Pre-Training Goal-based Models for Sample-Efficient Reinforcement Learning

Haoqi Yuan , Zhancun Mu , Feiyang Xie , Zongqing Lu
School of Computer Science, Peking University
Yuanpei College, Peking University
Beijing Academy of Artificial Intelligence

Presented as an Oral paper at ICLR 2024 (Accepted, 8/6/8)
https://openreview.net/forum?id=o2IEmeLL9r

2025.03.07
TaeYoon Kwack
njj05043@g.skku.edu

Table of Contents

분석 중심으로 리뷰하라는 말씀을 반영하여 생략

Motivation

1. **저수준 정책의 제한된 확장성**
   - 기존 연구들은 주로 저차원 RL 환경[1] 또는 제한된 로봇 도메인[2]에서 저수준 정책을 사전 학습하는 방법을 연구했으나, 고차원, 복잡한 환경[3]과 대규모 데이터셋[4]에는 적용이 어렵다.

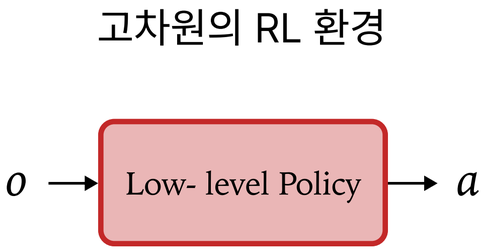
2. **Variational Inference 기반 저수준 스킬 모델의 한계**
   - 기존 방법[5][6][7]은 Variational Inference를 활용하여 연속된 잠재 변수(latent variables)로 스킬을 모델링하지만, 액션 시퀀스의 길이와 액션 공간이 큰 환경(예: Minecraft)에서는 불가능하다.

3. **샘플 효율성 문제**
   - RL을 고차원 연속 액션 공간에서 학습하는 것은 샘플 효율성이 낮다[8].
   - 기존 연구에서 다양한 차원축소가 시도 되었으나, 축소된 차원을 통해 다양한 행동을 학습하는 데 한계가 있었다. (해당 연구도 비슷한 현상이 발생하지만 저수준 정책을 통해 해결)

4. **대규모 행동 데이터셋을 활용한 한계**
   - Transformer 아키텍처 기반의 행동 복제(behavior cloning) 방법[9][10]은 Minecraft 같은 환경에서 다양한 행동을 효과적으로 모델링할 수 있지만, RL과 결합하여 목표를 생성하는 방식은 연구되지 않았다.

Motivation: Pre-Training Goal-based Models (PTGM)

기존 방식



- **Low sample efficiency**
- **Unstable learning process**

PTMG

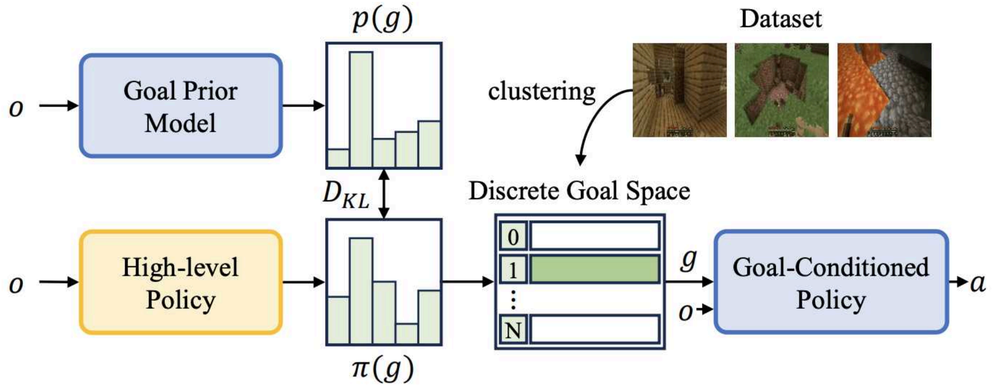
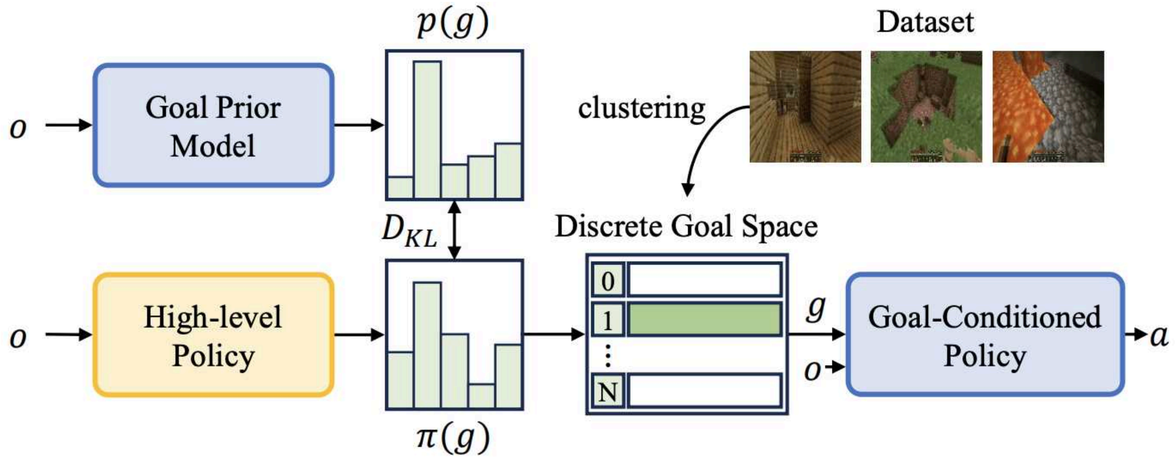


Figure 1: Overview of PTGM. A goal-conditioned policy and a goal prior model are pre-trained on the large task-agnostic dataset. The goals in the dataset are clustered into a discrete goal space. In downstream RL tasks, we train a high-level policy that outputs discrete actions to provide goals for the goal-conditioned policy, and regularize it with the KL divergence to the goal prior model.

- **High sample efficiency**
- **High interpretability**
- **Stable learning process**
- **Generalization of Low-level skills**

Method: Pre-Training Goal-based Models (PTGM)

Overview

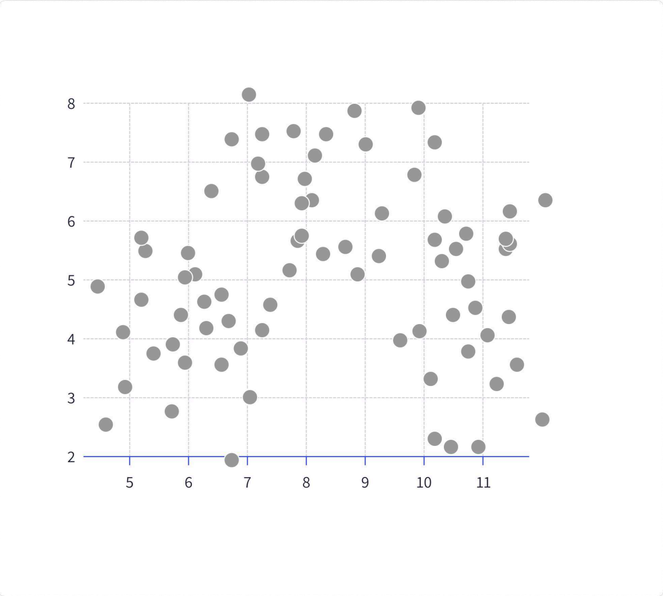


Policy / Model	Formula
Goal-conditioned policy	$P_\phi(a\|s,g)$
High-level policy	$\pi_\theta(g\|s)$
Goal prior model	$\pi_p^\psi(g\|s)$

- **Goal state (g) :** 목표 g 는 에이전트가 특정 시간 내에 도달해야 하는 상태로, state space를 클러스터링을 통해 일반화된 형태의 이산화된 state이다.
- **Goal-conditioned policy:** 주어진 목표 g 에 따라 환경에서 저수준 행동 a 를 수행하는 정책
- **High-level policy:** 강화학습을 통해 학습된 정책으로, 고수준 목표 g 를 생성하여 low-level 정책을 조정
- **Goal prior model:** 데이터셋을 통해 현재 state에서 가까운 g를 고르는 사전학습 모델로 high-level policy 가 급진적인 g를 고르지 않도록 정규화

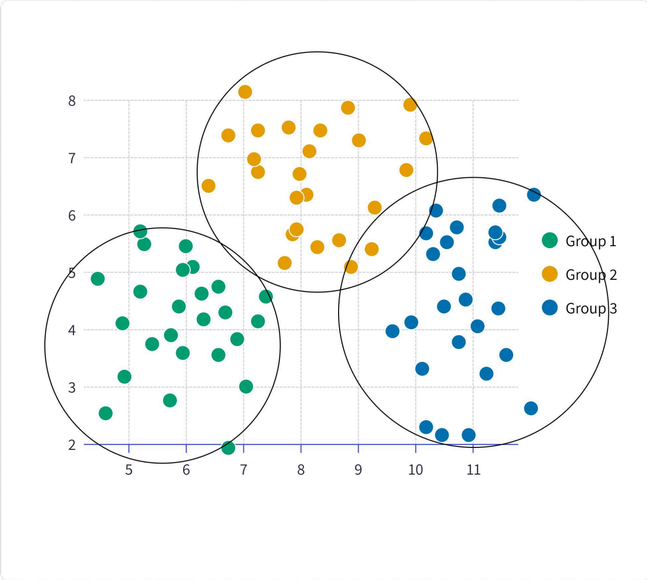
Method: Pre-Training Goal-based Models (PTGM)

1. Defining Goal States



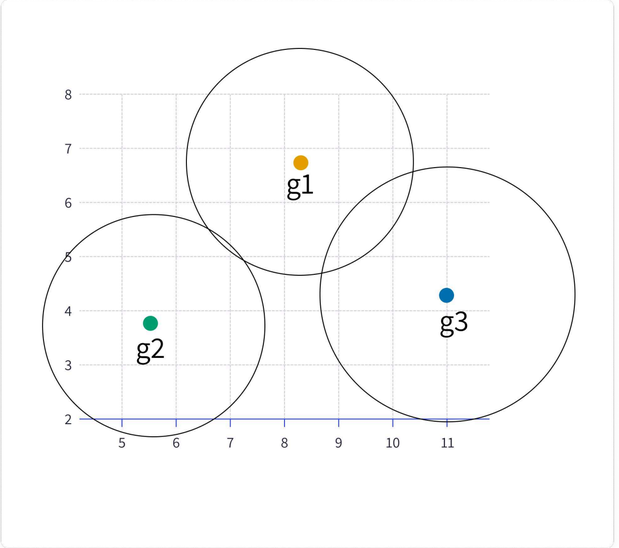
1. t-sne

t-sne 를 통해 고차원 데이터를 해석이 편한 저차원으로 변환



2. KNN

비슷한 State들을 일반화하기 위해 State를 clustering



3. Define centroids as goal states

형성된 cluster의 중심이 되는 state들을 비슷한 행동들이 일반화된 형태인 goal들로 설정

∴ Goal state는 연속적인 state space에서
데이터 셋을 클러스터링을 통해 얻은 일반화되고 이산적인 state들

Method: Pre-Training Goal-based Models (PTGM)

2. RL of High-level Policy

High-level policy 의 목적함수

$$J(\theta) = \mathbb{E}\pi_\theta \left[\sum_{t=0}^{\infty} \gamma^t \left(\sum_{i=kt}^{(k+1)t} R(s_i, a_i) - \alpha D_{\mathrm{KL}} \left(\pi_\psi^p(a^h|s_{kt}) \| \pi_\theta(a^h|s_{kt}) \right) \right) \right], \quad (4)$$

Goal prior Model의 확률분포

High-level policy의 확률분포

KL발산을 통해 두 확률분포가 비슷하도록 정규화

Goal prior Model

$$a^h = \arg\max_{i \in [N]} \left(\frac{s_i^g \cdot s^g}{\|s_i^g\| \cdot \|s^g\|} \right)$$

$$\mathcal{L}(\psi) = \mathbb{E}_D \left[-\log \pi_\psi^p(a^h|s_t) \right]. \quad (3)$$

현재 state와 goal state들간의 코사인 유사도를 기반으로 확률분포를 생성.
(비슷하면 확률 증가/ 다르면 확률 감소)

비슷한 goal state에 대하여 확률을 증가시키는 방향으로 학습된 신경망 모델

∴ High-level Policy는 reward를 최대화하도록 학습됨과 동시에 근처에 있는 goal state를 고르도록 정규화됨.

Method: Pre-Training Goal-based Models (PTGM)

3. Pretraining Goal-conditioned Policy

기존의 (s, a) → s 들로 이루어진 데이터셋을

- 결과(s)를 Goal-conditioned Policy 의 목표(g 또는 s^g)
- 이전 상태 (s)를 Goal-conditioned Policy의 관찰(o 또는 s)
- 행동(a)를 Goal-conditioned Policy의 행동(a)

로 변형해서 Goal-conditioned Policy를 학습시키기 위한 (o,g) → a 데이터 셋을 구성한다.
이후 아래의 오차함수를 통해 학습.

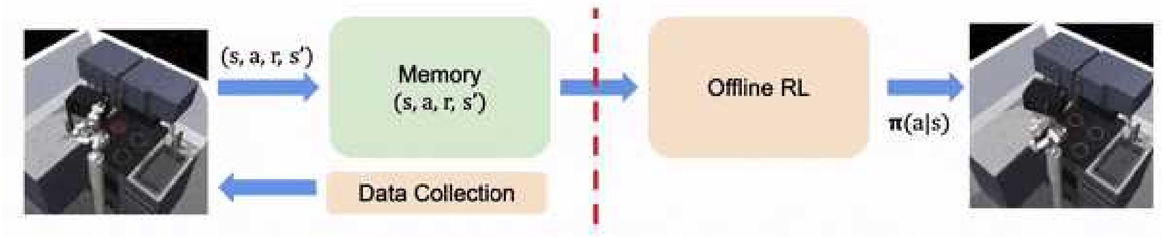
$$\mathcal{L}(\phi) = \mathbb{E}_D \left[-\log P_\phi(a_i | s_i, s^g) \right]. \quad (2)$$

∴ **Goal-conditioned Policy (Low-level policy)는 기존 데이터셋으로 사전학습된 모델로 RL을 통해 업데이트 되지 않는다.**

Experiment Settings: Benchmark

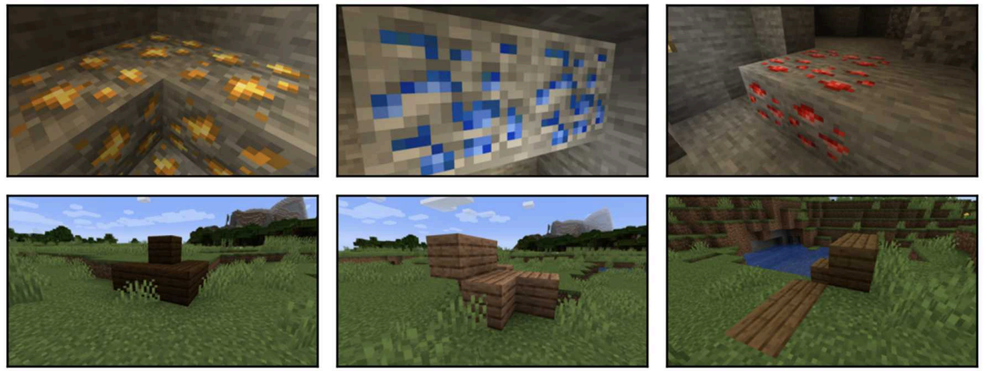
Kitchen[11]

- **환경:** 로봇 팔 (7-DOF) 기반의 시뮬레이션 환경
- **행동:** 로봇팔의 관절과 압력 등(Continuous action space)
- **데이터셋 크기:** 약 100K 프레임
- **에피소드 스텝:** 평균 280~400 스텝 (짧은 시퀀스)
- **보상:** subtask 실행 이후 성공/실패에 따른 binary reward
- 기존의 goal prior model에서의 KL 발산(KL divergence)을 SPIRAL[14] 논문에서 제안된 SAC(Soft Actor Critic)로 대체



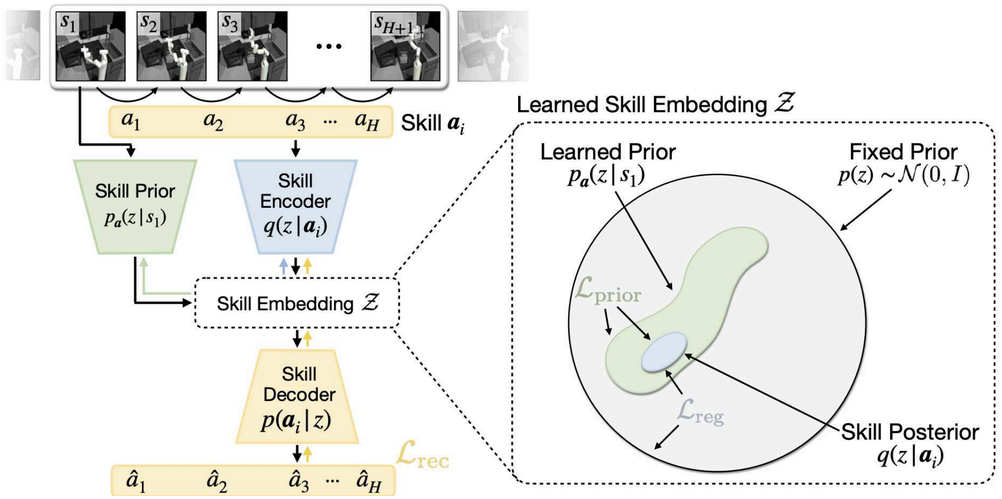
Minecraft[15]

- **환경:** 오픈월드(Open-world) 환경
- **행동:** 이동 및 상호작용 (1641 X 121 discrete action space)
- **데이터셋 크기:** 약 70M 프레임 (VPT 데이터셋 기준)
- **에피소드 스텝:** 평균 6000~8000 스텝 (장기 시퀀스)
- **보상:** MineCLIP Reward[16] 와 성공/실패에 따른 binary reward
- MineCLIP 임베딩과 PPO (Proximal Policy Optimization)[17]를 사용해 학습



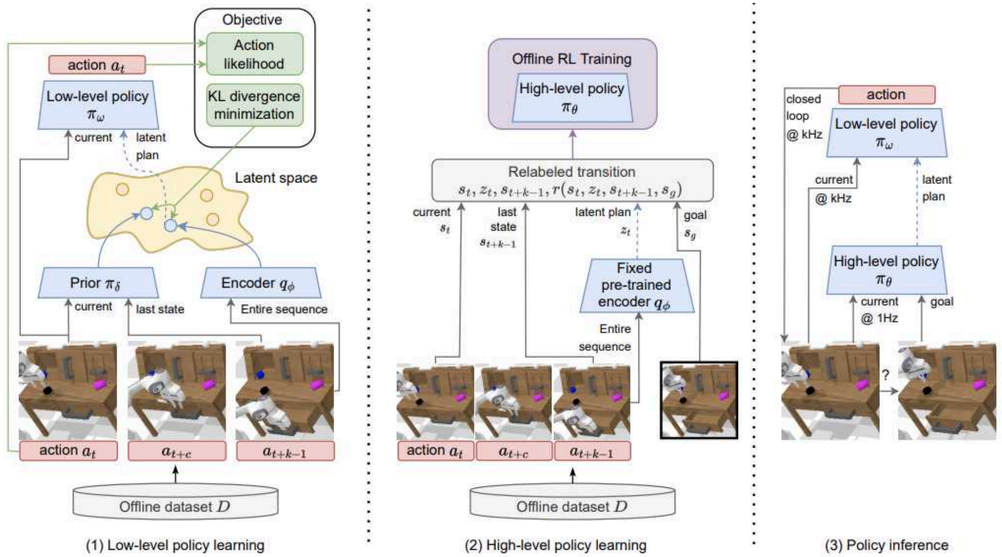
Experiment Settings: Baseline

SPiRL(Skill-based Policy in RL)[21]



Function	Description
$q(z\|s_t, a_{t:t+k})$	Skill Encoder
$p(a_{t:t+k}\|s_t, z)$	Action Decoder
$p_a(z\|s_t)$	Skill Prior

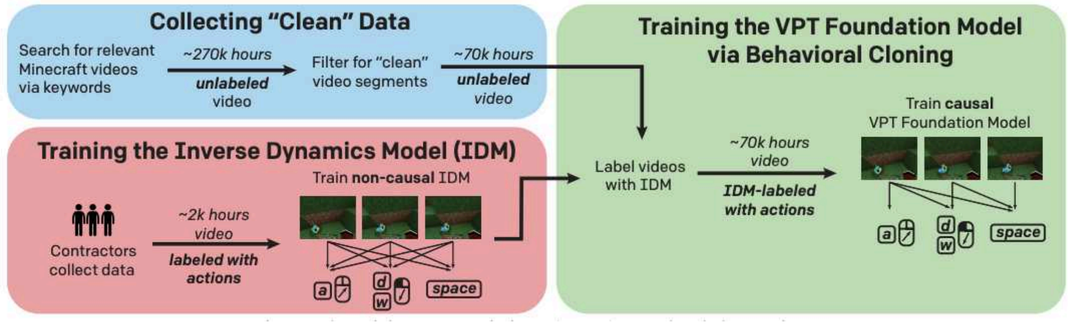
TACO(Trajectory-conditioned RL)[22]



Function	Description
$\pi(a_t\|s_t, z)$	Low-level policy
$q(z\|\tau)$	Skill Posterior
$p(z\|s_t, s_T)$	Skill Prior

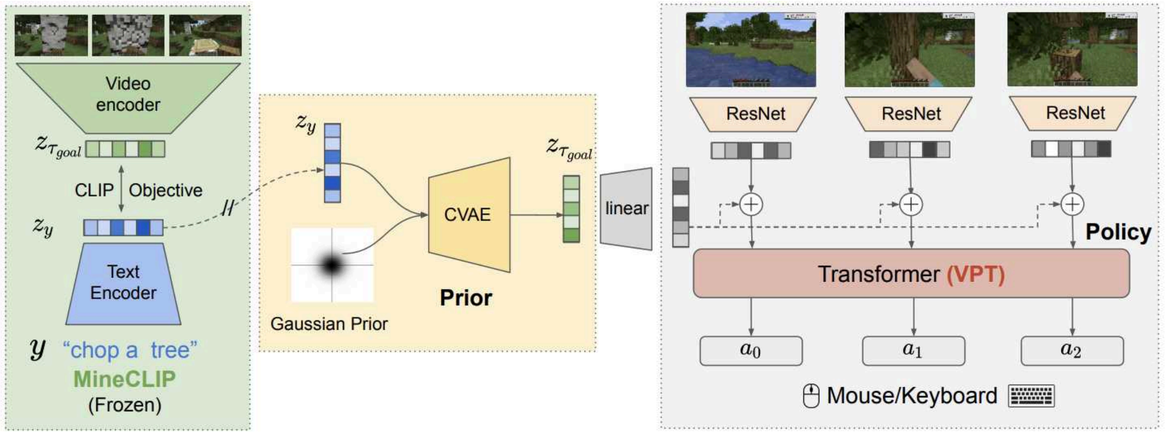
Experiment Settings: Baseline

VPT-finetune[23]
(Video PreTraining Fine-tuning)



Function	Description
$\pi(a_t \mid o_{0:t})$	Behavior Cloning (BC) Policy

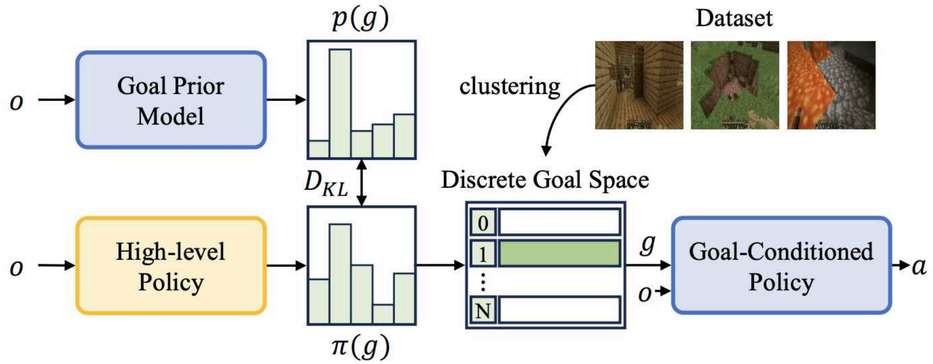
Steve-1[24]



Function	Description
$\pi(a_t \mid o_{0:t}, g)$	Goal-conditioned Policy

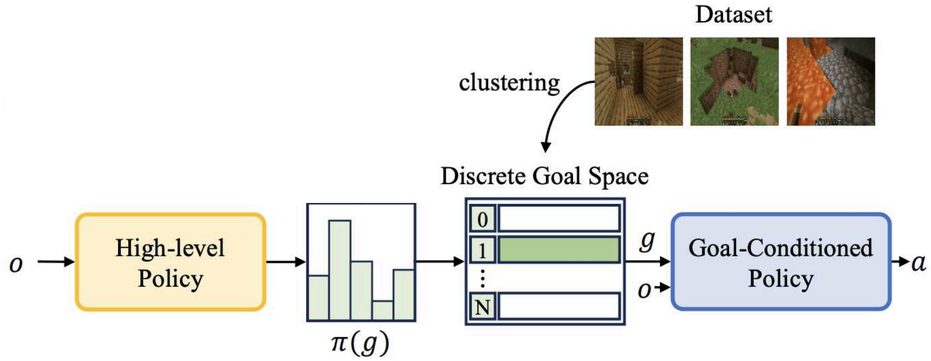
Experiment Settings: Proposed Method

PTGM
(Pre-Training Goal-based Models)



Policy / Model	Formula
Goal-conditioned policy	$P_\phi(a\|s,g)$
High-level policy	$\pi_\theta(g\|s)$
Goal prior model	$\pi_p^\psi(g\|s)$

PTGM-no-prior



Function	Description
$P_\phi(a\|s,g)$	Goal-conditioned Policy
$\pi_\theta(g\|s)$	High-level Policy

PTGM과 유사하지만, goal prior model 없이 강화학습 수행

Experiment Results

Main Results

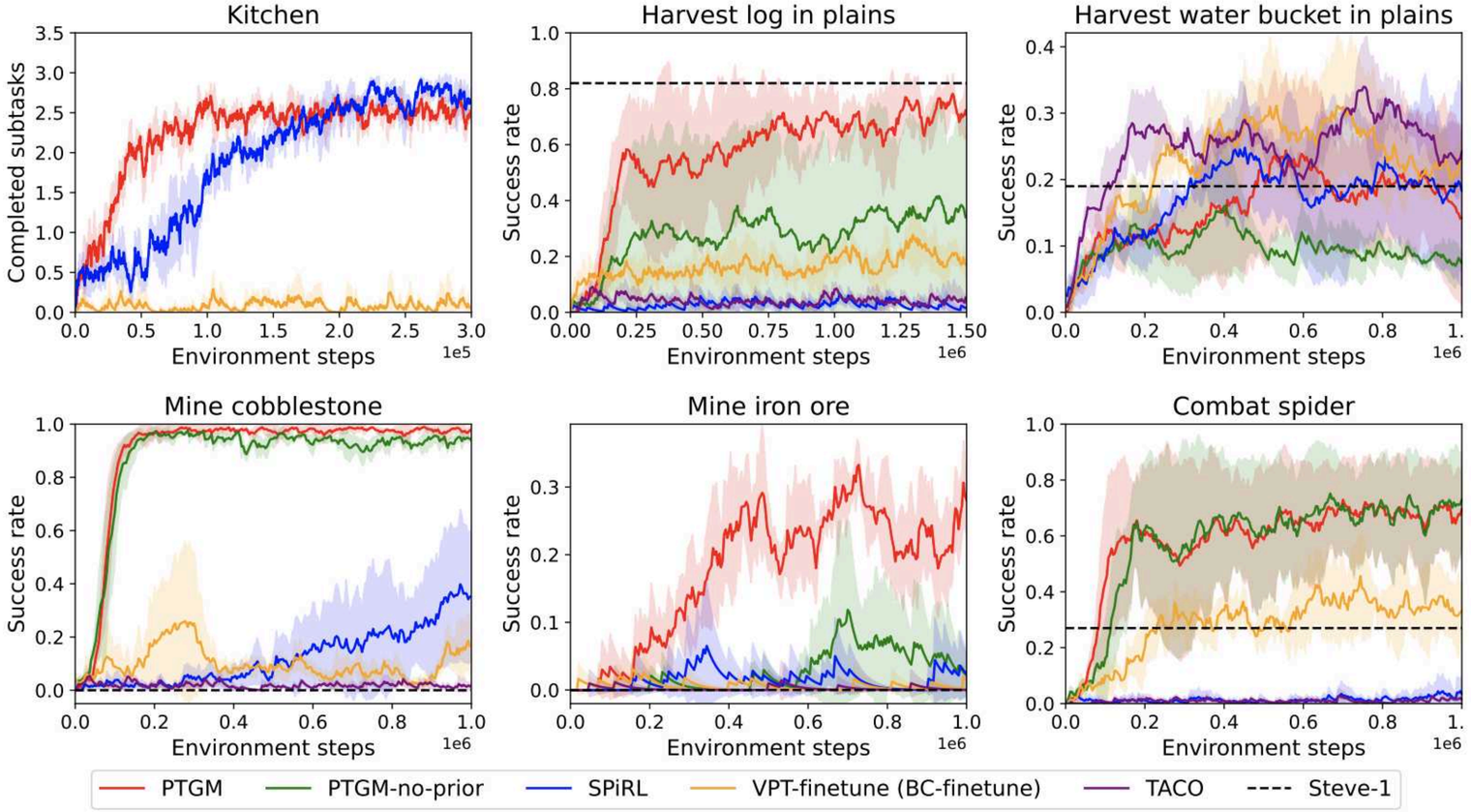


Figure 2: Training performance of PTGM against the baselines in all the tasks. The vertical axis indicates the number of completed subtasks in Kitchen and the success rate in the 5 Minecraft tasks.

거의 모든 영역에서 PTGM이 우수한 성능을 보임

Experiment Results: Kitchen

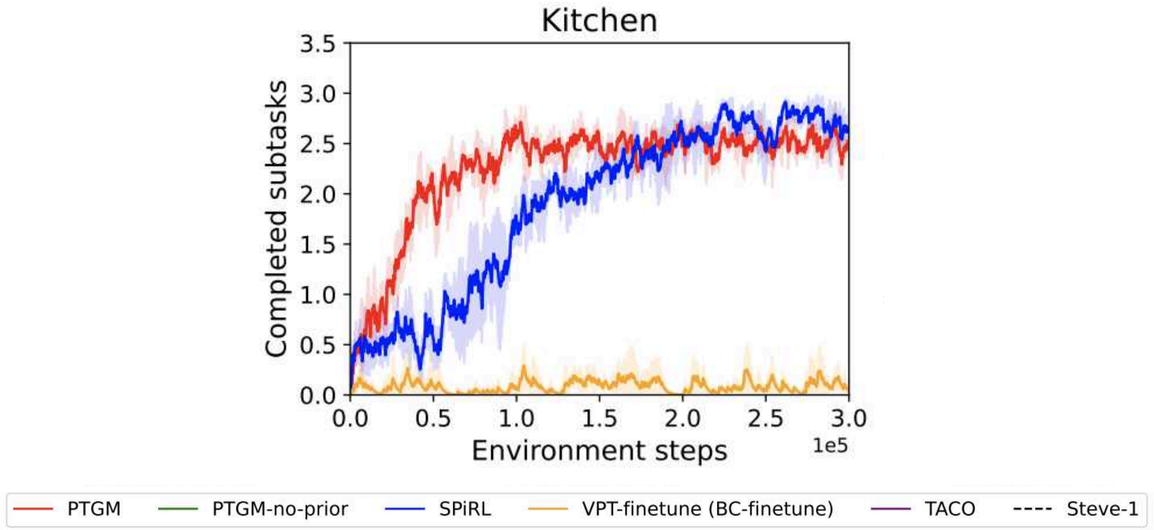


Figure 2: Training performance of PTGM against the baselines in all the tasks. The vertical axis indicates the number of completed subtasks in Kitchen and the success rate in the 5 Minecraft tasks.

PTGM과 SPiRL이 압도적인 성능을 보임 → **시간적 추상화(temporal abstraction)**가 강화학습의 샘플 효율성을 크게 향상시킴을 시사

Experiment Results: Minecraft

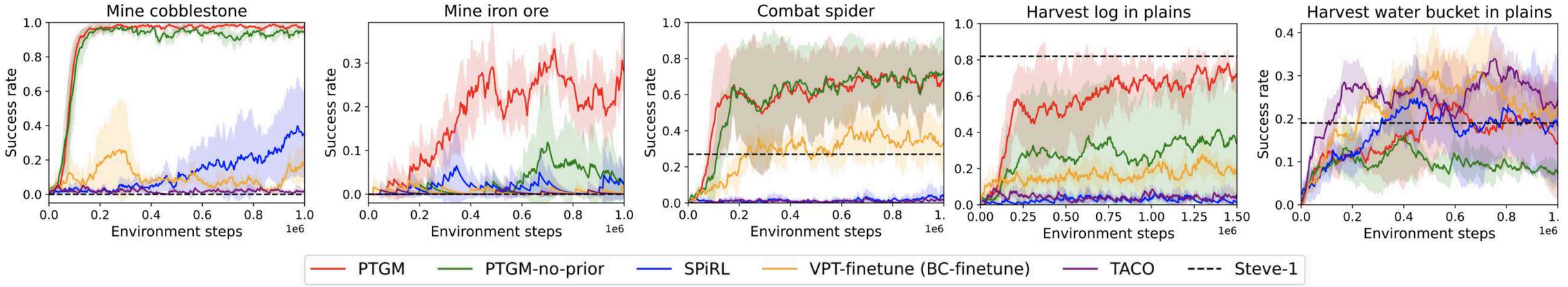


Figure 2: Training performance of PTGM against the baselines in all the tasks. The vertical axis indicates the number of completed subtasks in Kitchen and the success rate in the 5 Minecraft tasks.

- **PTGM:** 1M 환경 스텝에서 모든 작업에서 높은 성공률을 달성
- **VPT-finetune:** Cobblestone과 Iron-ore task에서 실패 ← RL 학습 중에 같은 동작을 반복하는 스킬을 잊어버리는 현상 때문으로 추정
- **Steve-1:** 나무 자르기(cut trees) 작업에서 강한 성능을 보였으나, 나머지 4개 작업에서는 PTGM보다 성능이 떨어짐
- **SPiRL:** Minecraft의 대규모 데이터셋 및 고차원 행동 공간에서 성능이 저하 (VAE가 긴 행동 시퀀스를 복원하는 데 어려움을 겪기 때문)

Ablation Study

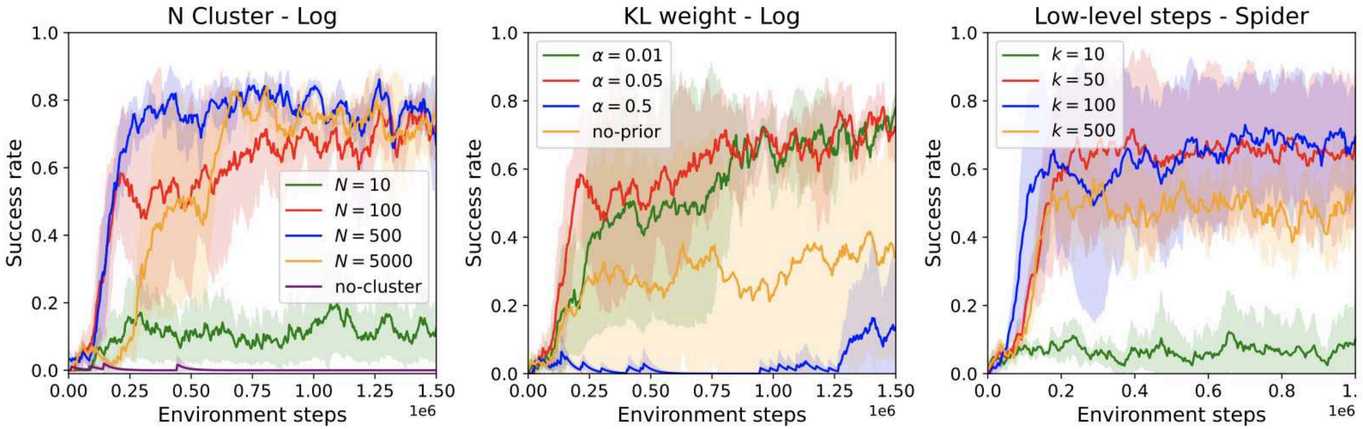


Figure 3: These figures show the curves of task success rates for the methods in the ablation study. The left figure shows PTGM with different numbers of goal clusters and without clustering (**PTGM-no-cluster**) in the Log task. The middle figure shows RL with different weights of the KL reward and RL without the goal prior model (**PTGM-no-prior**) in the Log task. The right figure shows RL with different numbers of low-level steps for each high-level action in the Spider task.

클러스터 갯수	설명
100개 클러스터	모든 작업에서 성공
10개 클러스터	목표 학습 실패
500-5000개 클러스터	성공률 유지, 단 학습 속도 감소
PTGM-no-cluster	목표 공간의 정규화 없이 학습, 실패

KL 계수 (α)	**설명**
$\alpha = 0.01$ 또는 $\alpha = 0.05$	높은 샘플 효율성과 낮은 변동성
$\alpha = 0.5$	성공률 증가가 느림
PTGM-no-prior	높은 학습 변동성, 성능 저하

Low-level Step (k)	**설명**
$k = 10$	샘플 효율성 저하
$k = 500$	높은 샘플 효율성과 강력한 성능
$k = 100$	모든 작업에서 최적 설정

Limitations

Future Works

Reference

[1] Fu, A., Kumar, A., Nachum, O., Tucker, G., & Levine, S. (2020). D4RL: Datasets for Deep Data-Driven Reinforcement Learning. arXiv preprint arXiv:2004.07219.

[2] Gupta, A., Kumar, V., Lynch, C., Levine, S., & Hausman, K. (2020). Relay Policy Learning: Solving Long-Horizon Tasks via Imitation and Reinforcement Learning. Conference on Robot Learning (CORL).

[3] Li, C., Zhang, R., Wong, J., Gokmen, C., Srivastava, S., Martin-Martin, R., ... & Fox, D. (2023). BEHAVIOR-1K: A Benchmark for Embodied AI with 1,000 Everyday Activities and Realistic Simulation. Conference on Robot Learning (CORL).

[4] Johnson, M., Hofmann, K., Hutton, T., & Bignell, D. (2016). The Malmo Platform for Artificial Intelligence Experimentation. International Joint Conference on Artificial Intelligence (IJCAI).

[5] Pertsch, K., Lee, Y., & Lim, J. (2021a). Accelerating Reinforcement Learning with Learned Skill Priors. Conference on Robot Learning (CORL).

[6] Pertsch, K., Lee, Y., Wu, Y., & Lim, J. J. (2021b). Guided Reinforcement Learning with Learned Skills. arXiv preprint arXiv:2107.10253.

[7] Shi, L. X., Lim, J. J., & Lee, Y. (2023). Skill-Based Model-Based Reinforcement Learning. Conference on Robot Learning (CORL).

[8] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous Control with Deep Reinforcement Learning. arXiv preprint arXiv:1509.02971.

[9] Baker, B., Akkaya, I., Zhokov, P., Huizinga, J., Tang, J., Ecoffet, A., ... & Clune, J. (2022). Video Pretraining (VPT): Learning to Act by Watching Unlabeled Online Videos. Advances in Neural Information Processing Systems (NeurIPS).

[10] Lifshitz, S., Paster, K., Chan, H., Ba, J., & McIlraith, S. (2023). Steve-1: A Generative Model for Text-to-Behavior in Minecraft. arXiv preprint arXiv:2306.00937.

[11] Gupta, A., Kumar, V., Lynch, C., Levine, S., & Hausman, K. (2020). Relay Policy Learning: Solving Long-Horizon Tasks via Imitation and Reinforcement Learning. Conference on Robot Learning (CoRL).

[12] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. International Conference on Machine Learning (ICML).

[13] Ziebart, B. D., Maas, A., Bagnell, J. A., & Dey, A. K. (2008). Maximum Entropy Inverse Reinforcement Learning. AAAI Conference on Artificial Intelligence (AAAI).

[14] Pertsch, K., Lee, Y., & Lim, J. (2021). Accelerating Reinforcement Learning with Learned Skill Priors. Conference on Robot Learning (CoRL).

[15] Baker, B., Kanervisto, A., Houghton, B., & Bowen, D. (2022). Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. Advances in Neural Information Processing Systems (NeurIPS).

[16] Fan, L., Xie, Y., Li, S., & Wu, Y. (2022). MineCLIP: Connecting Embodied Control with Internet-Scale Knowledge. Advances in Neural Information Processing Systems (NeurIPS).

[17] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms. arXiv preprint arXiv:1707.06347.

[18] Pertsch, K., Lee, Y., & Lim, J. (2021). Accelerating Reinforcement Learning with Learned Skill Priors. Conference on Robot Learning (CoRL).

[19] Emmons, S., Xie, A., Lynch, C., & Levine, S. (2022). TACO: Trajectory-Conditioned Reinforcement Learning. Advances in Neural Information Processing Systems (NeurIPS).

[20] Baker, B., Kanervisto, A., Houghton, B., & Bowen, D. (2022). Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. Advances in Neural Information Processing Systems (NeurIPS).

Reference

[21] Pertsch, K., Lee, Y., & Lim, J. (2021). Accelerating Reinforcement Learning with Learned Skill Priors. Conference on Robot Learning (CoRL).
[22] Emmons, S., Xie, A., Lynch, C., & Levine, S. (2022). TACO: Trajectory-Conditioned Reinforcement Learning. Advances in Neural Information Processing Systems (NeurIPS).
[23] Baker, B., Kanervisto, A., Houghton, B., & Bowen, D. (2022). Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. Advances in Neural Information Processing Systems (NeurIPS).
[24] Lifshitz, S., Paster, K., Chan, H., Ba, J., & McIlraith, S. (2023). Steve-1: A Generative Model for Text-to-Behavior in Minecraft. arXiv preprint arXiv:2306.00937.

Pre-Training Goal-based Models for Sample-Efficient Reinforcement Learning

Haoqi Yuan , Zhancun Mu , Feiyang Xie , Zongqing Lu
School of Computer Science, Peking University
Yuanpei College, Peking University
Beijing Academy of Artificial Intelligence

Presented as an Oral paper at ICLR 2024 (Accepted, 8/6/8)
https://openreview.net/forum?id=o2IEmeLL9r

2025.02.07
TaeYoon Kwack
njj05043@g.skku.edu

Appendix

Interpretability and Skill Generalization

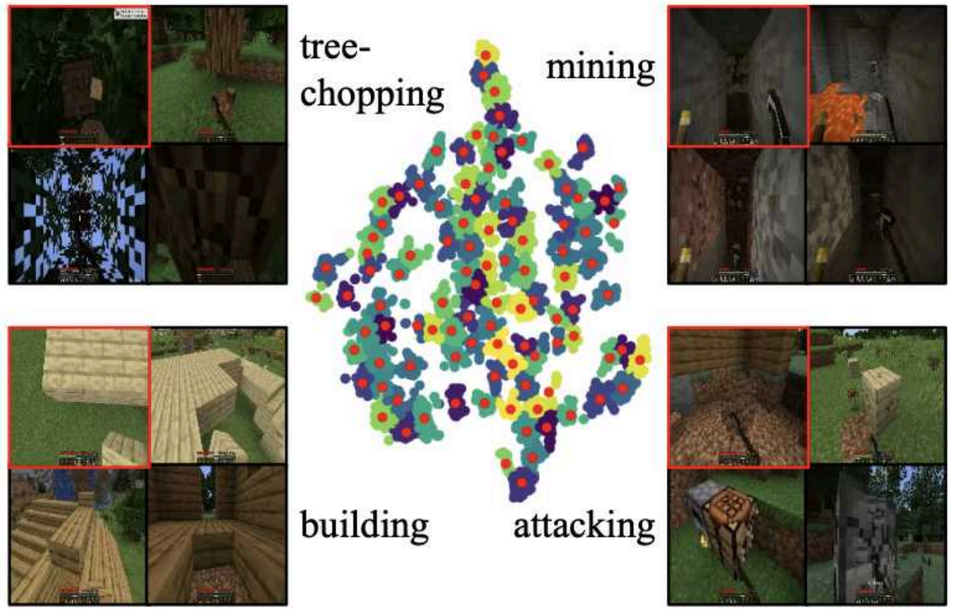


Figure 4: Visualization of goal clustering in Minecraft. Each group of 4 pictures is from the same cluster, where the picture with a red border is the cluster center. In the middle, each red dot means a cluster center.

Test task	Sheep	Pig	Chicken
Success rate	0.82	0.36	0.94
Test task	Place	Water	Wool
Success rate	0.65	0.16	0.44

Table 1: In each row, we pick a goal from the goal clusters and test the goal-conditioned policy in three tasks conditioned on this goal. For the first two rows, in the 16 frames corresponding to the goal, the agent is attacking a sheep. For the last two rows, in the 16 frames, the agent is building a house. In each test task, the target item (mobs or water) is initialized in front of the agent and we test for 100 episodes. The table shows the success rates.

Appendix

Experiment Details

Table 2: Hyperparameters of SAC.

Name	Symbol	Value
Discount factor	γ	0.99
Initial KL reward weight	α	1
KL reward target	δ	0.7
Low-level policy steps	k	20
Rollout buffer size	--	1e6
Training epochs per iteration	--	10
Optimizer	--	AdamW
Batch size	--	256
Learning rate	--	3e-4

Table 4: Hyperparameters of PPO.

Name	Symbol	Value
Discount factor	γ	0.999
KL reward weight	α	0.05
Low-level policy steps	k	100
Rollout buffer size	--	40
Training epochs per iteration	--	5
Optimizer	--	AdamW
Learning rate	--	1e-4
GAE lambda	λ	0.95
Clip range	--	0.2

Table 3: Settings for the five downstream tasks in Minecraft. *Language Description* for each task is used in both computing MineCLIP reward and testing the baseline of Steve-1. *Initial Tools* are provided in the inventory at each episode beginning. *Max Steps* is the maximal episode length.

Task	Language Description	Initial Tools	Max Steps
Harvest log in plains	"Cut a tree."	--	2000
Harvest water bucket in plains	"Find water, obtain water bucket."	bucket	2000
Mine cobblestone	"Obtain cobblestone."	wooden pickaxe	500
Mine iron ore	"Obtain iron ore."	stone pickaxe	2000
Combat spider	"Combat a spider in night plains."	diamond sword	500

Appendix

Experiment Pretraining

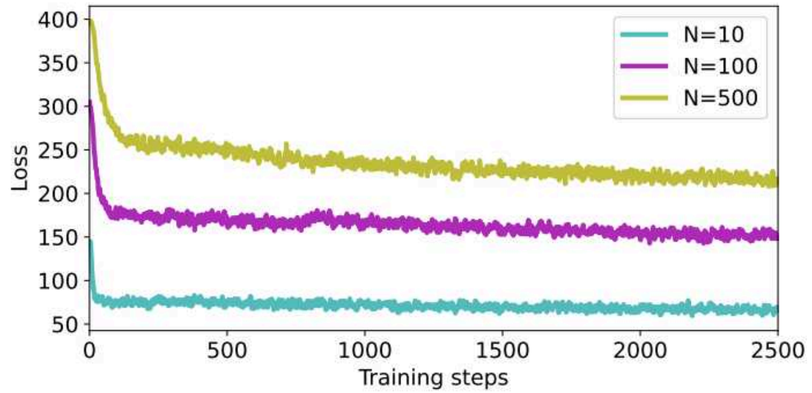


Figure 5: The training loss for pre-training the goal prior model in PTGM on the Minecraft dataset, with different numbers of goal clusters.

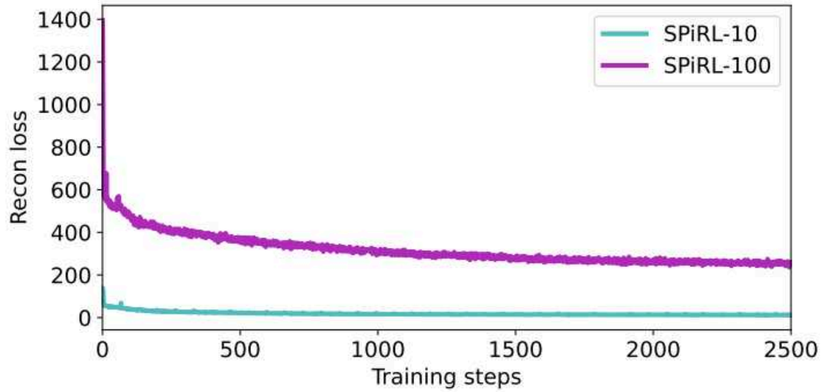
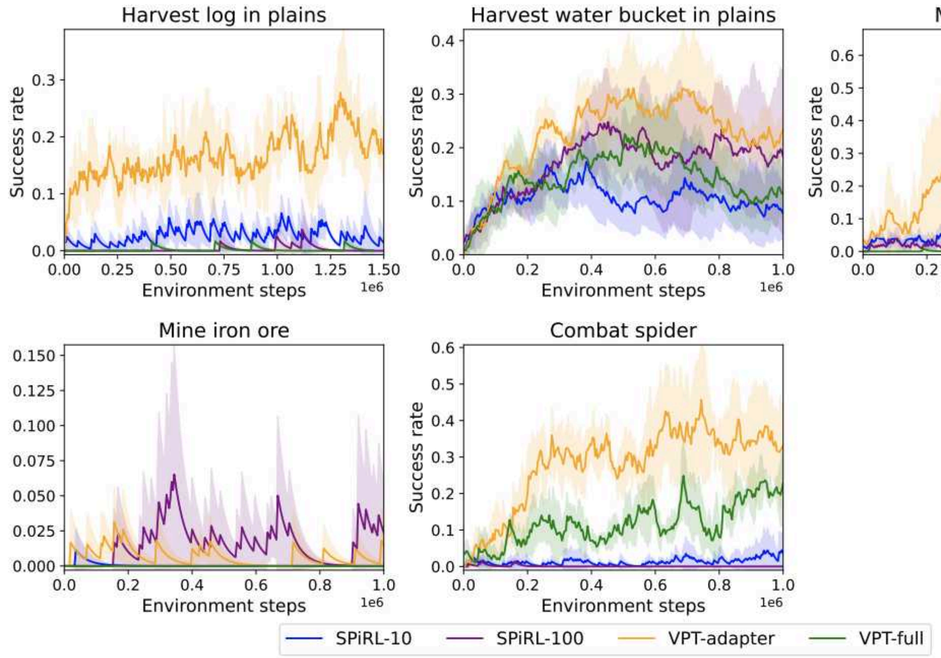


Figure 6: The action reconstruction loss for SPiRL-10 and SPiRL-100 on the Minecraft dataset.

Appendix

Additional Experiments on Baseline

Figure 7 shows results for the baseline methods SPiRL-10, SPiRL-100, VPT-adapter, and VPT-full.



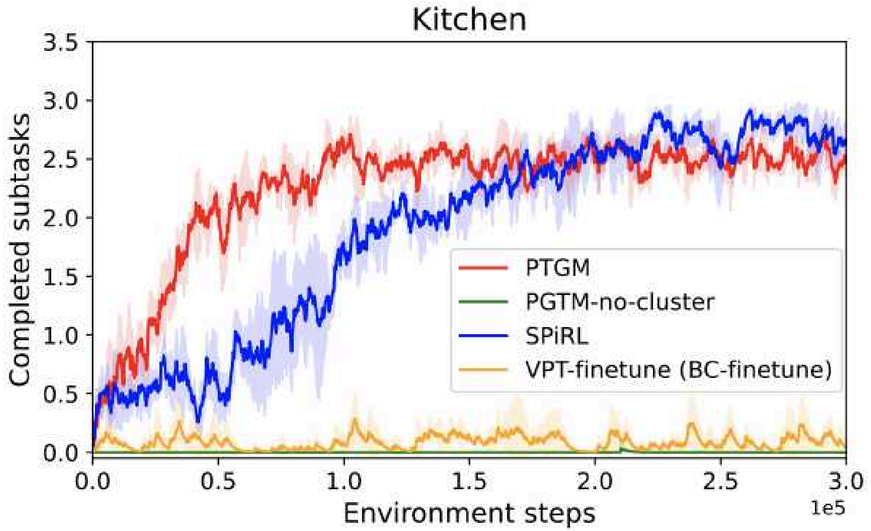


Figure 8: Results for different methods in Kitchen. PTGM-no-cluster means using the continuous goal space without clustering as the action space in the high-level policy.

- SPiRL-10: Low-level 정책에서 한 번의 high-level 액션에 대해 10개의 low-level 액션을 수행
- SPiRL-100: Low-level 정책에서 한 번의 high-level 액션에 대해 100개의 low-level 액션을 수행
- VPT-adapter: Transformer 기반 VPT 모델의 Adapter만 미세 조정
- VPT-full: 전체 VPT 모델을 미세 조정

VPT-adapter와 VPT-full이 대부분의 작업에서 우수한 성능을 보임 / SPiRL-10과 SPiRL-100은 장기 작업에서 취약

Appendix

Additional Ablation Study

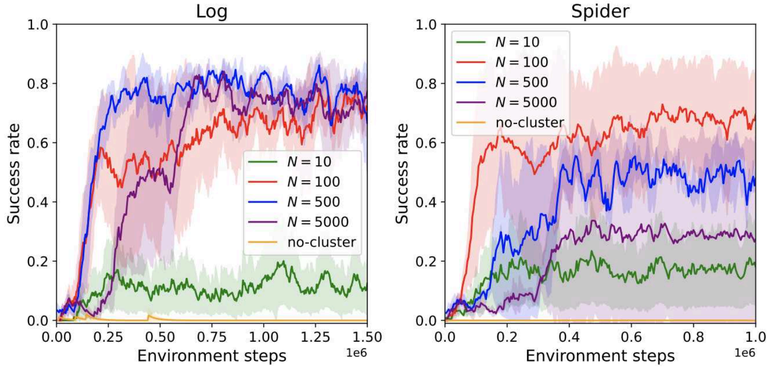


Figure 9: Results for the ablation study of PTGM with different numbers of goal clusters and without clustering (**PTGM-no-cluster**) in Minecraft tasks.

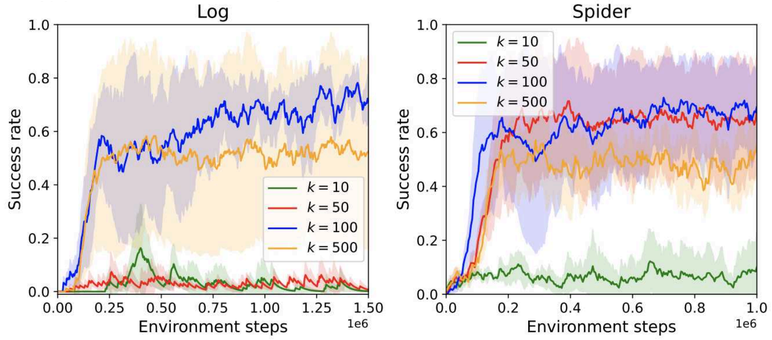


Figure 10: Results for the ablation study of PTGM with different numbers of low-level steps for each high-level action in Minecraft tasks.

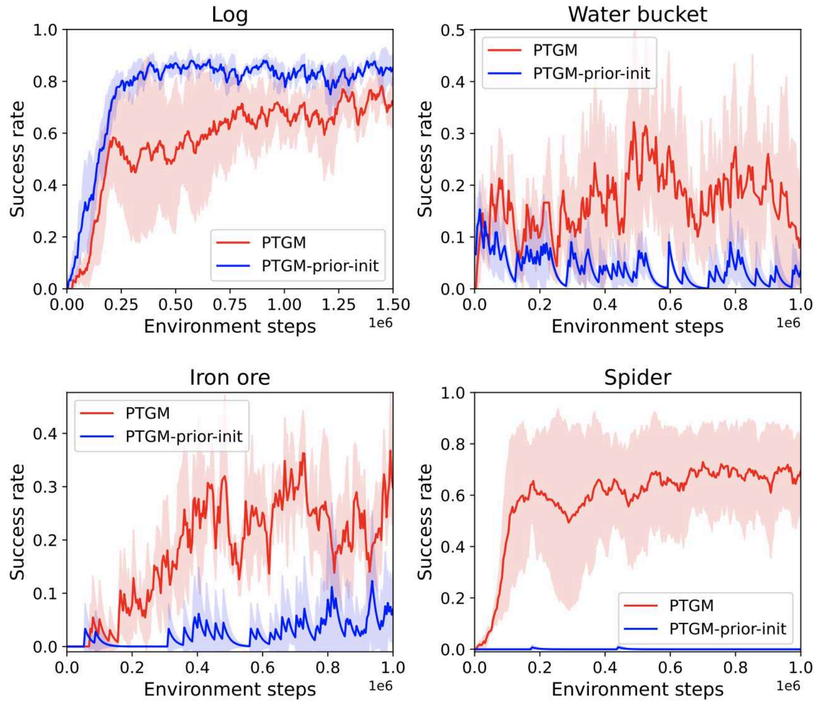


Figure 11: Ablation study for PTGM that initializes the high-level policy with the pre-trained goal prior for RL (**PTGM-prior-init**) in Minecraft tasks.

- **Goal Clustering:** N = 100 또는 N = 500이 최적
- **Low-level Steps:** k = 100이 가장 효과적
- **Goal Prior Initialization:** PTGM이 전반적으로 강력한 성능을 보였으며, PTGM-prior-init은 특정 작업(예: Log task)에 유리

Appendix

Goal Cluster Visualization