two-phase evaluation protocol of URLB [Laskin et al., 2021].

[1] Texture assets are adapted from the VLM-RM codebase [Rocamonde et al., 2024].

Pretraining. Each agent is pretrained for 2×10^6 environment frames without access to any extrinsic reward signal. The final snapshot is used for all downstream evaluations.

Finetuning. The pretrained snapshot is finetuned for 10^5 environment frames on each downstream task using extrinsic reward alone. We adopt the probe-then-finetune procedure of Liu and Abbeel [2021a], Laskin et al. [2022], in which an initial 4,096-step probe phase identifies the optimal skill or task vector before the actor and critic are updated for the remaining $\approx$96,000 steps. All experiments are run with three independent random seeds. Full details are given in Appendix D.

Chapter 5

Results and Analysis

5.1 Preliminary Sanity Check: CLIP Encodes Behavioural Structure

Before evaluating downstream task performance, we first verify that CLIP's visual representation encodes behaviourally meaningful structure. To test this, we construct a controlled dataset of 1,400 rendered frames from the Textured Walker environment, comprising 200 images per class across seven manually specified poses: Standing, Mid-Stride, Fallen, Flipping, Leaning-Fwd, One-Leg, and Landing. Each image is rendered at 84×84 pixels under randomised lighting conditions (varying light position, ambient intensity, and diffuse intensity) and with random horizontal agent displacement, ensuring that clusters cannot be explained by low-level visual cues such as brightness or position.

Each frame is encoded with a frozen CLIP ViT-B/32 encoder and the resulting 512-dimensional embeddings are projected to two dimensions via t-SNE (perplexity 15). We additionally quantify cluster quality using 5-fold cross-validated k-nearest neighbour (k=5) classification accuracy.

As shown in Figure 5.1, CLIP ViT-B/32 produces seven well-separated clusters with no meaningful overlap, achieving a kNN accuracy of 99%. This demonstrates that a frozen CLIP encoder, without any environment-specific training, organises pixel observations into a latent space that reflects human-interpretable postural categories, even under significant photometric variation. This provides direct empirical support for our hypothesis that CLIP's semantic prior can substitute for a task-trained encoder in the skill discovery loop.

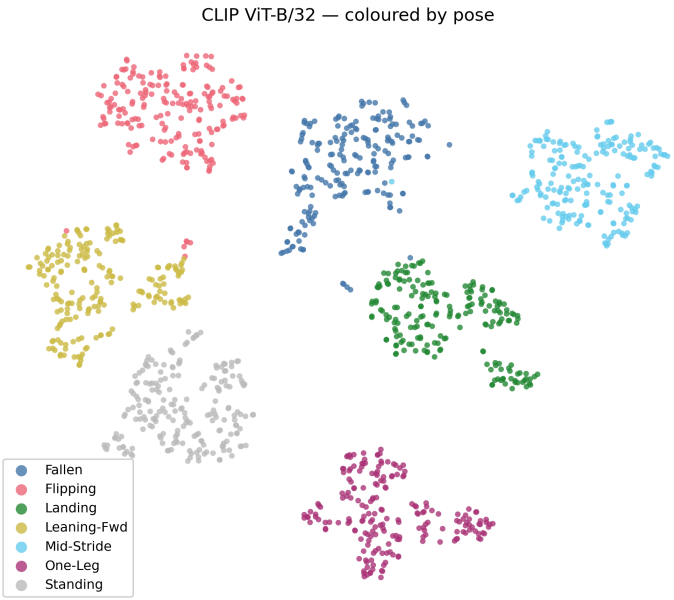


Figure 5.1: t-SNE projection of CLIP ViT-B/32 embeddings for 1,400 rendered frames across seven manually specified Walker poses, with randomised lighting and agent position. Each colour corresponds to a distinct pose class. The clusters are well-separated, achieving a k-NN classification accuracy of 99%, demonstrating that CLIP organises pixel observations by behavioural semantics rather than low-level visual features.

5.2 Pretraining Results

We report pretraining metrics at the final checkpoint for all CLIP methods across three domains. CNN pretraining horizons are shorter due to memory constraints[1]. CNN pretraining was therefore limited to $\approx$ 500k frames on Textured Walker and Textured Quadruped, and to $\approx$ 1M frames on Textured Cheetah; CNN metrics are reported at these shorter horizons for comparison. CNN finetuning on Walker and Quadruped is not required as Laskin et al. [2021] already provide the relevant baselines; CNN pretraining on these domains is run only to provide indicative representation quality metrics. Both CLIP agents are pretrained to 2M frames across all three domains.

Table 5.1 reports final pretraining metrics. For DIAYN we report discriminator accuracy (chance $= 1/16 = 6.25\%$), where higher values indicate cleaner skill separation. For APS we report the entropy reward r^{ent} (broader state coverage), the successor feature reward r^{sf} (trajectory coherence with the sampled task vector $\mathbf{w}$), and incidental extrinsic reward (a proxy for how often the agent visits task-relevant states without optimising for them).

Training curves are shown in Figures 5.2, 5.3, and 5.4, respectively.

[1]Storing raw stacked pixel observations in the replay buffer requires approximately 32GB of RAM at 500k transitions (500k $\times$ 9 $\times$ 84 $\times$ 84 bytes), exceeding available system RAM. CLIP runs store pre-cached 512-dimensional embeddings instead, reducing buffer memory to approximately 1GB (500k $\times$ 512 $\times$ 4 bytes.)

Table 5.1: Pretraining metrics at final checkpoint. CLIP methods are reported at 2M frames; CNN methods at 500k frames (Walker, Quadruped) and 1M frames (Cheetah) due to memory constraints. All values averaged over the final 10k training steps. Discriminator accuracy reported for DIAYN (chance = 6.25%). Intrinsic reward components and extrinsic reward reported for APS. †CNN Walker and Quadruped pretraining was not run for downstream finetuning; metrics are indicative of representation quality only.

Domain	Method	Frames	Disc. Acc. (%) ↑	r^{ent} ↑	r^{sf} ↑	Extr. Reward ↑
Textured Walker	DIAYN-CLIP	2M	99.6	–	–	0.409
	DIAYN-CNN†	500k	70.1	–	–	0.488
	APS-CLIP	2M	–	1.127	0.574	0.638
	APS-CNN†	500k	–	1.225	0.657	0.461
Textured Quadruped	DIAYN-CLIP	2M	84.8	–	–	0.471
	DIAYN-CNN†	500k	58.1	–	–	0.989
	APS-CLIP	2M	–	1.100	0.492	0.525
	APS-CNN†	500k	–	1.123	0.396	0.409
Textured Cheetah	DIAYN-CLIP	2M	99.0	–	–	0.099
	DIAYN-CNN	1M	70.2	–	–	0.060
	APS-CLIP	2M	–	1.146	0.665	0.236
	APS-CNN	1M	–	1.287	0.703	0.230

DIAYN. DIAYN-CLIP reaches near-ceiling discriminator accuracy on Textured Walker and Textured Cheetah (99.6% and 99.0% respectively), converging rapidly and remaining stable throughout training (Figure 5.2). On Textured Quadruped, accuracy increases more gradually and continues rising at the 2M frame horizon, consistent with Quadruped's higher morphological complexity: with 12 DoF compared to 6 for Walker and Cheetah, the space of visually distinct body configurations is larger and CLIP's embedding space continues to find separable structure throughout training. DIAYN-CNN, by contrast, plateaus considerably lower across all domains and exhibits a declining trend after its peak, suggesting CNN representations struggle to maintain clean skill separation over extended pretraining. Incidental extrinsic reward is higher under DIAYN-CNN on Walker and Quadruped but lower on Cheetah (Figure 5.3).

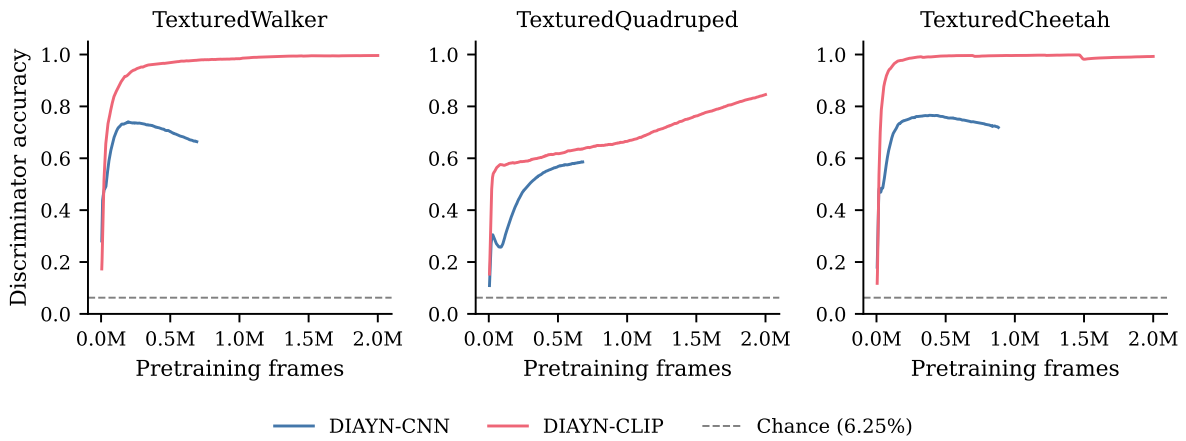


Figure 5.2: DIAYN discriminator accuracy during pretraining across three domains. The dashed horizontal line indicates chance performance ($1/16 = 6.25\%$).

APS. Although APS-CNN produces stronger entropy (r^{ent}) and SF (r^{sf}) rewards than APS-CLIP across almost all domains (Figure 5.4), this does not translate to higher incidental extrinsic reward. At a shared intermediate checkpoint, APS-CNN leads on entropy by 22–26% across all three domains, yet the SF reward gap is small and inconsistent, with CLIP leading on Quadruped and trailing by only 6–10% on Walker and Cheetah. This pattern is consistent with pixel-level novelty being cheaper to accumulate than semantic novelty where CNN entropy is inflated by lighting changes, background shifts, and positional variation that register as novel in pixel space but correspond to identical behavioural states. CLIP suppresses these perturbations by construction, so its lower entropy reflects more selective exploration rather than less exploration, while the comparable SF reward indicates that task-conditioned behaviours of similar coherence are still being learned.

The domain-level extrinsic reward curves (Figure 5.3) support this interpretation. On Textured Walker, APS-CLIP not only exceeds the CNN extrinsic plateau of 0.461 but is still rising at the 2M frame horizon, suggesting that semantically targeted exploration continues to find task-relevant states beyond the point where CNN-based exploration saturates. On Textured Quadruped, both encoders converge to comparable extrinsic reward of approximately 0.5, suggesting that the semantic filter provides no particular advantage or disadvantage in coverage terms on this domain. On Textured Cheetah the pattern reverses: APS-CLIP plateaus at 0.236 while APS-CNN remains on an upward trajectory at its final checkpoint, consistent with CLIP's frozen encoder failing to generate sufficient novelty signal on a domain where the discriminating feature between states is motion rather than posture.

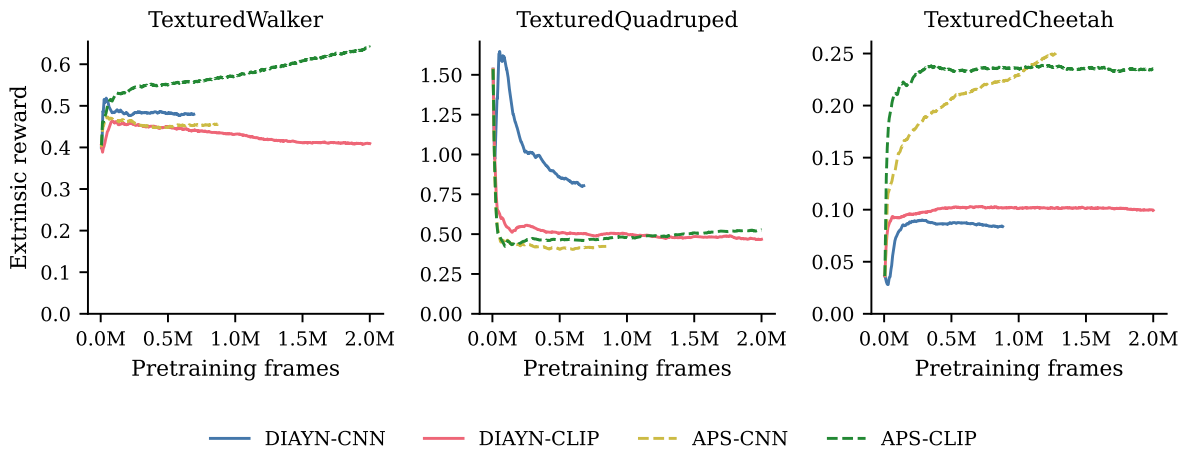


Figure 5.3: Incidental extrinsic reward during pretraining across all four methods and three domains. Solid lines denote DIAYN variants; dashed lines denote APS variants.

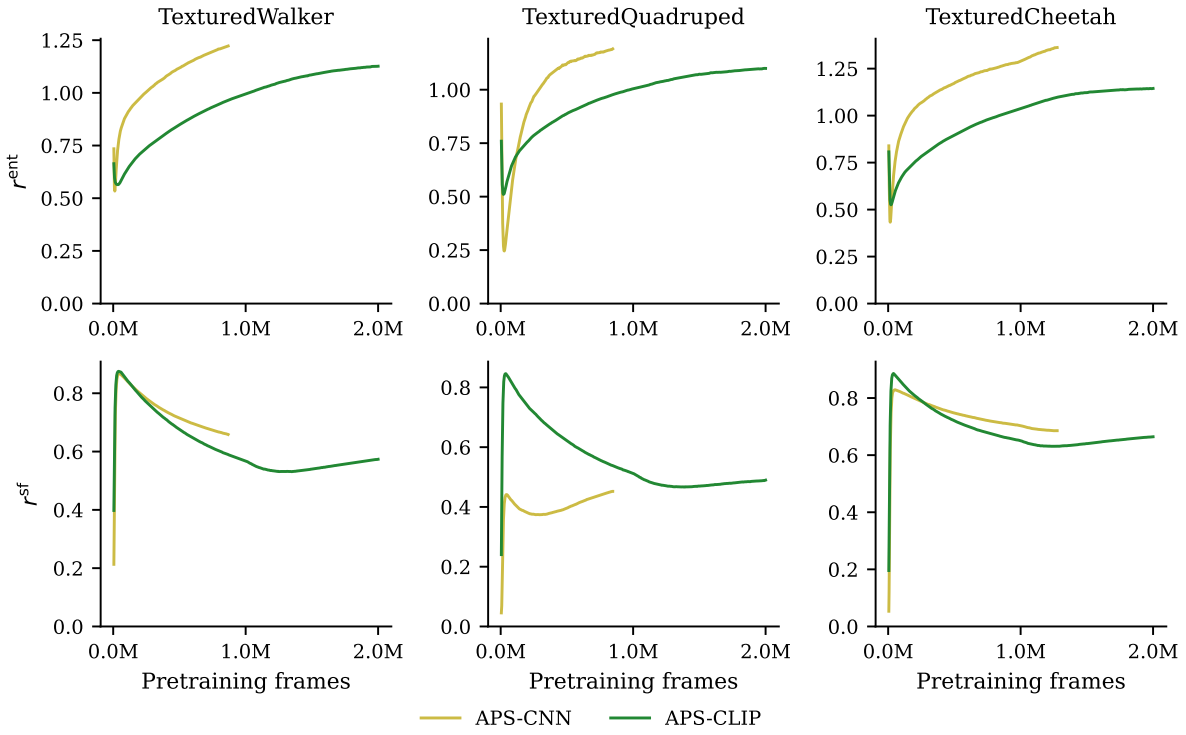


Figure 5.4: APS intrinsic reward components during pretraining. Top row: entropy reward r^{ent}; bottom row: SF reward r^{sf}.

The pretraining results present a superficially counterintuitive picture: APS-CNN produces higher intrinsic rewards across almost all domains, yet as the following results show, this does not translate to stronger downstream performance on Textured Walker and Textured Quadruped. The interpretation, that CNN entropy is inflated by semantically trivial novelty while CLIP's lower entropy reflects more targeted exploration, is developed fully in Section 6. We present the finetuning results first and return to this explanation after the quantitative picture is complete.

5.3 Finetuning Results

Domain	Task	Pre-training for 2×10^6 environment steps			
		DIAYN CNN	DIAYN CLIP	APS CNN	APS CLIP
Textured Walker	Stand	$\mathbf{298 \pm 71}$	157 ± 17	172 ± 10	$\mathbf{303 \pm 17}$
	Walk	$\mathbf{132 \pm 19}$	106 ± 16	37 ± 5	$\mathbf{231 \pm 11}$
	Run	45 ± 3	$\mathbf{60 \pm 11}$	30 ± 3	$\mathbf{88 \pm 14}$
	Flip	114 ± 21	$\mathbf{138 \pm 17}$	38 ± 9	$\mathbf{271 \pm 57}$
Textured Quadruped	Stand	$\mathbf{426 \pm 40}$	120 ± 44	331 ± 43	$\mathbf{538 \pm 140}$
	Walk	$\mathbf{184 \pm 23}$	63 ± 15	146 ± 24	$\mathbf{167 \pm 54}$
	Run	$\mathbf{190 \pm 18}$	71 ± 23	$\mathbf{156 \pm 24}$	132 ± 38
	Jump	$\mathbf{262 \pm 22}$	140 ± 29	199 ± 29	$\mathbf{251 \pm 31}$
Textured Cheetah†	Run	$\mathbf{212 \pm 75}$	90 ± 12	$\mathbf{195 \pm 64}$	40 ± 66
	Flip	$\mathbf{572 \pm 78}$	459 ± 44	$\mathbf{609 \pm 86}$	500 ± 126
	Run backward	$\mathbf{286 \pm 37}$	103 ± 10	$\mathbf{412 \pm 33}$	58 ± 96
	Flip backward	$\mathbf{565 \pm 63}$	502 ± 3	$\mathbf{704 \pm 20}$	393 ± 331

Table 5.2: Finetuning performance across three textured locomotion domains. Results report mean episode reward $\pm$ standard deviation across three seeds. CNN baselines for Textured Walker and Textured Quadruped are taken from Laskin et al. [2021] using plain, untextured environments; CNN baselines for Textured Cheetah are from our own runs. †CNN Cheetah baselines are reported at 1×10^6 pretraining steps due to memory constraints (Section 4). The highest score within each DIAYN and APS pair is in **bold**. Aggregate improvement ratios are computed as the mean of per-task score ratios APS-CLIP$_i$/APS-CNN$_i$ across the four downstream tasks within each domain; APS-CLIP achieves $4.52\times$ on Textured Walker and $1.22\times$ on Textured Quadruped. CLIP-based methods underperform CNN on all Textured Cheetah tasks.

5.3.1 Textured Walker

On Textured Walker, DIAYN-CLIP underperforms its CNN counterpart on stand (157 ± 17 vs 298 ± 71) and walk (106 ± 16 vs 132 ± 19), but matches or exceeds CNN on the more dynamic tasks of run (60 ± 11 vs 45 ± 3) and flip (138 ± 17 vs 114 ± 21). APS-CLIP shows a more pronounced and consistent improvement over APS-CNN across all four tasks, with particularly large gains on the locomotion tasks of walk (231 ± 11 vs 37 ± 5) and flip (271 ± 57 vs 38 ± 9), where CNN-based APS largely fails.

5.3.2 Textured Quadruped

On Textured Quadruped the DIAYN/APS divergence is more pronounced. DIAYN-CLIP substantially underperforms DIAYN-CNN across all four tasks, with the largest gap on stand (120 ± 44 vs 426 ± 40), while the remaining three tasks show broadly comparable deficits of approximately 120 points each. The explanation mirrors the Walker case but is amplified by Quadruped's higher morphological complexity. The failure is one of behavioural coverage:

with 12 DoFs and fixed-length episodes that impose no penalty for falling, the diversity objective discovers skills that are visually distinct in CLIP space but cluster within a narrow, task-irrelevant region consisting predominantly of variations of the agent in collapsed or fallen configurations (Appendix G). The discriminator successfully separates these poses from one another, but none of the resulting skills span the locomotion-relevant configurations that downstream tasks reward, meaning the probe phase has little useful structure to select from.

5.3.3 Textured Cheetah

CNN baselines outperform CLIP-based methods across all four tasks. APS-CLIP shows extreme variance on directed locomotion tasks — run (40 ± 66) and run backward (58 ± 96) — with the majority of seeds producing near-zero scores. DIAYN-CLIP achieves broadly comparable scores to DIAYN-CNN on the flip tasks but falls considerably short on run (90 ± 12 vs 212 ± 75) and run backward (103 ± 10 vs 286 ± 37). The underlying cause is discussed in Chapter 6: CLIP's frozen image-level encoder encodes no motion information, making it fundamentally ill-suited to domains where the discriminating feature between skills is temporal rather than postural.

5.3.4 Aggregate Comparison

APS-CLIP achieves a $4.52\times$ aggregate improvement over APS-CNN on Textured Walker and $1.22\times$ on Textured Quadruped, while CLIP-based methods underperform CNN on all Textured Cheetah tasks. DIAYN-CLIP presents a more mixed picture, matching or exceeding DIAYN-CNN on dynamic Walker tasks but underperforming on Quadruped and Cheetah. The conditions under which CLIP helps and fails are examined in Chapter 6.

5.4 Qualitative Analysis: Textured Walker

Downstream task performance alone presents an incomplete picture of what each method has learned: a high score on Walker Run, for instance, reflects speed achieved but not how that speed is produced, nor whether the underlying skill repertoire is semantically coherent or behaviourally diverse. To complement the quantitative results, we conduct a qualitative analysis of the discovered skills on Textured Walker, the domain where CLIP shows the strongest gains, comparing all four methods across both DIAYN and APS. We additionally analyse our two remaining environments, Textured Cheetah and Textured Quadruped, in Appendix G.

As DIAYN uses discrete skill representations, the agent is conditioned on each of the 16 one-hot skill vectors in turn, producing one rollout per skill. For APS, as no fixed skill set exists, the task vector w is sampled uniformly at random from the 10-dimensional unit sphere, and we sample 10 task vectors to provide a representative cross-section of the continuous skill space. DIAYN-CNN rollouts are visualised in Figure 5.5 as motion composites of three evenly spaced frames (start, midpoint, terminal) to illustrate trajectory dynamics, whilst the remaining three methods produce predominantly static skills and are therefore shown as terminal frames only.



(a) DIAYN-CNN: 16 skills (3-frame composite)



(b) DIAYN-CLIP: 16 skills (terminal frame)



(c) APS-CNN: 10 skills (terminal frame)



(d) APS-CLIP: 10 skills (terminal frame)

Figure 5.5: Discovered skill repertoires on Textured Walker after 2M pretraining frames. DIAYN-CNN is shown as a motion composite of three frames (start, midpoint, terminal) with earlier frames faded, revealing trajectory dynamics. All other methods are shown as terminal frames, reflecting the static nature of their discovered skills. CNN skills are evaluated at 500k pretraining frames due to memory constraints. Skills are 0-indexed from top-left to bottom-right.

The motion composite for DIAYN-CNN reveals highly erratic, unstable motion across all 16 skills, where the faded earlier frames show the walker in mid-fall configurations that differ substantially from the terminal state, indicating that the CNN discriminator separates skills through chaotic trajectories rather than coherent behaviours. In contrast, DIAYN-CLIP, APS-CNN, and APS-CLIP all learn static skills in which the agent reaches a held posture early in the episode and remains there.

DIAYN-CLIP produces stable held postures, confirming that CLIP's semantic structure encourages the discriminator to anchor skills to visually distinct configurations. However, the skill repertoire is not fully diverse. Skills 2 and 3, for instance, both show the walker with one leg raised, differing only in the angle of the foot, a subtle distinction sufficient for CLIP's

discriminator to classify but representing a failure of the objective to exploit the full breadth of CLIP's embedding space.

APS-CNN also converges to static poses rather than dynamic behaviours, but with noticeably less diversity. Skills 5 and 8 are visually identical, both showing a raised leg in the same configuration, suggesting that APS-CNN's successor feature objective has collapsed multiple task vectors onto the same region of the state space. APS-CLIP produces the most visually distinct repertoire of the four methods, where we observe terminal states spanning a wider range of body configurations with no duplicated postures. This behaviour is consistent with CLIP's structured embedding space providing a stronger signal for the particle-entropy objective to spread skills apart. This qualitative contrast directly supports the quantitative finding that APS-CLIP substantially outperforms all other methods on Walker locomotion tasks (Table 5.2).

5.5 Ablations

We conduct a series of ablation experiments to isolate the factors that contribute to the performance of CLIP-based skill discovery. We ablate APS-CLIP on Textured Quadruped Stand as APS consistently outperforms DIAYN in our main experiments, and Stand is the highest-scoring task across all domains, providing the clearest signal for detecting performance differences. The skill space size ablation uses DIAYN-CLIP, as the number of skills is specific to DIAYN's discriminator-based objective. The frame stacking ablation additionally evaluates all four Quadruped downstream tasks to see if temporal context specifically benefits dynamic location.

Ablation	**Method**	**Task(s)**	**Motivation**
Pretraining Scalability	APS	Quadruped Stand	Does more pretraining translate to increased performance on downstream tasks?
Textured environments	APS	Quadruped Stand	Do the sky and floor textures better activate CLIP's visual priors?
CLIP model scale	APS	Quadruped Stand	Does representation quality improve with encoder scale, and is the additional compute justified?
Frame stacking	APS	Quadruped Stand, Walk, Run, Jump	Does providing temporal context via stacked frames improve performance on dynamic tasks?
Skill space size	DIAYN	Quadruped Stand	Does a larger skill vocabulary improve coverage or saturate CLIP's representational capacity?

Table 5.3: Summary of ablation studies. All finetuning runs are averaged across 3 seeds unless otherwise stated.

5.5.1 Pretraining Scalability

To assess how pretraining duration affects downstream performance, we evaluate APS-CLIP and APS-CNN from four pretraining checkpoints (100k, 500k, 1M, and 2M frames) on Textured Quadruped Stand across three seeds. Results are shown in Figure 5.6.

APS-CLIP improves substantially with pretraining duration, rising from 304 ± 130 at 100k steps to 538 ± 140 at 2M steps. A temporary drop at 1M (381 ± 238) is attributable to high seed variance at that checkpoint rather than a genuine decline. APS-CNN, by contrast, plateaus early and shows minimal improvement beyond 500k steps (281 ± 51 at 500k vs 331 ± 43 at 2M), suggesting that CNN representations do not continue to benefit from additional pretraining in the same way. This is consistent with the CNN encoder reaching representational capacity relatively quickly, whereas CLIP's frozen embedding space allows the policy and SF to continue improving as the agent explores more of the semantic skill space.

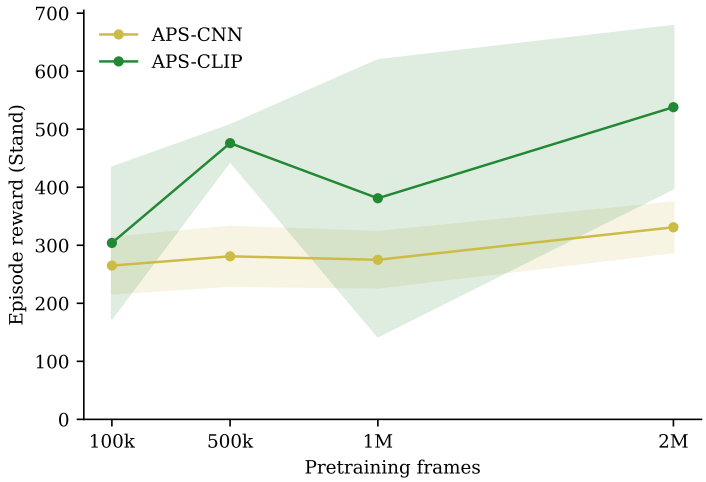


Figure 5.6: Downstream performance on Textured Quadruped Stand as a function of pretraining duration for APS-CLIP and APS-CNN. Shaded regions indicate $\pm$ one standard deviation across three seeds.

These results suggest that APS-CLIP continues to benefit from additional pretraining beyond the point where APS-CNN plateaus. Performance has not plateaued at the 2M frame horizon, and establishing a true saturation point would require running beyond 2M frames, something not feasible within the compute constraints of this work.

5.5.2 Effect of Textured Environments

The textured environments introduced in Section 4.2 replace the plain grey surfaces of the original URLB domains with realistic sky, floor, and robot textures. This ablation tests whether those textures specifically benefit CLIP-based agents, since CLIP was pretrained on natural images and may produce more semantically structured embeddings when the visual input more closely resembles its training distribution.

We compare APS-CLIP pretraining and finetuning on plain DMC Quadruped against our textured equivalent, holding all other settings fixed. The pretraining curves (Figure 5.7) show that incidental extrinsic reward is 54% higher in the textured setting (0.525 vs 0.242),

suggesting that CLIP's embeddings are more structured and task-relevant when observations are visually closer to its training distribution. This advantage transfers to finetuning: APS-CLIP achieves 538 ± 140 on Textured Quadruped Stand compared to 441 ± 205 on the plain equivalent, an improvement of 22%.

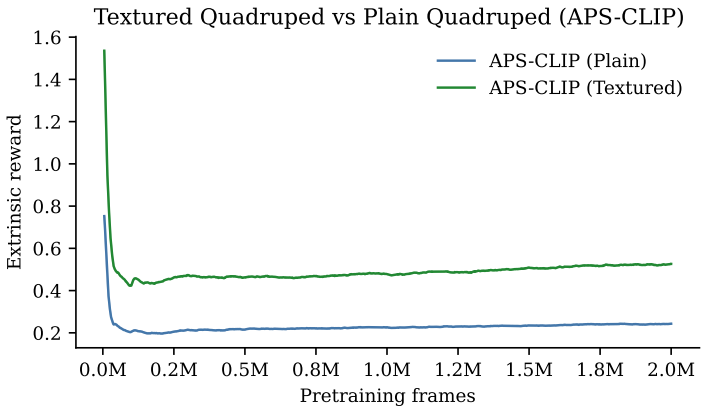


Figure 5.7: Incidental extrinsic reward during APS-CLIP pretraining on plain versus textured Quadruped. Shaded regions indicate the range across seeds.

These results support the hypothesis that CLIP's in-domain advantage grows with visual realism, and suggest that performance would improve further in photorealistic environments such as Habitat [Savva et al., 2019].

5.5.3 Effect of CLIP Model Scale

The main experiments use ViT-B/32, the smallest standard CLIP architecture, for computational tractability. We ablate against ViT-g-14 pretrained on LAION-2B [Schuhmann et al., 2022], a substantially larger model trained on 2 billion image-text pairs with approximately 1.8 billion parameters. ViT-g-14 produces 1024-dimensional embeddings and requires approximately 6GB of GPU memory at inference compared to 500MB for ViT-B/32, with roughly twice the per-step wall-clock cost. This tests whether the semantic structure beneficial for skill discovery scales with model capacity, and whether any gains justify the additional compute within the URLB finetuning budget.

Figure 5.8 shows pretraining curves for both encoders on Textured Quadruped. The intrinsic reward components are marginally higher for ViT-g-14: entropy reward (1.131 vs 1.100), SF reward (0.539 vs 0.492), and total intrinsic reward (1.669 vs 1.591), suggesting that the larger embedding space provides a modestly richer signal for APS's skill discovery objective. However, incidental extrinsic reward tells a different story: ViT-B/32 achieves a final value of 0.525 compared to 0.490 for ViT-g-14, indicating that the smaller model incidentally visits more task-relevant states during pretraining despite its lower intrinsic rewards. This discrepancy suggests that ViT-g-14's higher-dimensional embedding space may encourage broader exploration of semantically distinct but task-irrelevant states.

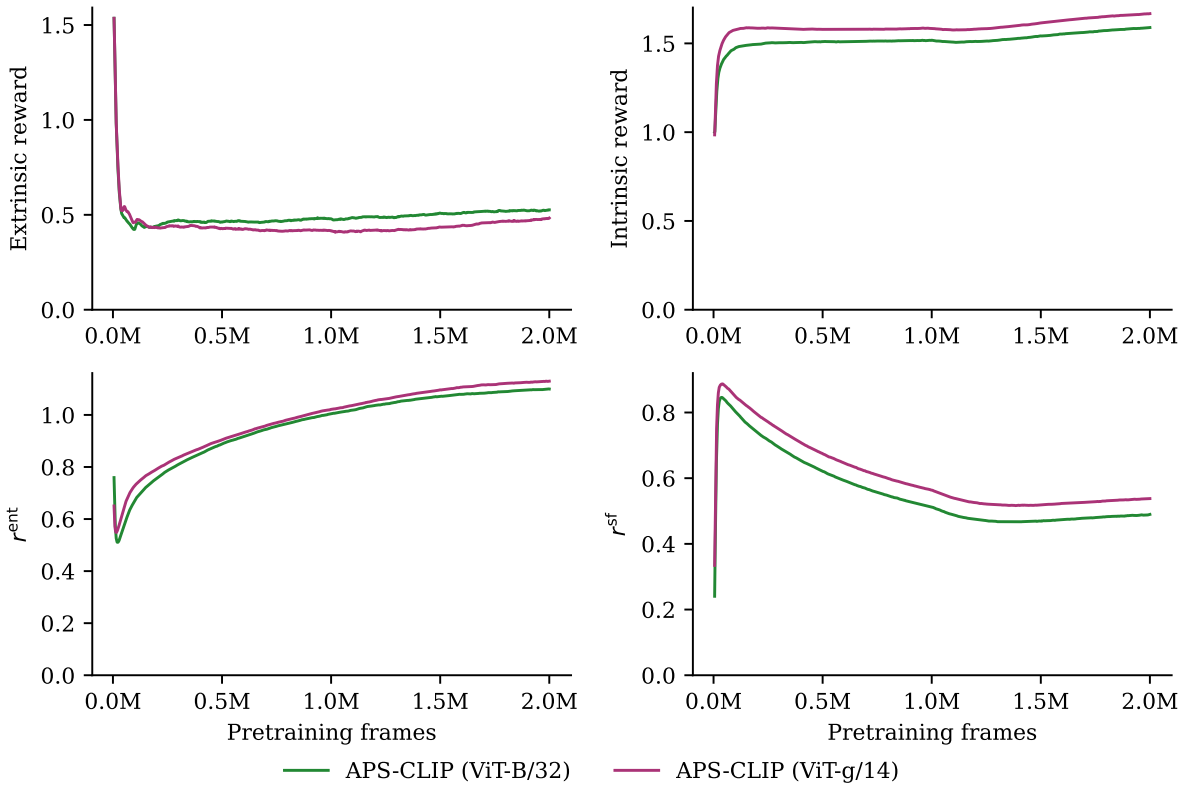


Figure 5.8: Pretraining curves for APS-CLIP on Textured Quadruped comparing ViT-B/32 and ViT-g-14 encoders. Top row: extrinsic reward and total intrinsic reward. Bottom row: entropy reward r^{ent} and SF reward r^{sf}.

This pattern is reflected in downstream finetuning: ViT-B/32 achieves 538 ± 140 on Textured Quadruped Stand compared to 381 ± 95 for ViT-g-14, despite the latter's stronger intrinsic signals during pretraining. One possible explanation is that ViT-g-14's 1024-dimensional embeddings are harder for the APS SF regression to compress into a useful task vector during the probe phase, increasing sensitivity to the specific skills discovered and reducing the reliability of finetuning. Notably, both models exhibit similar plateau behaviour in pretraining, suggesting that the performance ceiling is not determined by encoder capacity alone. Rather, both models share the same fundamental constraint: they are frozen encoders pretrained on internet-scale image-text pairs, a training distribution far removed from the textured locomotion environments evaluated here. We argue that this domain gap, rather than model scale, is the primary bottleneck, a point we develop further in Section 6. Overall, ViT-B/32 represents a better trade-off between representational capacity and computational cost for these environments.

5.5.4 Effect of Frame Stacking

The default configuration stacks 3 consecutive frames and concatenates their individual CLIP embeddings to produce a 1,536-dimensional representation, providing the policy with temporal context across recent observations. We ablate against a single-frame encoding (512-dimensional) to test whether this temporal information improves downstream task performance. The benefit is not guaranteed, as CLIP was trained on individual images, so the concatenation of per-frame embeddings is not a representation that the model was trained to produce.

Figure 5.9 shows pretraining curves for both configurations on Textured Quadruped. Entropy reward is similar across both settings, suggesting that the particle-based coverage signal is not strongly affected by temporal context. SF reward is higher with frame stacking (0.492 vs 0.434), indicating that the additional temporal context helps the policy learn more coherent task-directed behaviour: by encoding the direction of change across consecutive frames, the concatenated representation allows ϕ to better distinguish states that are moving toward a given task direction w from those that are not, producing stronger alignment between state features and the sampled task vector. Interestingly, incidental extrinsic reward is higher without frame stacking (0.572 vs 0.525), though this does not translate to better downstream performance.

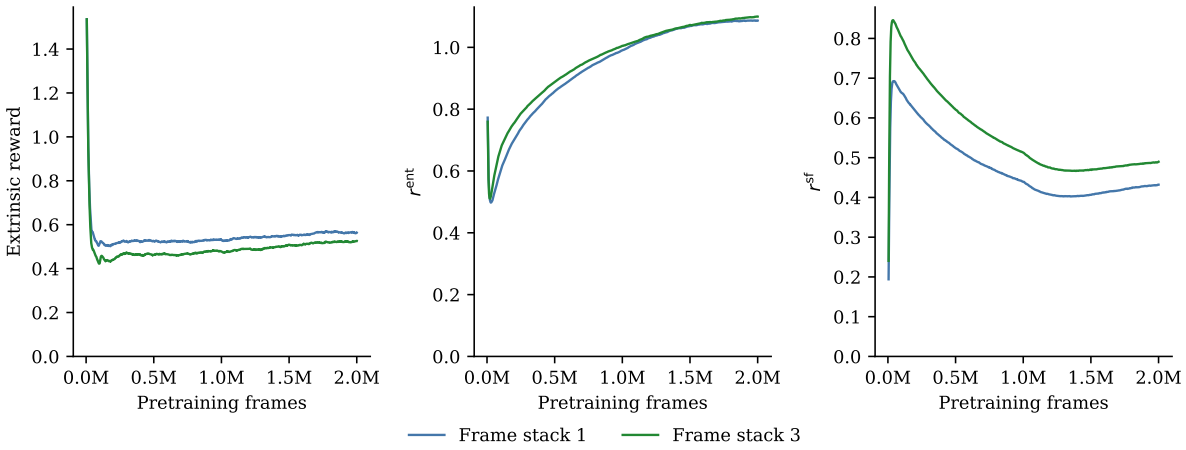


Figure 5.9: APS-CLIP pretraining curves on Textured Quadruped comparing single-frame (frame stack 1) and three-frame (frame stack 3) encodings. From left to right: extrinsic reward, entropy reward r^{ent}, and successor feature reward r^{sf}.

At finetuning, frame stacking produces substantially better performance on Stand (538 ± 140 vs 377 ± 77) and Run (132 ± 38 vs 121 ± 72). The improvement on Stand is particularly pronounced. Since Stand rewards maintaining a stable upright torso orientation, a single frame cannot distinguish a reliably upright configuration from one that is momentarily upright during a fall. Three consecutive frames encoding the same configuration signal postural stability, allowing ϕ to learn features that reward sustained balance rather than any instantaneous upright pose. These results confirm that frame stacking provides a meaningful benefit despite CLIP not being trained on sequential inputs. However, the gains are modest relative to the gap between CLIP and CNN on dynamic tasks, suggesting that frame stacking alone does not fully address the absence of motion information in CLIP's representation. More principled approaches to temporal encoding are discussed in Section 7.2.

5.5.5 Effect of Skill Space Size

The main DIAYN experiments use 16 skills, matching the URLB default. We ablate skill space sizes of 8 and 32 to test whether CLIP's embedding space supports a larger skill vocabulary without sacrificing discriminability.

As shown in Figures 5.10 and 5.11, increasing skill cardinality drives broader state coverage. The incidental extrinsic reward rises from 0.453 for 8 skills to 1.172 for 32, but at the cost of

discriminator accuracy, which falls monotonically from 92.2% to 35.3%. This reveals a tension inherent to DIAYN in high-dimensional embedding spaces: with 32 skills, the discriminator must partition a space in which many behaviours embed closely together in CLIP space, making reliable skill separation increasingly difficult. Downstream performance is highest at 8 skills (268 ± 63) and falls substantially beyond this, with 16 and 32 skills producing broadly comparable scores of 120 ± 44 and 140 ± 114 respectively. The wide overlap in standard deviations means 16 and 32 skills cannot be meaningfully distinguished, and the high variance at 32 is consistent with unreliable probe-phase selection: with a fixed probe budget of 4,096 steps divided across 32 skills, each skill receives only 128 evaluation steps, which is insufficient to reliably identify the best-performing skill. The primary takeaway is therefore not a monotonic degradation with skill cardinality but rather that URLB's [Laskin et al., 2021] probe budget imposes a practical ceiling on the number of skills that can be reliably exploited at finetuning time.

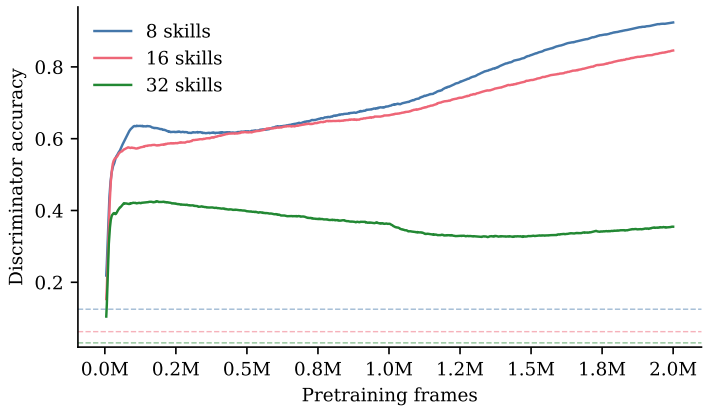


Figure 5.10: DIAYN discriminator accuracy during pretraining on Textured Quadruped for skill space sizes of 8, 16, and 32. Dashed lines indicate chance performance for each configuration.

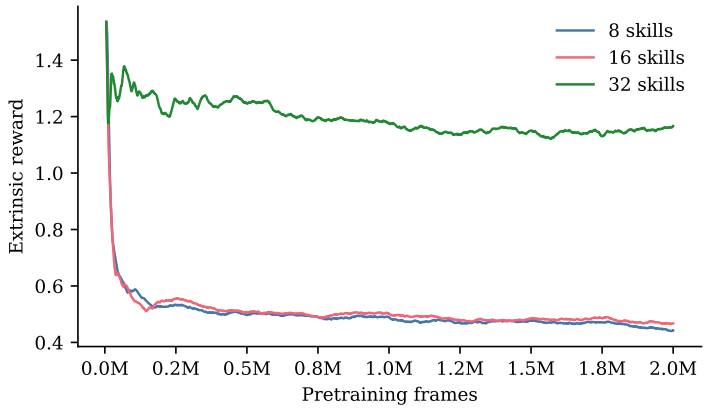


Figure 5.11: Incidental extrinsic reward during pretraining on Textured Quadruped for skill space sizes of 8, 16, and 32.

Chapter 6

Discussion

6.1 When Does CLIP Improve Over CNN?

CLIP's advantage is concentrated where skill variation is visually and semantically distinctive in a static frame, which is logical given its internet-scale pretraining on individual images.

Walker is the domain where this condition holds most strongly: standing, mid-stride, and fallen poses are separated by large distances in CLIP space, and APS-CLIP substantially outperforms APS-CNN across all four downstream tasks (Table 5.2). Incidental extrinsic reward also rises faster under CLIP during pretraining (Figure 5.3), suggesting task-relevant states are visited earlier without needing to construct a useful representation from scratch. Qualitative analysis (Section 5.4) confirms that the pretrained skills span a diverse set of visually distinct body configurations without any task-specific training.

Quadruped also improves under CLIP, though more modestly (Table 5.2). The complexity of the 3D morphology interacts poorly with the fixed skill space size: qualitative analysis (Appendix G) reveals that discovered skills cluster predominantly around variations of the agent lying on its back, with the discriminator finding subtle distinctions within a narrow region of CLIP space rather than covering the full range of possible configurations. The skill space size ablation (Section 5.5.5) suggests this is partly a consequence of URLB's default skill cardinality being too low for the breadth of behaviourally distinct states available in a 12-DoF environment.

Cheetah is the clear failure mode. All four downstream tasks require sustained directed locomotion with no static posture component, making the discriminating feature between skills temporal rather than visually salient in a static frame, which is precisely what CLIP's image-level prior does not capture.

DIAYN under CLIP. DIAYN's 16 discrete skills naturally favour static, visually distinct poses that are easiest to classify reliably. In the CNN baseline, the discriminator degrades after peaking: the encoder progressively overfits to established skill boundaries, collapsing toward a small set of easily separable features and losing sensitivity to the broader state space. Under CLIP, the discriminator makes increasingly clear distinctions throughout training, reaching near-ceiling accuracy on Walker and Cheetah and continuing to improve on Quadruped at the 2M frame horizon (Figure 5.2), consistent with CLIP's embedding space revealing separable structure as the agent explores the larger 12-DoF configuration space. However,

the skill space size ablation (Section 5.5.5) reveals a tension in high-dimensional embedding spaces: increasing skill cardinality drives broader coverage but degrades discriminator accuracy, since many behaviours embed closely together in CLIP space. Downstream performance consequently degrades with 32 skills, suggesting DIAYN's classification-based objective is fundamentally better suited to the low-dimensional proprioceptive spaces it was designed for than to high-dimensional semantic embeddings.

APS under CLIP. APS benefits more substantially from the CLIP substitution than DIAYN, which we attribute to two structural properties of its objective. First, the SF reward $\phi(e_{t+1})^\top w$ is linear in the state features, requiring task-relevant variation to fall along linear axes of ϕ. A CNN encoder must simultaneously extract meaningful features from pixels and organise them linearly whereas CLIP provides a semantically structured input in which behaviourally distinct states are already separated along interpretable directions, making the linearity assumption considerably more likely to hold. Second, the continuous skill representation allows the probe phase to interpolate between pretrained behavioural axes via least-squares regression, rather than being constrained to whichever of DIAYN's K discrete skills best matches the downstream task, which is particularly advantageous in CLIP space where the compressed geometry means the optimal downstream behaviour may lie between discrete skill boundaries. Both effects are amplified with visual realism, as confirmed by the textured environment ablation (Section 5.5.2).

Semantic entropy as a regulariser. APS-CNN produces higher raw entropy reward than APS-CLIP across almost all domains (Figure 5.4), yet achieves lower downstream performance on Walker and Quadruped. This is consistent with CNN entropy being partially consumed by semantically trivial novelty such as lighting changes, background shifts, and positional variation that register as novel in pixel space but correspond to identical behavioural states. Because APS-CLIP computes nearest-neighbour distances in a space already invariant to such perturbations, its entropy reward functions as a *semantic filter*, driving exploration toward behaviourally distinct states rather than perceptually noisy ones.

High seed variance. CLIP methods exhibit consistently higher seed variance than CNN counterparts, with finetuning standard deviations often two to three times larger. We attribute this to probe phase sensitivity: APS's continuous skill space requires identifying a specific direction on the unit sphere from only 4,096 probe steps, whereas DIAYN's argmax selects the best of 16 fixed options. Allocating more steps to the probe phase, as in the original 40k:60k split of Liu and Abbeel [2021a], would likely reduce this variance.

6.2 Static Skills and the Limits of a Fixed Prior

The results across all three domains confirm that CLIP improves over CNN when skill variation is postural and visually salient in a static frame, and underperforms when it is temporal.

On Walker and Quadruped, our CLIP agents discover semantically grounded static postures which are well-matched to downstream tasks that reward upright configurations. On Cheetah, the same static bias becomes a failure mode. DIAYN-CLIP and APS-CLIP collapse to held poses while CNN variants discover directed locomotion behaviours under the same objective (Appendix G), demonstrating that the problem is not the MI objective itself but CLIP's inability to distinguish states that differ primarily in their dynamics. Because CLIP was trained

on individual images, states that are dynamically distinct but visually similar receive nearby embeddings, leaving the diversity objective with no signal to separate them. Frame stacking provides only a weak remedy: in a 2-action-repeat environment, consecutive frames are visually similar, so the difference signal between $\widehat{e}_t^{(1)}$ and $\widehat{e}_t^{(3)}$ is small.

A further consequence of the frozen encoder is that the representation is fixed from the first training update and cannot adapt to reveal new discriminative structure as training progresses. The consequences of this are visible in the intrinsic reward curves, where both r^{ent} and r^{sf} plateau well before the 2M frame horizon (Figure 5.4), indicating that the diversity objective has exhausted its ability to generate novelty signals from the frozen embedding. However, this does not imply that the representation itself has stopped being useful. If the embedding were the binding constraint on downstream performance, we would expect task reward to plateau in lockstep with the intrinsic signal, yet as shown in Section 5.5.1, downstream performance continues to rise as a function of pretraining duration for APS-CLIP while APS-CNN plateaus early, demonstrating that the embedding retains exploitable semantic structure beyond what the diversity objective alone can surface. Intrinsic reward saturation therefore reflects the limits of the diversity objective's capacity to incentivise further exploration, not the limits of the representation itself. Because the encoder is frozen, the policy continues to improve within a fixed representational structure even after the diversity signal is exhausted. What is needed, then, is not a different encoder architecture but a richer semantic embedding space that better captures motor-control-relevant variation, allowing the full 2M frame training budget to be exploited more effectively.

A related limitation is that CLIP's metric structure reflects semantic similarity as captured by internet-scale image-text pairs rather than distances meaningful for motor control. Two quadruped gaits that differ substantially in joint velocity and physical displacement (e.g., walking versus sprinting) may receive a small cosine distance because both resemble a robot on its feet to a model trained on natural images. This is most evident in the Quadruped DIAYN results, where the embedding space does not cleanly separate behaviours that are motorically distinct but visually similar. Cosine distances between qualitatively different behaviours can be as low as 0.1–0.2 (e.g., standing versus kneeling), meaning the skill discovery algorithm operates within a compressed and potentially ill-conditioned subspace regardless of pretraining duration.

These limitations point toward concrete directions rather than fundamental barriers. CLIP's semantic prior demonstrably improves skill discovery in visually rich, postural domains, and the remaining gaps are precisely the problems that motion-aware encoders and trajectory-level diversity objectives are designed to address, as discussed in Section 7.2.

6.3 Limitations of Experimental Setup

Fixed Episode Lengths and Skill Space Size. We use DMC [Tassa et al., 2018] to match the URLB evaluation protocol [Laskin et al., 2021], which runs episodes to a fixed time limit regardless of agent behaviour. Unlike OpenAI Gym-style environments, where early termination on failure implicitly penalises falling and restricts the viable behavioural set, DMC agents are free to fall and continue, dramatically expanding the set of reachable states and making a fixed skill space increasingly inadequate. As Laskin et al. [2021] note, when the skill space is too small relative to this expanded set, the most easily classifiable skills tend to be different configurations of the agent lying on the ground which is precisely what we observe on Textured

Quadruped. Fixed-length episodes are more representative of real-world deployment conditions but result in lower absolute scores than Gym-style benchmarks, where early termination inflates performance by reducing the opportunity to accumulate zero reward.

Evaluation Conditions. Comparison between CLIP and CNN is complicated by partial overlap in evaluation conditions. CNN finetuning baselines for Walker and Quadruped are drawn from URLB [Laskin et al., 2021], which evaluated on plain DMC environments, while our CLIP agents are evaluated on textured variants. For Cheetah, where no published baseline exists, we run our own CNN experiments on the textured environment directly, providing a controlled comparison. A fully controlled comparison would require CNN finetuning baselines on textured Walker and Quadruped. The pretraining results nonetheless support the textured design choice: CLIP achieves near-perfect discriminator accuracy on textured Walker compared to CNN at 500k frames, suggesting substantially cleaner skill separation when observations are closer to CLIP's training distribution.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This dissertation asked whether frozen CLIP encoders could replace task-trained CNN encoders in unsupervised skill discovery, and whether internet-scale vision-language pretraining provides a useful semantic prior for continuous control without task-specific supervision. The answer is conditionally yes. APS-CLIP substantially outperforms its CNN counterpart on Walker and Quadruped, produces a 96% more compact representation, and eliminates encoder gradient computation entirely, all without any environment-specific annotation or reward engineering.

The conditions under which CLIP helps are specific. Gains are concentrated where skill variation is postural and visually salient, matching CLIP's static image training distribution. On Cheetah, where task success requires sustained directed locomotion, CLIP provides no advantage: its frozen encoder cannot develop the motion sensitivity that dynamic tasks require, and frame stacking is an insufficient substitute. The observation that both DIAYN and APS converge to static skills under CLIP, while CNN variants discover directed locomotion under the same objectives, confirms that this is a representational rather than an objective-level failure on dynamic domains.

Realising the full potential of semantically grounded skill discovery will require motion-aware encoders that extend CLIP's semantic structure into the temporal domain, and objectives that reward trajectory-level diversity rather than instantaneous discriminability. Both directions are tractable extensions of the framework introduced here.

7.2 Future Work

7.2.1 Beyond Static Skill Discovery: Trajectory-Level Diversity

A natural extension of this work is to replace the MI-based objectives of DIAYN and APS with a distance-maximising approach that rewards behavioural diversity over trajectories rather than instantaneous state separation. LGSD [Rho et al., 2025a] is the most promising candidate: it grounds the Lipschitz constraint of the WDM objective in cosine distance between language embeddings of consecutive states, ensuring that latent displacement reflects semantically meaningful physical change rather than arbitrary pixel difference. Replacing the LLM pipeline with a frozen CLIP encoder would eliminate the two limitations that currently prevent LGSD

from operating in pixel-based settings: the requirement for step-wise language generation and the dependence on symbolic state descriptions unavailable from raw observations. Because CLIP's cosine distance becomes large only when the semantic content of a state changes, CLIP-LGSD would reward the agent for transitioning between visually and semantically distinct body configurations, which should encourage richer locomotion behaviours than the static postures discovered by DIAYN and APS.

However, CLIP-LGSD would still be subject to the cyclic behaviour limitation (identified in Section 2.1.5) shared by all current distance-maximising methods [Park et al., 2022, 2023, Park, Rybkin and Levine, 2024] as the WDM objective measures the distance between the start and end states of a trajectory, meaning a backflip that returns the agent to its starting pose receives near-zero reward despite the semantic richness of the intermediate frames. Two directions could address this. The first is to accumulate cosine distance step-by-step along the trajectory rather than measuring endpoint displacement, rewarding the agent for every semantically distinct frame it traverses. The second is to learn an encoder that maps states to a circular latent space [Park et al., 2025], enabling periodic behaviours to be represented as closed loops rather than as zero net displacement, directly resolving the cyclic collapse that defeats endpoint-based metrics.

7.2.2 Richer Representations and More Realistic Environments

The results suggest that CLIP embeddings capture high-level semantic structure but lack the fine-grained kinematic information needed for dynamic tasks. The most direct remedy is a multimodal state representation $\phi(s) = [\text{CLIP}(s_{\text{image}}), s_{\text{proprio}}]$, concatenating the CLIP embedding with a proprioceptive vector of joint angles and velocities. This would allow the skill discovery algorithm to discriminate high-frequency kinematics while retaining CLIP's semantic prior, directly addressing the static skill problem without replacing the encoder. A complementary approach is to identify which dimensions of the CLIP embedding space carry the most behaviourally discriminative information and project onto that subspace, as Bagatella et al. [2025] suggest that useful state representations must capture both low-level dynamical information and higher-level contextual features simultaneously. The baseline prompt technique of Rocamonde et al. [2024] could also be adapted for skill discovery: defining K skill directions as vectors from a neutral prompt embedding to K semantically distinct target prompts (e.g. "a quadruped standing", "a quadruped running") would encourage skills to spread along semantically grounded axes rather than arbitrary directions in CLIP space, potentially alleviating the low cosine similarity issue identified in Section 6.

The scale ablation (Section 5.5.3) demonstrates that larger CLIP models produce higher incidental extrinsic reward during pretraining, consistent with Baumli et al. [2023], who found that reward signal quality improves with encoder scale. Finetuning CLIP on in-domain image-caption pairs represents a complementary direction: Di Palo et al. [2023] found that as few as 1,000 in-domain pairs significantly improved performance on goal-conditioned locomotion tasks. Both directions were out of scope due to computational constraints but together suggest a path toward stronger pretrained representations without task-specific supervision. The textured environment ablation (Section 5.5.2) further confirms that CLIP's advantage grows with visual realism, motivating evaluation in photorealistic environments such as Habitat [Savva et al., 2019] where observations more closely resemble CLIP's natural image training distribution. For tasks requiring precise positional discrimination such as manipulation, CLIP's limited spatial awareness becomes an additional bottleneck that spatially-aware encoders such as SAM-CLIP

[Wang et al., 2024a] and SpatialCLIP [Wang et al., 2025] are designed to address.

7.2.3 Motion-Aware Encoders

Extensions to CLIP that model temporal dynamics represent a direct remedy to the static skill problem. VideoCLIP [Xu et al., 2021] extends the contrastive objective to video-text pairs, MoCLIP [Maldonado et al., 2025] distils motion priors into the CLIP embedding space, and MINECLIP [Fan et al., 2022] demonstrates that video-language models can capture behavioural variation (running, swimming, climbing) that static image encoders miss.

7.2.4 Zero-Shot Skill Selection

The pretraining-then-finetuning paradigm used in URLB still requires a reward signal at test time. CLIP's joint image-text embedding space motivates a more ambitious direction: if the pretrained skill space is sufficiently rich, the most task-relevant skill could be selected directly via a text query, for instance, embedding the instruction "walk forward" and selecting the nearest skill vector without any finetuning (similar to Rho et al. [2025a]). This represents a potential path towards Zero-Shot RL [Touati, Rapin and Ollivier, 2022], in which an agent is deployed directly from its pretrained skill space without downstream reward supervision.

Word count: 10,984.

Bibliography

Achiam, J., Edwards, H., Amodei, D. and Abbeel, P., 2018. Variational option discovery algorithms [Online]. 1807.10299, Available from: https://arxiv.org/abs/1807.10299.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J. and Mané, D., 2016. Concrete problems in ai safety [Online]. 1606.06565, Available from: https://arxiv.org/abs/1606.06565.

Andres, A. and Del Ser, J., 2024. Fostering intrinsic motivation in reinforcement learning with pretrained foundation models. *arxiv preprint arxiv:2410.07404*.

Aubret, A., Matignon, L. and Hassas, S., 2019. A survey on intrinsic motivation in reinforcement learning. *arxiv preprint arxiv:1908.06976*.

Badia, A.P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z.D. and Blundell, C., 2020. Agent57: Outperforming the Atari human benchmark [Online]. In: H.D. III and A. Singh, eds. *Proceedings of the 37th international conference on machine learning*. PMLR, *Proceedings of Machine Learning Research*, vol. 119, pp.507–517. Available from: https://proceedings.mlr.press/v119/badia20a.html.

Bagatella, M., Pirotta, M., Touati, A., Lazaric, A. and Tirinzoni, A., 2025. Td-jepa: Latent-predictive representations for zero-shot reinforcement learning. *arxiv preprint arxiv:2510.00739*.

Barber, D. and Agakov, F., 2004. The im algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16(320), p.201.

Baumli, K., Baveja, S., Behbahani, F., Chan, H., Comanici, G., Flennerhag, S., Gazeau, M., Holsheimer, K., Horgan, D., Laskin, M. et al., 2023. Vision-language models as a source of rewards. *arxiv preprint arxiv:2312.09187*.

Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., Arx, S. von, Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J.Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D.E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P.W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X.L., Li, X., Ma, T., Malik, A., Manning, C.D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J.C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J.S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J.,

Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A.W., Tramèr, F., Wang, R.E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S.M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K. and Liang, P., 2022. On the opportunities and risks of foundation models [Online]. 2108.07258, Available from: https://arxiv.org/abs/2108.07258.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D., 2020. Language models are few-shot learners [Online]. 2005.14165, Available from: https://arxiv.org/abs/2005.14165.

Burda, Y., Edwards, H., Storkey, A. and Klimov, O., 2018. Exploration by random network distillation [Online]. 1810.12894, Available from: https://arxiv.org/abs/1810.12894.

Campos, V., Trott, A., Xiong, C., Socher, R., Nieto, X.G. i and Torres, J., 2020. Explore, discover and learn: Unsupervised discovery of state-covering skills [Online]. 2002.03647, Available from: https://arxiv.org/abs/2002.03647.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P. and Joulin, A., 2021. Emerging properties in self-supervised vision transformers [Online]. 2104.14294, Available from: https://arxiv.org/abs/2104.14294.

Christiano, P., Leike, J., Brown, T.B., Martic, M., Legg, S. and Amodei, D., 2023. Deep reinforcement learning from human preferences [Online]. 1706.03741, Available from: https://arxiv.org/abs/1706.03741.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2019. Bert: Pre-training of deep bidirectional transformers for language understanding [Online]. 1810.04805, Available from: https://arxiv.org/abs/1810.04805.

Di Palo, N., Byravan, A., Hasenclever, L., Wulfmeier, M., Heess, N. and Riedmiller, M., 2023. Towards a unified agent with foundation models. *arxiv preprint arxiv:2307.09668*.

Eysenbach, B., Gupta, A., Ibarz, J. and Levine, S., 2018. Diversity is all you need: Learning skills without a reward function [Online]. 1802.06070, Available from: https://arxiv.org/abs/1802.06070.

Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.A., Zhu, Y. and Anandkumar, A., 2022. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in neural information processing systems*, 35, pp.18343–18362.

Fang, A., Ilharco, G., Wortsman, M., Wan, Y., Shankar, V., Dave, A. and Schmidt, L., 2022. Data determines distributional robustness in contrastive language image pre-training (clip). *International conference on machine learning*. PMLR, pp.6216–6234.

Firoozi, R., Tucker, J., Tian, S., Majumdar, A., Sun, J., Liu, W., Zhu, Y., Song, S., Kapoor, A., Hausman, K. et al., 2025. Foundation models in robotics: Applications, challenges, and the future. *The international journal of robotics research*, 44(5), pp.701–739.

Gregor, K., Rezende, D.J. and Wierstra, D., 2016. Variational intrinsic control [Online]. 1611.07507, Available from: https://arxiv.org/abs/1611.07507.

Gu, S., Holly, E., Lillicrap, T. and Levine, S., 2016. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates [Online]. 1610.00633, Available from: https://arxiv.org/abs/1610.00633.

Gupta, T., Karkus, P., Che, T., Xu, D. and Pavone, M., 2022. Foundation models for semantic novelty in reinforcement learning. *arxiv preprint arxiv:2211.04878*.

Haarnoja, T., Hartikainen, K., Abbeel, P. and Levine, S., 2018. Latent space policies for hierarchical reinforcement learning. *International conference on machine learning*. PMLR, pp.1851–1860.

Hansen, S., Dabney, W., Barreto, A., Wiele, T.V. de, Warde-Farley, D. and Mnih, V., 2020. Fast task inference with variational intrinsic successor features [Online]. 1906.05030, Available from: https://arxiv.org/abs/1906.05030.

Hussonnois, M., Karimpanal, T.G. and Rana, S., 2025. Human-aligned skill discovery: Balancing behaviour exploration and alignment [Online]. 2501.17431, Available from: https://arxiv.org/abs/2501.17431.

Kim, H., Lee, B., Lee, H., Hwang, D., Kim, D. and Choo, J., 2025. Do's and don'ts: Learning desirable skills with instruction videos [Online]. 2406.00324, Available from: https://arxiv.org/abs/2406.00324.

Kim, S., Kwon, J., Lee, T., Park, Y. and Perez, J., 2022. Safety-aware unsupervised skill discovery. *2023 international conference on robotics and automation (icra)*. IEEE.

Klyubin, A., Polani, D. and Nehaniv, C., 2005. Empowerment: a universal agent-centric measure of control [Online]. *2005 ieee congress on evolutionary computation*. vol. 1, pp.128–135 Vol.1. Available from: https://doi.org/10.1109/CEC.2005.1554676.

Langosco, L.L. di, Koch, J., Sharkey, L.D., Pfau, J. and Krueger, D., 2021. Goal misgeneralization in deep reinforcement learning [Online]. *International conference on machine learning*. Available from: https://api.semanticscholar.org/CorpusID:249954130.

Laskin, M., Liu, H., Peng, X.B., Yarats, D., Rajeswaran, A. and Abbeel, P., 2022. Cic: Contrastive intrinsic control for unsupervised skill discovery [Online]. 2202.00161, Available from: https://arxiv.org/abs/2202.00161.

Laskin, M., Yarats, D., Liu, H., Lee, K., Zhan, A., Lu, K., Cang, C., Pinto, L. and Abbeel, P., 2021. Urlb: Unsupervised reinforcement learning benchmark. *arxiv preprint arxiv:2110.15191*.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. and Wierstra, D., 2019. Continuous control with deep reinforcement learning [Online]. 1509.02971, Available from: https://arxiv.org/abs/1509.02971.

Liu, H. and Abbeel, P., 2021a. Aps: Active pretraining with successor features [Online]. 2108.13956, Available from: https://arxiv.org/abs/2108.13956.

Liu, H. and Abbeel, P., 2021b. Behavior from the void: Unsupervised active pre-training. *Advances in neural information processing systems*, 34, pp.18459–18473.

Ma, Y.J., Liang, W., Wang, G., Huang, D.A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L. and Anandkumar, A., 2024. Eureka: Human-level reward design via coding large language models [Online]. 2310.12931, Available from: https://arxiv.org/abs/2310.12931.

Maldonado, G., Pazho, A.D., Noghre, G.A., Katariya, V. and Tabkhi, H., 2025. Moclip: Motion-aware fine-tuning and distillation of clip for human motion generation [Online]. 2505.10810, Available from: https://arxiv.org/abs/2505.10810.

Mu, J., Zhong, V., Raileanu, R., Jiang, M., Goodman, N., Rocktäschel, T. and Grefenstette, E., 2022. Improving intrinsic exploration with language abstractions. *Advances in neural information processing systems*, 35, pp.33947–33960.

Nguyen, H., Motzoi, F., Metcalf, M., Whaley, K., Bukov, M. and Schmitt, M., 2024. Reinforcement learning pulses for transmon qubit entangling gates. *Machine learning: Science and technology* [Online], 5. Available from: https://doi.org/10.1088/2632-2153/ad4f4d.

Oudeyer, P.Y. and Kaplan, F., 2008. How can we define intrinsic motivation? *the 8th international conference on epigenetic robotics: Modeling cognitive development in robotic systems*. Lund University Cognitive Studies, Lund: LUCS, Brighton.

Park, J., Cho, D., Lee, J., Shim, D., Jang, I. and Kim, H.J., 2025. Periodic skill discovery. *arxiv preprint arxiv:2511.03187*.

Park, S., Choi, J., Kim, J., Lee, H. and Kim, G., 2022. Lipschitz-constrained unsupervised skill discovery [Online]. 2202.00914, Available from: https://arxiv.org/abs/2202.00914.

Park, S., Lee, K., Lee, Y. and Abbeel, P., 2023. Controllability-aware unsupervised skill discovery [Online]. 2302.05103, Available from: https://arxiv.org/abs/2302.05103.

Park, S., Rybkin, O. and Levine, S., 2024. Metra: Scalable unsupervised rl with metric-aware abstraction [Online]. 2310.08887, Available from: https://arxiv.org/abs/2310.08887.

Pathak, D., Agrawal, P., Efros, A.A. and Darrell, T., 2017. Curiosity-driven exploration by self-supervised prediction. *International conference on machine learning*. PMLR, pp.2778–2787.

Polanyi, M., 2009. *The tacit dimension*. Cited by: 166. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85071808588&partnerID=40&md5=6893368ba93dc5ce845de31349b4a16c.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. and Sutskever, I., 2021. Learning transferable visual models from natural language supervision [Online]. 2103.00020, Available from: https://arxiv.org/abs/2103.00020.

Rajaraman, A., Myers, V. and Dragan, A., 2024. Discovering skills with language.

Rho, S., Smith, L., Li, T., Levine, S., Peng, X.B. and Ha, S., 2025a. Language guided skill discovery [Online]. 2406.06615, Available from: https://arxiv.org/abs/2406.06615.

Rho, S., Trinh, A., Xu, D. and Ha, S., 2025b. Reference grounded skill discovery [Online]. 2510.06203, Available from: https://arxiv.org/abs/2510.06203.

Rocamonde, J., Montesinos, V., Nava, E., Perez, E. and Lindner, D., 2024. Vision-language models are zero-shot reward models for reinforcement learning [Online]. 2310.12921, Available from: https://arxiv.org/abs/2310.12921.

Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J.,

Koltun, V., Malik, J. et al., 2019. Habitat: A platform for embodied ai research. *Proceedings of the ieee/cvf international conference on computer vision*. pp.9339–9347.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T. and Silver, D., 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature* [Online], 588(7839), p.604–609. Available from: `https://doi.org/10.1038/s41586-020-03051-4`.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M. et al., 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35, pp.25278–25294.

Shah, R., Krasheninnikov, D., Alexander, J., Abbeel, P. and Dragan, A., 2019. Preferences implicit in the state of the world [Online]. `1902.04198`, Available from: `https://arxiv.org/abs/1902.04198`.

Sharma, A., Gu, S., Levine, S., Kumar, V. and Hausman, K., 2020. Dynamics-aware unsupervised discovery of skills [Online]. `1907.01657`, Available from: `https://arxiv.org/abs/1907.01657`.

Shi, L.X., Lim, J.J. and Lee, Y., 2022. Skill-based model-based reinforcement learning [Online]. `2207.07560`, Available from: `https://arxiv.org/abs/2207.07560`.

Singh, A., Bhaskar, A., Yu, P., Chakraborty, S., Dasyam, R., Bedi, A. and Tokekar, P., 2025. Varp: Reinforcement learning from vision-language model feedback with agent regularized preferences. *arxiv preprint arxiv:2503.13817*.

Sontakke, S.A., Zhang, J., Arnold, S.M.R., Pertsch, K., Bıyık, E., Sadigh, D., Finn, C. and Itti, L., 2023. Roboclip: One demonstration is enough to learn robot policies [Online]. `2310.07899`, Available from: `https://arxiv.org/abs/2310.07899`.

Stout, A., Konidaris, G.D. and Barto, A.G., 2005. Intrinsically motivated reinforcement learning: A promising framework for developmental robot learning.

Sutton, R.S. and Barto, A.G., 2018. *Reinforcement learning: An introduction* [Online]. 2nd ed. The MIT Press. Available from: `http://incompleteideas.net/book/the-book-2nd.html`.

Tam, A., Rabinowitz, N., Lampinen, A., Roy, N.A., Chan, S., Strouse, D., Wang, J., Banino, A. and Hill, F., 2022. Semantic exploration from language abstractions and pretrained representations. *Advances in neural information processing systems*, 35, pp.25377–25389.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D.d.L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A. et al., 2018. Deepmind control suite. *arxiv preprint arxiv:1801.00690*.

Touati, A., Rapin, J. and Ollivier, Y., 2022. Does zero-shot reinforcement learning exist? *arxiv preprint arxiv:2209.14935*.

Turner, A.M., Ratzlaff, N. and Tadepalli, P., 2020. Avoiding side effects in complex environments [Online]. `2006.06547`, Available from: `https://arxiv.org/abs/2006.06547`.

Wang, H., Vasu, P.K.A., Faghri, F., Vemulapalli, R., Farajtabar, M., Mehta, S., Rastegari, M., Tuzel, O. and Pouransari, H., 2024a. Sam-clip: Merging vision foundation models towards

semantic and spatial understanding. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. pp.3635–3647.

Wang, X., Lee, K., Hakhamaneshi, K., Abbeel, P. and Laskin, M., 2021. Skill preferences: Learning to extract and execute robotic skills from human feedback [Online]. 2108.05382, Available from: https://arxiv.org/abs/2108.05382.

Wang, Y., Sun, Z., Zhang, J., Xian, Z., Biyik, E., Held, D. and Erickson, Z., 2024b. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback [Online]. 2402.03681, Available from: https://arxiv.org/abs/2402.03681.

Wang, Z., Zhou, S., He, S., Huang, H., Yang, L., Zhang, Z., Cheng, X., Ji, S., Jin, T., Zhao, H. et al., 2025. Spatialclip: Learning 3d-aware image representations from spatially discriminative language. *Proceedings of the computer vision and pattern recognition conference*. pp.29656–29666.

Xiao, X., Liu, J., Wang, Z., Zhou, Y., Qi, Y., Jiang, S., He, B. and Cheng, Q., 2025. Robot learning in the era of foundation models: A survey. *Neurocomputing*, 638, p.129963.

Xie, T., Zhao, S., Wu, C.H., Liu, Y., Luo, Q., Zhong, V., Yang, Y. and Yu, T., 2024. Text2reward: Reward shaping with language models for reinforcement learning [Online]. 2309.11489, Available from: https://arxiv.org/abs/2309.11489.

Xu, H., Ghosh, G., Huang, P.Y., Okhonko, D., Aghajanyan, A., Metze, F., Zettlemoyer, L. and Feichtenhofer, C., 2021. Videoclip: Contrastive pre-training for zero-shot video-text understanding [Online]. 2109.14084, Available from: https://arxiv.org/abs/2109.14084.

Yang, Z., Moerland, T.M., Preuss, M., Plaat, A., François-Lavet, V. and Hu, E.S., 2025. Guiding skill discovery with foundation models [Online]. 2510.23167, Available from: https://arxiv.org/abs/2510.23167.

Yarats, D., Fergus, R., Lazaric, A. and Pinto, L., 2021. Mastering visual continuous control: Improved data-augmented reinforcement learning [Online]. 2107.09645, Available from: https://arxiv.org/abs/2107.09645.

Yu, D., Goodman, N.D. and Mu, J., 2023. Characterizing tradeoffs between teaching via language and demonstrations in multi-agent systems. *arxiv preprint arxiv:2305.11374*.

Appendix A

Derivation of the DIAYN Variational Lower Bound

Direct maximisation of $I(S;Z)$ is intractable because the conditional entropy term requires access to the true posterior $p(z|s)$. By definition:

$$H(Z|S) = -E_{s\sim\pi(z),z\sim p(z)}[\log p(z|s)] \tag{A.1}$$

Substituting into the mutual information decomposition:

$$I(S;Z) = H(Z) + E_{s,z}[\log p(z|s)] \tag{A.2}$$

To render $\log p(z|s)$ tractable, we introduce a variational approximation $q_\phi(z|s)$ parameterised by a neural network discriminator. The non-negativity of the KL divergence gives:

$$D_{KL}(p(z|s) \parallel q_\phi(z|s)) = E_{s,z}[\log p(z|s) - \log q_\phi(z|s)] \geq 0 \tag{A.3}$$

Rearranging:

$$E_{s,z}[\log p(z|s)] \geq E_{s,z}[\log q_\phi(z|s)] \tag{A.4}$$

Substituting into the mutual information expression yields the tractable variational lower bound:

$$I(S;Z) \geq H(Z) + E_{z\sim p(z),\, s\sim\pi(z)}[\log q_\phi(z|s)] \tag{A.5}$$

Maximising this bound is equivalent to minimising the cross-entropy loss of the discriminator q_ϕ, with the term inside the expectation serving directly as the agent's intrinsic reward.

Appendix B

Derivation of the APS Exploitation Reward

Following Liu and Abbeel [2021a], we derive the exploitation term of the APS intrinsic reward. The goal is to optimise the intractable conditional entropy, $-H(S \mid Z)$. By the variational lower bound of Barber and Agakov [2004]:

$$\begin{aligned} -H(S \mid Z) &= \sum_{s,z} p(s,z) \log p(s \mid z) \\ &= \sum_{s,z} p(s,z) \log q(s \mid z) + \sum_{z} p(z) D_{\mathrm{KL}}(p(\cdot \mid z) \| q(\cdot \mid z)) \\ &\geq E_{s,z}[\log q(s \mid z)], \end{aligned} \tag{B.1}$$

where the terms are defined as follows:

- S and s: the states induced by the policy.
- Z and z: the task variables, which are drawn from a fixed distribution.
- $p(s,z)$ and $p(s \mid z)$: the true joint and conditional probability distributions of the states and task variables.
- $q(s \mid z)$: a variational approximation of the true, intractable conditional distribution.
- D_{KL}: the Kullback-Leibler divergence, measuring the difference between the true distribution and the variational approximation.
- $E_{s,z}$: the expected value over the sampled states and task variables.

The inequality holds since $D_{\mathrm{KL}} \geq 0$. Setting $z \equiv w$ and choosing $q(s \mid w)$ to be a Von Mises-Fisher distribution with scale parameter 1, the log-likelihood evaluates to:

$$\log q(s \mid w) = \phi(s)^{\top} w, \tag{B.2}$$

where:

- w: the task vector, which is ℓ_2-normalised.

- $\phi(s)$: the ℓ_2-normalised state feature representation of state s.

This yields the exploitation reward used in practice:

$$r_{\text{APS}}^{\text{exploitation}}(s, a, s', w) = \phi(s')^\top w \tag{B.3}$$

where:

- $r_{\text{APS}}^{\text{exploitation}}$: the intrinsic exploitation reward given to the agent for transitioning from state s to next state s' via action a, conditioned on task vector w.

The lower bound in Equation B.1 is then tightened during training by minimising the negative log-likelihood over the encoder parameters:

$$\mathcal{L} = -E_{s,w}[\phi(s)^\top w] \tag{B.4}$$

where:

- $\mathcal{L}$: the negative log-likelihood loss function.

Appendix C

Implementation Details and Hyperparameters

Table C.1: Hyperparameters used in all experiments (excluding ablations). CNN values follow the URLB pixel-based configuration [Laskin et al., 2021]. CLIP values match the CNN configuration except where explicitly justified; differences are in bold.

Hyperparameter	CNN	CLIP
Replay buffer capacity	10^6	10^6
Action repeat	2	2
Seed frames	4,000	4,000
n-step returns	3	3
Mini-batch size	256	256
Discount (γ)	0.99	0.99
Optimiser	Adam	Adam
Learning rate	10^{-4}	10^{-4}
Agent update frequency	2	2
Critic target EMA (τ_Q)	0.01	0.01
Hidden dim.	1,024	1,024
Exploration stddev clip	0.3	0.3
Exploration stddev value	0.2	0.2
Pretraining frames	2×10^6	2×10^6
Finetuning frames	1×10^5	1×10^5
Render resolution	84×84	$\mathbf{224 \times 224}$
Encoder output dim.	39,200	**1,536**
Feature dim.	50	**256**

DIAYN hyperparameter	Value
Skill dim	16
Skill sampling frequency (steps)	50
Discriminator net arch.	$512 \to 1024 \to 1024 \to 16$ ReLU MLP
APS hyperparameter	**Value**
Reward transformation	$\log(r + 1.0)$
Successor feature dim.	10
Successor feature net arch.	$\lvert\mathcal{O}\rvert \to 1024 \to 1024 \to 10$ ReLU MLP
k in NN	12
Avg top k in NN	True
Least square batch size	4096

Table C.2: Algorithm-specific hyperparameters for DIAYN and APS, shared across CNN and CLIP variants unless otherwise stated.

Encoder resolution. All agents render observations at 84×84 pixels, consistent with the URLB baseline [Laskin et al., 2021]. For CLIP-based agents, observations are bicubically upscaled to 224×224 prior to encoding, matching CLIP's native pretraining resolution. CNN-based agents receive the raw 84×84 renders directly, consistent with the DrQ-v2 architecture for which this resolution was designed. Rendering at 84×84 for both encoder types ensures that any performance differences are attributable to the encoder architecture rather than differences in observation fidelity.

Batch size. Following Laskin et al. [2021], all pixel-based experiments use a mini-batch size of 256, consistent across both CNN and CLIP conditions. Although CLIP-based agents store compact embedding vectors rather than pixel tensors in the replay buffer, we match the CNN batch size to ensure fair comparison of downstream performance between encoder conditions.

Temporal context. As described in Section 3.1.3, each CLIP-based observation consists of three frames encoded independently and concatenated to form a 1,536-dimensional representation e_t. CNN-based agents receive all three frames as a stacked $(9, 84, 84)$ tensor, processed jointly by the convolutional encoder.

Data augmentation. CNN-based agents apply random shift augmentation to pixel observations sampled from the replay buffer, following Yarats et al. [2021], regularising the actor and critic by discouraging overfitting to exact pixel positions. CLIP-based agents do not apply augmentation: since embeddings are cached at collection time, raw pixel observations are unavailable at update time. This represents an intentional design decision rather than a limitation. Random shift augmentation is a spatial regulariser designed to compensate for the sensitivity of randomly initialised CNN encoders to pixel-level variation, a sensitivity that does not arise in frozen CLIP embeddings, which are already invariant to the kinds of low-level visual perturbations that augmentation is designed to handle [Fang et al., 2022]. We therefore argue that augmentation is unnecessary for CLIP-based agents and that its absence does not

disadvantage the CLIP condition relative to CNN.

Embedding caching. Since CLIP weights are frozen throughout training, the embedding $e_t \in R^{1536}$ is a deterministic function of the observation $o_t \in R^{3\times224\times224}$, i.e. $e_t = f_{\text{CLIP}}(o_t)$ where f_{CLIP} is fixed. We therefore pre-compute and store embeddings at environment collection time, replacing raw pixel tensors in the replay buffer with their 1,536-dimensional representations. This reduces CLIP forward passes from $\mathcal{O}(B \cdot U)$ per environment step during training to $\mathcal{O}(1)$, where B is the replay buffer batch size and U is the number of gradient updates per environment step, removing the vision transformer from the gradient update loop entirely.

Compute. All experiments were run on NVIDIA RTX 3090 GPUs. Pretraining a single CLIP-based agent for 2×10^6 frames takes approximately 72 hours; CNN-based agents run at a similar wall-clock rate, as the reduced gradient computation cost of the frozen encoder is offset by the inference latency of the ViT-B/32 forward pass. Finetuning a single seed on a single downstream task takes approximately 6 hours. Full results across three seeds and all tasks required approximately 720 GPU-hours in total, excluding ablation and CNN pretraining runs.

Feature dimension. The trunk feature dimension d_{feat} controls the size of the learned projection applied within the actor and critic networks before their policy and value heads. URLB specifies $d_{\text{feat}} = 50$ for pixel-based agents, calibrated for the CNN encoder where the trunk simultaneously serves as a dimensionality reduction layer from the 39,200-dimensional convolutional output. For CLIP-based agents the encoder already outputs a compact 1,536-dimensional embedding, making an aggressive bottleneck of 50 dimensions potentially lossy with respect to the semantic structure already present in the representation.

We ablate $d_{\text{feat}} \in \{50, 256, 1536\}$ under APS-CLIP on Textured Quadruped. Incidental extrinsic reward at the end of pretraining increases with feature dimension: 0.445 for $d_{\text{feat}} = 50$, 0.525 for 256, and 0.556 for 1536. The improvement from 256 to 1536 is marginal (0.030), and comes at the cost of substantially larger actor and critic networks and increased memory consumption, a meaningful constraint given the GPU budget available for this work. We therefore adopt $d_{\text{feat}} = 256$ as a pragmatic trade-off between representational capacity and computational tractability. The skill discovery objectives operate directly on the full 1,536-dimensional embedding upstream of the trunk and are unaffected by this choice.

Appendix D

Finetuning Approach for DIAYN and APS

DIAYN

During the probe phase, `update_meta` cycles through all K=16 discrete skills in sequence, allocating approximately 256 environment steps (out of the 4,096 budget) to each. The replay buffer therefore contains transitions labelled with the active skill and the associated extrinsic reward. `regress_meta` computes the mean extrinsic reward received under each skill and selects the one-hot vector corresponding to the highest-performing skill, which is then held fixed for the remainder of finetuning. The quality of this selection depends entirely on how well the 16 pretrained skills partition the state space: if the discriminator has learned to associate distinct, semantically coherent state distributions with each skill index, the probe can reliably identify the skill whose distribution overlaps most with the downstream task. With a CNN encoder, the discriminator must simultaneously learn to extract task-relevant features from raw pixels and maintain 16 separable state distributions, limiting the semantic coherence of the resulting partition. With CLIP, the embedding space is already organised into semantically distinct regions by contrastive pretraining on internet-scale image-text pairs, so the discriminator only needs to find linear boundaries between pre-existing clusters. The probe therefore encounters more behaviourally distinct skills, making it more likely that one of the 16 aligns closely with the downstream task.

APS

During the probe phase, `update_meta` resamples the task vector $\mathbf{w}$ from the unit sphere every five steps, producing dense coverage of the d_{SF}=10 dimensional successor feature space. `regress_meta` then solves the ordinary least-squares problem

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \|\Phi\mathbf{w} - \mathbf{r}\|^2, \tag{D.1}$$

where $\Phi \in R^{B\times 10}$ is the matrix of SF vectors collected during the probe and $\mathbf{r} \in R^B$ is the corresponding extrinsic reward vector. The solution $\mathbf{w}^*$, normalised to the unit sphere, identifies the direction in SF space most correlated with the downstream reward. Because the particle entropy objective during pretraining organises ϕ's dimensions around the principal axes of

behavioural variation, the regression can identify which of those axes co-varies most strongly with the downstream reward. Since the pretrained policy already maximises $\phi(s)^\top \mathbf{w}$ for any $\mathbf{w}$, setting $\mathbf{w} = \mathbf{w}^*$ immediately directs it toward task-relevant behaviour, providing a warm start for the finetuning phase.

The quality of $\mathbf{w}^*$ depends on how well the axes of ϕ align with task-relevant dimensions of variation. When ϕ is trained on top of CNN features, both the feature extraction and the successor feature decomposition must be learned simultaneously from pixels, which limits the extent to which ϕ's axes correspond to semantically meaningful quantities. When ϕ is trained on top of CLIP embeddings, the input space is already structured so that semantically distinct states (a walking robot and a fallen robot, for instance) are placed far apart, and semantically similar states are placed close together. The SF network can therefore focus entirely on finding the right linear combinations of pre-existing semantic axes rather than constructing them from scratch. The regression in Equation D.1 consequently finds a $\mathbf{w}^*$ that more accurately captures the task-relevant direction from the same 4,096 probe transitions, translating directly into a better-initialised finetuning phase.

Evaluation episodes are conducted every 10,000 frames throughout finetuning. At each checkpoint, 10 episodes are rolled out with the actor in deterministic mode. The reported downstream score is the mean episode reward at the *final* evaluation checkpoint at frame 10^5, measuring stable final performance rather than peak performance at any point during training.

Appendix E

Reward Function Specifications

This appendix documents the reward signal for each downstream task, derived from the dm_control implementations used in our experiments. All per-step rewards lie in $[0, 1]$; episode rewards are the undiscounted sum over all steps and can therefore reach a maximum of 1000.

Cheetah

Episodes run for 10 seconds (1000 environment steps).

- **Run.** Proportional to forward horizontal torso speed, saturating at 1 above $10\,\mathrm{m/s}$.
- **Run backward.** Proportional to backward horizontal torso speed, saturating at 1 above $10\,\mathrm{m/s}$.
- **Flip.** Proportional to forward angular momentum of the torso about the y-axis, saturating at 1 above $5\,\mathrm{kg\,m^2\,s^{-1}}$.
- **Flip backward.** Proportional to backward angular momentum of the torso about the y-axis, saturating at 1 above $5\,\mathrm{kg\,m^2\,s^{-1}}$.

Walker

Episodes run for 25 seconds (1000 environment steps). All Walker tasks share a stand reward r_{stand}, which rewards maintaining torso height above $1.2\,\mathrm{m}$ and an upright orientation. Locomotion tasks multiply this by a velocity component.

- **Stand.** $r = r_{\text{stand}}$.
- **Walk.** $r = r_{\text{stand}} \cdot f(\dot{x})$, where $f(\dot{x})$ rewards forward speed, saturating above $1\,\mathrm{m/s}$.
- **Run.** $r = r_{\text{stand}} \cdot f(\dot{x})$, where $f(\dot{x})$ rewards forward speed, saturating above $8\,\mathrm{m/s}$.
- **Flip.** $r = r_{\text{stand}} \cdot f(L_y)$, where $f(L_y)$ rewards angular momentum about the y-axis, saturating above $5\,\mathrm{kg\,m^2\,s^{-1}}$.

Quadruped

Episodes run for 20 seconds (1000 environment steps). All Quadruped tasks share an upright reward r_{upright}, which rewards keeping the torso z-axis aligned with the global z-axis. Locomotion tasks multiply this by a movement component.

- **Stand.** $r = r_{\text{upright}}$.
- **Walk.** $r = r_{\text{upright}} \cdot f(v_x)$, where $f(v_x)$ rewards forward torso speed, saturating above $0.5\,\text{m/s}$.
- **Run.** $r = r_{\text{upright}} \cdot f(v_x)$, where $f(v_x)$ rewards forward torso speed, saturating above $5\,\text{m/s}$.
- **Jump.** $r = r_{\text{upright}} \cdot f(h)$, where $f(h)$ rewards centre-of-mass height, saturating above $1.0\,\text{m}$.

Appendix F

Individual Numerical Results

		Pre-training for 2×10^6 environment steps															
		DIAYN CNN				DIAYN CLIP				APS CNN				APS CLIP			
Domain	Task	S1	S2	S3	Mean ± Std	S1	S2	S3	Mean ± Std	S1	S2	S3	Mean ± Std	S1	S2	S3	Mean ± Std
Textured Walker	Stand	—	—	—	298 ± 71	154	141	175	157 ± 17	—	—	—	172 ± 10	288	300	322	$\mathbf{303 \pm 17}$
	Walk	—	—	—	132 ± 19	87	113	117	106 ± 16	—	—	—	37 ± 5	218	237	238	$\mathbf{231 \pm 11}$
	Run	—	—	—	45 ± 3	53	73	54	60 ± 11	—	—	—	30 ± 3	100	73	90	$\mathbf{88 \pm 14}$
	Flip	—	—	—	114 ± 21	119	141	153	138 ± 17	—	—	—	38 ± 9	337	240	237	$\mathbf{271 \pm 57}$
Textured Quadruped	Stand	—	—	—	426 ± 40	107	85	169	120 ± 44	—	—	—	331 ± 43	457	458	700	$\mathbf{538 \pm 140}$
	Walk	—	—	—	$\mathbf{184 \pm 23}$	56	53	81	63 ± 15	—	—	—	146 ± 24	129	142	229	167 ± 54
	Run	—	—	—	$\mathbf{190 \pm 18}$	56	60	98	71 ± 23	—	—	—	156 ± 24	175	104	117	132 ± 38
	Jump	—	—	—	$\mathbf{262 \pm 22}$	171	113	135	140 ± 29	—	—	—	199 ± 29	224	285	244	251 ± 31
Textured Cheetah	Run	229	278	130	$\mathbf{212 \pm 75}$	86	80	104	90 ± 12	260	132	192	195 ± 64	4	116	0	40 ± 66
	Flip	562	499	655	572 ± 78	502	415	459	459 ± 44	633	513	680	$\mathbf{609 \pm 86}$	555	356	588	500 ± 126
	Run Backward	311	244	303	286 ± 37	114	101	94	103 ± 10	374	431	430	$\mathbf{412 \pm 33}$	2	169	2	58 ± 96
	Flip Backward	623	498	575	565 ± 63	505	503	499	502 ± 3	701	686	726	$\mathbf{704 \pm 20}$	524	17	639	393 ± 331

Table F.1: Per-seed fine-tuning performance after 2×10^6 pre-training steps. S1, S2, S3 denote individual random seeds. The highest mean score for each task is shown in bold. CNN baselines for Textured Walker and Textured Quadruped are taken from Laskin et al. [2021] and per-seed results are therefore unavailable (—). CNN baselines for Textured Cheetah are from our own runs. Aggregate mean and standard deviation are reported in Table 5.2.

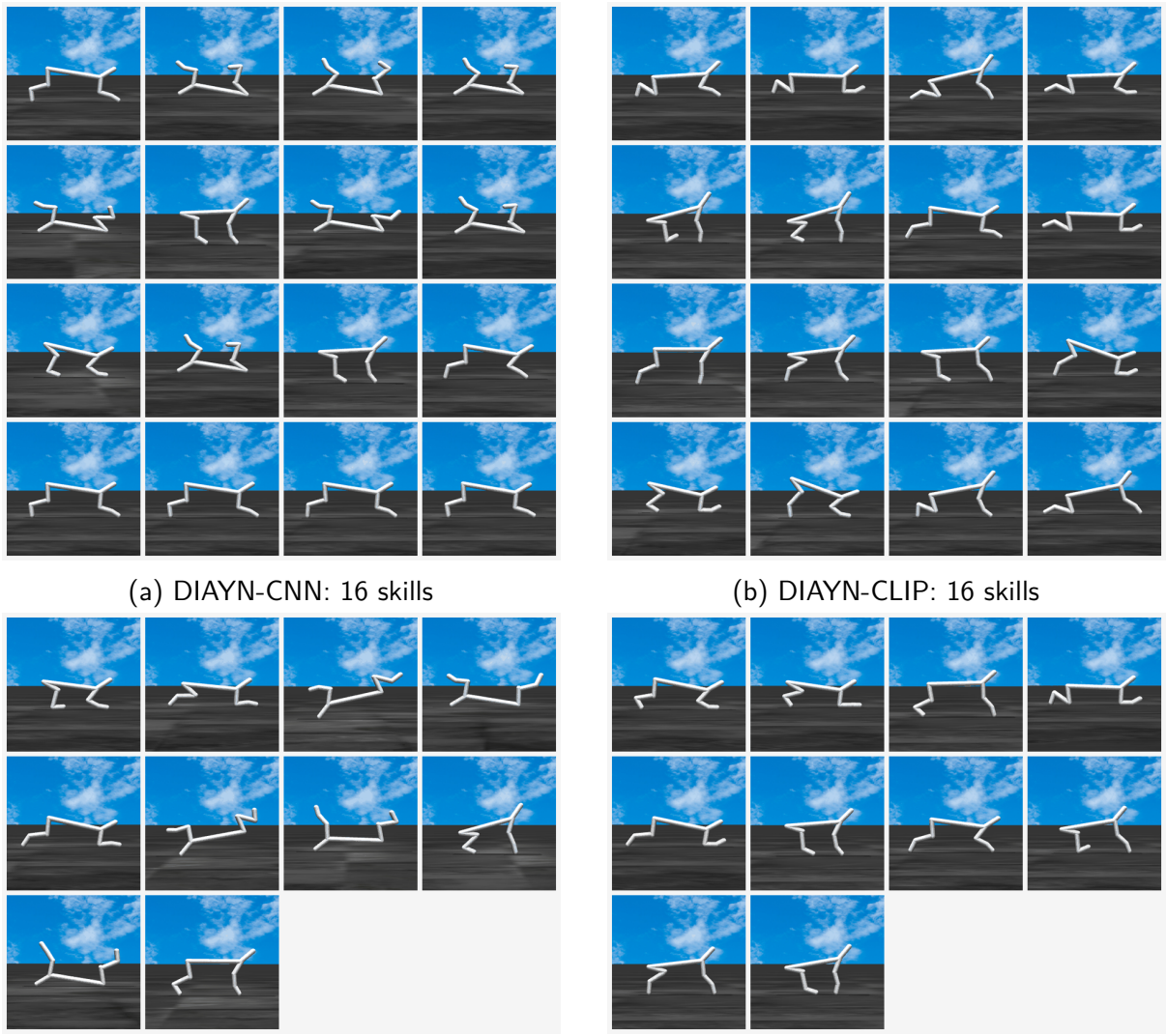
Appendix G

Extended Qualitative Analysis: Textured Cheetah and Textured Quadruped

In addition to the qualitative analysis of Textured Walker in Section 5.4, we analyse the skills discovered on Textured Quadruped and Textured Cheetah.

Textured Cheetah

Textured Cheetah is a planar 2D environment and the simplest morphology in our benchmark, yet it represents CLIP's most pronounced failure mode. Unlike Walker and Quadruped, all four Cheetah downstream tasks require sustained directed locomotion (there is no static posture task) meaning that useful skills must capture dynamic rather than postural variation. As shown in Figure G.1, DIAYN-CNN and APS-CNN discover behaviourally distinct skills including running and backflipping configurations, consistent with the CNN encoder learning task-relevant temporal features through gradient updates. DIAYN-CLIP and APS-CLIP, by contrast, produce predominantly static poses despite the dynamic nature of the tasks, with many skills appearing visually similar at the final frame. This is consistent with CLIP's image-level pretraining providing no motion signal: without temporal information, the embedding space cannot separate behaviours that differ primarily in their dynamics rather than their instantaneous appearance. This failure mode is discussed further in Section 6.



(a) DIAYN-CNN: 16 skills

(b) DIAYN-CLIP: 16 skills

(c) APS-CNN: 10 skills

(d) APS-CLIP: 10 skills

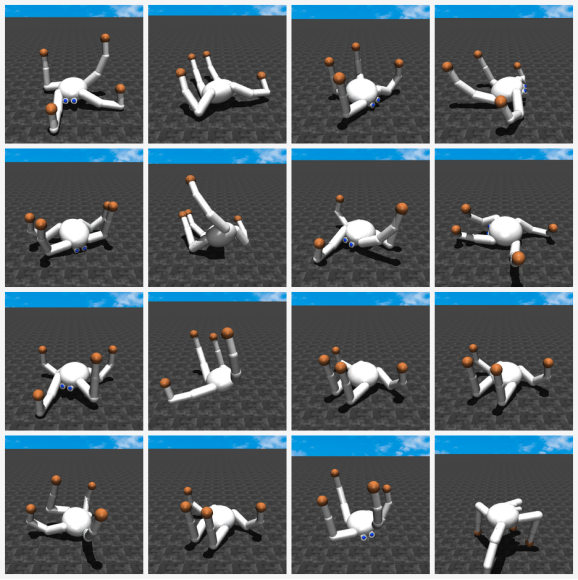
Figure G.1: Final-frame collages of discovered skills on Textured Cheetah after 2M pretraining frames. Each cell shows the terminal state of a single skill rollout. CNN skills are shown at 1M pretraining frames due to memory constraints. Skills are 0-indexed from top-left to bottom-right.

Textured Quadruped

Textured Quadruped is the most morphologically complex environment in our benchmark, with 12 DoF and three-dimensional dynamics. As noted in Section 6.3, the fixed-length episode structure of DMC means the agent is never penalised for falling, and the set of reachable states is correspondingly large. As shown in Figure G.2, this manifests clearly in the qualitative results: across all four methods, the majority of discovered skills correspond to the quadruped in various collapsed or fallen configurations, consistent with the observation of Laskin et al. [2021] that small skill spaces on DMC tend to discover different configurations of the agent lying on the ground.

Within this shared failure mode, meaningful differences between methods are still visible. DIAYN-CNN produces the most erratic terminal states, suggesting that the CNN discriminator

separates skills through unstable trajectories rather than stable poses. DIAYN-CLIP produces more stable held configurations, confirming that CLIP's semantic structure anchors skills to visually distinct postures even in a collapsed state. APS-CNN and APS-CLIP both converge to more stable terminal states than their DIAYN counterparts, consistent with the successor feature objective encouraging longer-horizon behavioural coherence.



(a) DIAYN-CNN: 16 skills



(b) DIAYN-CLIP: 16 skills



(c) APS-CNN: 10 skills



(d) APS-CLIP: 10 skills

Figure G.2: Final-frame collages of discovered skills on Textured Quadruped after 2M pretraining frames. Each cell shows the terminal state of a single skill rollout. CNN skills are shown at 500k pretraining frames due to memory constraints. Skills are 0-indexed from top-left to bottom-right.