PRESELECTION-FREE KNOWLEDGE-GROUNDED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing agents solve sequential decision-making tasks by accessing knowledge, such as expert-designed data, or navigating the environments through trial and error. Research has explored knowledge as supervision or reinforcement learning for interactive improvement, but typically studied independently. Recent work introduces a unified paradigm called Knowledge-Grounded Reinforcement Learning (KGRL), which studies how RL can efficiently leverage external knowledge and how integrating these two complementary signals can enhance learning. However, prior KGRL methods require an expert-preselected knowledge set for each task, which is hard to obtain in real-world applications. Without it, performance can degrade significantly due to misleading information. We propose a **trade-and-curate** mechanism for KGRL that *trades* stochastic information among knowledge policies and *curates* knowledge-policy fusion, making the agent resilient to unhelpful information in an unselected knowledge set. Our experiments in KGRL MiniGrid environments show that the trade-and-curate mechanism achieves up to 89.6% performance improvement without knowledge preselection.

1 INTRODUCTION

Learning through interactions with environments, such as reinforcement learning (RL), is pivotal for training agents when supervision data is unavailable. For example, robotics agents interact with physical environments and gather feedback from diverse sensors to support motion generation under constraints and hard-to-model dynamics (Morimoto & Doya, 2001; Riedmiller et al., 2009; Navarro-Guerrero et al., 2012; Qureshi et al., 2018; Ibarz et al., 2021; Luo et al., 2025). Digital conversational agents need continual human input to stay relevant to modern language, news, human preferences, and safety regulations (Gašić et al., 2011; Li et al., 2016; Ouyang et al., 2022; Perez et al., 2022). These applications highlight the value of interactive learning in helping agents navigate complex, dynamic environments, even though RL usually requires substantial interaction data to solve a task.

In parallel to RL, another line of research studies how knowledge-grounded learning can leverage available information to achieve desirable performance. For example, imitation learning uses expert-collected trajectories to provide agents with supervised signals for learning various tasks (Abbeel & Ng, 2004; Bagnell et al., 2006; Ross et al., 2011; Ho & Ermon, 2016; Wu et al., 2019; Wang et al., 2023; Hu et al., 2024). Research also studies how to leverage knowledge, including that embedded in pretrained models, to enhance an agent's reasoning ability (Ghazvininejad et al., 2018; Tuan et al., 2019; Petroni et al., 2019; Tuan et al., 2020; Lewis et al., 2020; Wei et al., 2022; Edge et al., 2024). However, knowledge-grounded learning requires sufficient supervision data, which can be hard to collect in fields such as embodied or privacy-sensitive applications. Combining interactive and knowledge-grounded learning therefore becomes important to achieve the best of both worlds.

The integration of interactive and knowledge-grounded learning has recently been unified under the Knowledge-Grounded Reinforcement Learning (KGRL) paradigm (Chiu et al., 2023). KGRL focuses on how RL agents can use knowledge, regardless of its optimality (optimal, sub-optimal, or irrelevant), sources (from experts or crowd-sourcing), or types (skills, guidance, or regulations), as learning signals in addition to environmental rewards during interactive learning. Knowledge in KGRL takes the form of state-action mappings, which are actionable for decision-making. Prior KGRL methods (Rajendran et al., 2017; Zhang et al., 2020; Chiu et al., 2023; Ho et al., 2025) have shown that they require fewer interactive samples by leveraging selected knowledge. Nonetheless,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

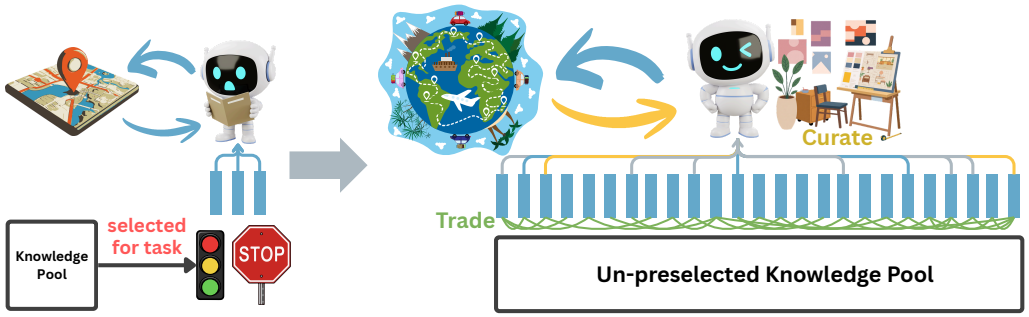


Figure 1: (Left) Prior KGRL methods integrate expert preselected knowledge to improve RL sample efficiency for each individual environment. (Right) To obviate precautionary knowledge selection, we introduce the trade-and-curate mechanism, enabling knowledge policies to *trade* their information and the agent to *curate* useful knowledge before final deployment.

they assume experts have selected knowledge for each environment, whereas a real-world knowledge set, without precautionary selection, can be large, noisy, duplicated, and contain irrelevant items. Moreover, prior methods generate actions using every piece of given knowledge at each training step, but retain only a small set of information and discard others deemed less useful. This indicates significant information loss, and how to prevent data waste needs to be addressed.

We propose a trade mechanism and curated policy fusion for embedding-based KGRL frameworks to leverage an unselected knowledge set and to prevent data waste during training (Figure 1). The trade mechanism enhances the knowledge representation space by exchanging information among knowledge policies through optimizing a trade loss. This mechanism supports bottom-up navigation of a large, noisy knowledge set since, through information exchange, the knowledge policies learn their semantic relationships and self-organize into structured representations. All data contributes to this representation learning; hence, no data is wasted. Then, given an input state, the curated policy fusion uses these representations to identify a small subset of the most relevant knowledge policies and combine them into a policy distribution. Our trade-and-curate mechanism can leverage a knowledge set more effectively when no preselection is performed.

We integrate our trade-and-curate mechanism into the state-of-the-art embedding-based KGRL method, Knowledge-Inclusive Attention Network (KIAN) (Chiu et al., 2023), and introduce TC-KIAN. We evaluate TC-KIAN and prior KGRL methods on KGRL MiniGrid environments. The results indicate two key findings: First, with the same unselected knowledge set, our method leverages it more effectively and improves performance by up to 89.6% compared to not adding our trade-and-curate mechanism. Second, compared to KIAN with an expert-selected knowledge set, which is presumed to be our performance upper bound, our method significantly reduces the performance gap from a 25.4% drop to nearly 0%.

2 Related Work

KGRL was formally defined in Chiu et al. (2023), studying how RL agents can effectively leverage a set of external knowledge policies. This concept can be traced back to RL from demonstrations (RLfD) (Vecerik et al., 2017; Hester et al., 2018; Rajeswaran et al., 2018; Nair et al., 2018; Pfeiffer et al., 2018; Goecks et al., 2019), which combines a reward-driven objective with a behavior-cloning objective that aims to imitate behavior suggested by knowledge represented as expert-recorded trajectories. Other work expands the forms of knowledge from trajectories in RLfD to state-action mappings (Rajendran et al., 2017; Zhang et al., 2020). While these methods incorporate more general forms of knowledge other than the demonstration data, they assign a fixed index to each knowledge policy and learn to select knowledge by index. Chiu et al. (2023) proposed the Knowledge-Inclusive Attention Network (KIAN) to loosen the restriction of index-based attention by introducing knowledge keys (representations) and a state-dependent query for knowledge selection, and proved theoretically and empirically the potential of KIAN to achieve sample-efficient,

compositional, and incremental learning in RL. Recently, KIAN-ACE (Ho et al., 2025) adds an auxiliary loss to KIAN's objective to maximize the utility of preselected knowledge policies during initial training. Although KIAN-ACE can supposedly explore all knowledge policies early before deciding their usefulness, its design motive contradicts our goal of being preselection-free. In addition, unlike KIAN and KIAN-ACE, which adopt top-down knowledge selection driven only by the state-dependent query, our trade-and-curate mechanism lets knowledge policies exchange information, making it more effective for navigating a noisier, larger knowledge set.

3 BACKGROUND

In this section, we first define the KGRL problem, which studies how to leverage an available knowledge set to facilitate maximizing the expected return. We then describe KIAN's model architecture and knowledge-policy fusion, which is the state-of-the-art model that we build upon.

3.1 KNOWLEDGE-GROUNDED REINFORCEMENT LEARNING (KGRL)

The KGRL paradigm (Chiu et al., 2023) describes a sequential decision-making problem with accessible knowledge as a Knowledge-Grounded Markov Decision Process (KGMDP), represented by the tuple $\mathcal{M}_K = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \rho, \gamma, \mathcal{G})$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition probability distribution $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, reward function R, initial state distribution ρ, discount factor γ, and an external knowledge set $\mathcal{G}$ available for the problem. The goal of KGRL is to find an optimal policy $\pi^*(\cdot|\cdot; \mathcal{G}) : \mathcal{S} \to \mathcal{A}$ that maximizes the expected return: $\mathbb{E}_{\mathbf{s}_0 \sim \rho, \mathcal{T}, \pi^*}[\sum_{t=0}^{T} \gamma^t R_t]$.

The external knowledge set $\mathcal{G}$ contains n knowledge policies, $\mathcal{G} = \{\pi_{g_1}, \ldots, \pi_{g_n}\}$. Each π_{g_j} is a state-action mapping, $\pi_{g_j}(\cdot|\cdot) : \mathcal{S} \to \mathcal{A}, \forall\ j = 1, \ldots, n$, and can be in any form, such as handcrafted rules, programs, or neural networks. A knowledge policy π_{g_j} is not necessarily designed for the problem. Therefore, directly applying π_{g_j} can lead to suboptimal behavior or a complete failure.

While a KGMDP $\mathcal{M}_K$ and a regular MDP $\mathcal{M}$ share the same optimal policies if they have the same policy class $\Pi : \mathcal{S} \to \mathcal{A}$, KGRL agents are expected to simultaneously achieve knowledge-acquirable, generalizable, compositional, incremental, and sample-efficient properties described as necessary for a general agent in Kaelbling (2020). In this work, we focus on the sample-efficient property, i.e., an agent requires fewer samples to solve for $\mathcal{M}_k$ than for $\mathcal{M}$ when the external knowledge set $\mathcal{G}$ provides mildly useful knowledge toward the goal.

3.2 KNOWLEDGE-INCLUSIVE ATTENTION NETWORK (KIAN)

KIAN (Chiu et al., 2023; Ho et al., 2025) incorporates external knowledge through a knowledge-inclusive policy architecture and a policy-fusion mechanism.

KIAN's policy architecture includes three learnable components: (1) an inner actor $\pi_{in}(\cdot|\cdot; \boldsymbol{\theta}) : \mathcal{S} \to \mathcal{A}$, which is a function approximator that enables agents to develop skills not covered by the external knowledge, (2) state-independent knowledge keys $\mathbf{k}_{in} \in \mathbb{R}^{d_k}$ and $\mathbf{k}_{g_j} \in \mathbb{R}^{d_k}, \forall j \in \{1, \ldots, n\}$, which are d_k-dimensional embeddings representing the knowledge policies, and (3) a state-dependent query $\Phi(\cdot; \boldsymbol{\phi}) : \mathcal{S} \to \mathbb{R}^{d_k}$, which is a function approximator that outputs a d_k-dimensional vector given a state, $\mathbf{u}_t = \Phi(\mathbf{s}_t; \boldsymbol{\phi})$. This vector $\mathbf{u}_t$ will attend to each knowledge key for knowledge selection at each time step.

KIAN outputs an action by its policy-fusion mechanism that fuses the inner actor and external knowledge policies: $\pi = \hat{w}_{in}\pi_{in} + \sum_{j=1}^{n} \hat{w}_{g_j}\pi_{g_j}$, where $\hat{w}_{in} + \sum_{j=1}^{n} \hat{w}_{g_j} = 1$. The weights for all knowledge policies are calculated by

$$[\hat{w}_{in}, \hat{w}_{g_1}, ..., \hat{w}_{g_n}] = \texttt{softmax}([\hat{\mathbf{u}}_t \cdot \hat{\mathbf{k}}_{in}, \hat{\mathbf{u}}_t \cdot \hat{\mathbf{k}}_{g_1}, \ldots, \hat{\mathbf{u}}_t \cdot \hat{\mathbf{k}}_{g_n}]), \tag{1}$$

where $\hat{\mathbf{u}}_t = \mathbf{u}_t / \|\mathbf{u}_t\|$, $\hat{\mathbf{k}}_{in} = \mathbf{k}_{in} / \|\mathbf{k}_{in}\|$, and $\hat{\mathbf{k}}_{g_j} = \mathbf{k}_{g_j} / \|\mathbf{k}_{g_j}\|$. Specifically, for discrete action spaces, the knowledge policies π_{in} and π_{g_j} can be the softmax of unnormalized d_a-dimensional vectors, $\mathbf{z}_{in} \in \mathbb{R}^{d_a}$ and $\mathbf{z}_{g_j} \in \mathbb{R}^{d_a}$, where $d_a \in \mathbb{R}$ is the number of possible actions. Therefore, KIAN's fused policy becomes

$$\pi(\cdot|\mathbf{s}_t) = \hat{w}_{in}\texttt{softmax}(\mathbf{z}_{in}) + \sum_{j=1}^{n} \hat{w}_{g_j}\texttt{softmax}(\|\mathbf{u}_t\|\|\mathbf{k}_{g_j}\|\mathbf{z}_{g_j}). \tag{2}$$

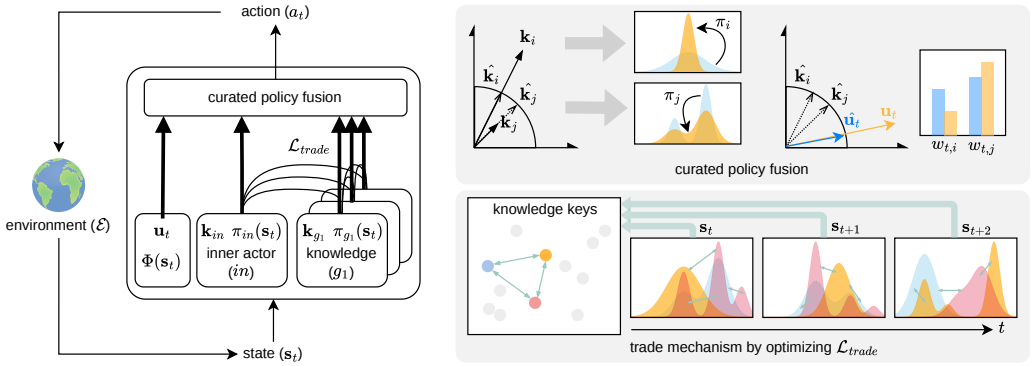


Figure 2: Illustration of TC-KIAN. (Left) The policy architecture with (1) the trade mechanism that optimizes $\mathcal{L}_{trade}$ and (2) curated policy fusion. (Right Bottom) The trade mechanism learns knowledge keys through information trading among policies across states and time. (Right Top) Curated policy fusion uses $\|\mathbf{k}_i\|$ to adjust sharpness of π_i and $\|\mathbf{u}_t\|$ to adjust weight distribution w_t.

Note that the term $\|\mathbf{u}_t\|\|\mathbf{k}_{g_j}\|$ in Eq. 2 acts as the inverse temperature for each knowledge policy π_{g_j} to mitigate the entropy imbalance issue (Chiu et al., 2023), which causes a biased knowledge selection due to a large entropy difference between the inner and external knowledge policies. All learnable components are end-to-end updates with an RL algorithm.

While KIAN is shown to be effective for training KGRL agents, its performance degrades when the external knowledge set $\mathcal{G}$ is not optimally preselected. Our method addresses this issue.

4 Method

Navigating an un-preselected knowledge set is a major challenge in KGRL that can lead to scalability issues. To address this challenge, we propose the Information **T**rading and Policy **C**uration KIAN (TC-KIAN). The method introduces the **Trade Mechanism** and **Curated Policy Fusion** to KIAN (Chiu et al., 2023) and is detailed in Algorithm 1 (Appendix B).

Conceptually, instead of relying on a single orchestrator to decide how to fuse knowledge, we enable the knowledge policies to collaborate bottom-up. Each knowledge, $(\mathbf{k}_{in}, \pi_{in})$ or $(\mathbf{k}_{g_j}, \pi_{g_j})$, acts as a decision maker that *trades* information to understand each other's capabilities, similar to how group members in a transactive memory system (TMS) (Wegner et al., 1985) collaboratively encode knowledge. Upon receiving a query from the orchestrator, the decision-makers can quickly propose a set of them that are best suited to address the request. The orchestrator then *curates* the proposed knowledge into the final policy.

4.1 Trade Mechanism: Information Trading across Knowledge Policies

We propose the trade mechanism that guides knowledge policies to exchange functionality information and store it in their representation space. Specifically, we align the relationships among the knowledge representations, $\mathbf{k}_{in}$ and $\mathbf{k}_{g_j}$, with the relationships among the knowledge policy behaviors, π_{in} and π_{g_j}, by minimizing the following *trade loss*:

$$\mathcal{L}_{trade} = \mathbb{E}_{\mathbf{s}'\sim\mathcal{T}^{\pi}(\cdot|\mathbf{s})} \sum_{i\in\mathcal{I},j\in\mathcal{I}} S^{ij}_{\mathbf{k}} \cdot \mathtt{sg}\left(D^{ij}_{\pi}(\mathbf{s}') - th\right), \tag{3}$$

where $\mathcal{T}^{\pi}(\cdot|\mathbf{s})$ is the state-transition probability under the agent's policy π, $\mathcal{I} := \{in, g_1, \ldots, g_n\}$, $S^{ij}_{\mathbf{k}} \in \mathbb{R}$ is the similarity between two knowledge keys $\mathbf{k}_i$ and $\mathbf{k}_j$, $D^{ij}_{\pi}(\mathbf{s}) \in \mathbb{R}$ is the distance between two knowledge policies given the same state, $\pi_i(\cdot|\mathbf{s})$ and $\pi_j(\cdot|\mathbf{s})$, and $th \in \mathbb{R}$ is the threshold. As the main purpose of $\mathcal{L}_{trade}$ is to update the geometric relationships among knowledge keys, we apply the stop-gradient function $\mathtt{sg}(\cdot)$ to other learnable components when optimizing $\mathcal{L}_{trade}$.

Since the knowledge representation space is a subspace of the d_k-dimensional vector space, we use the negative Euclidean distance for $S^{ij}_{\mathbf{k}}$ to measure the geometric similarity between two knowledge

keys. With each knowledge policy being a probability distribution, we use the Kullback-Leibler (KL) Divergence (Jeffreys, 1998) for $D_\pi^{ij}(\mathbf{s})$ to estimate the directional distance of $\pi_j(\cdot|\mathbf{s})$ from $\pi_i(\cdot|\mathbf{s})$, which are both action-space probability distributions of knowledge policies given state $\mathbf{s}$:

$$S_{\mathbf{k}}^{ij} = -\|\mathbf{k}_i - \mathbf{k}_j\|_2 \tag{4}$$

$$D_\pi^{ij}(\mathbf{s}) = D_{KL}(\pi_i(\cdot|\mathbf{s}) \parallel \pi_j(\cdot|\mathbf{s})). \tag{5}$$

Note that in Eq. 3, the knowledge pairs (i, j) and (j, i) are distinct terms, so in effect the bi-directional KL Divergence is computed between two knowledge policies π_i and π_j.

The threshold th in Eq. 3 provides a pivot point for when $S_{\mathbf{k}}^{ij}$ should be maximized or minimized. When $D_\pi^{ij} \geq th$, minimizing $\mathcal{L}_{trade}$ equals to minimizing $S_{\mathbf{k}}^{ij}$. This implies that when the distance between two policies is high enough, their keys in the knowledge representation space are pushed away. On the other hand, when $D_\pi^{ij} \leq th$, minimizing $\mathcal{L}_{trade}$ equals to maximizing $S_{\mathbf{k}}^{ij}$. This implies that when the distance between two policies is low enough, their keys in the knowledge representation space are pulled together.

The threshold in $\mathcal{L}_{trade}$ is critical for stabilizing the training. Without it, the knowledge keys would always be pushed away from each other, except for completely identical policies, causing the knowledge key space to explode and fail to encode relative relationship information. While in theory, an ideal threshold should depend on $D_\pi^{ij}(\mathbf{s})$ estimated from all possible states, it is impractical to compute the threshold with infinitely many states and a continually changing π_{in}. Therefore, we set the threshold th to be the average of D_π^{ij} over a sampled batch of states at each training iteration.

Our method optimizes $\mathcal{L}_{trade}$ in addition to RL's reward-driven objective, providing explicit learning signals for knowledge keys. This differs from prior work, where knowledge keys are implicitly, end-to-end updated based on rewards obtained by the fused policy in Eq. 2. In addition, Eq. 2 requires $n+1$ forward passes from all knowledge policies at each time step, but KIAN's policy optimization uses only a few outputs, wasting a significant amount of data during training, especially when n is large. Our trade mechanism uses all the outputs from knowledge policies to update the knowledge key relationships, so the outputs not selected for policy optimization still contribute to learning.

We mathematically show in Appendix A.1 (Property 1) that if a unique set of solutions for $\arg\min_{\mathbf{k}_{in},\mathbf{k}_{g_1},\ldots,\mathbf{k}_{g_n}} \mathcal{L}_{trade}$ exists, then minimizing $\mathcal{L}_{trade}$ will follow the desired pattern: the keys of knowledge policies with similar functionality will be pulled together, and those with different functionality will be pushed away. We also show in Appendix A.2 (Theorem 1) that minimizing $\mathcal{L}_{trade}$ with a stochastic gradient-based optimizer under accessible assumptions will converge.

4.2 Curated Policy Fusion

The knowledge keys $\mathbf{k}_{in}, \mathbf{k}_{g_1}, \ldots, \mathbf{k}_{g_n}$ directly affect knowledge fusion. KIAN uses Eq. 1 to compute the weights of selecting knowledge policies. However, Eq. 1 only accounts for the query and each key's directions and not their magnitudes, as $\hat{\mathbf{u}}_t \cdot \hat{\mathbf{k}}_i = \mathbf{u}_t \cdot \mathbf{k}_i / (\|\mathbf{u}_t\| \|\mathbf{k}_i\|)$. In other words, the effective input space of `softmax` in Eq. 1 collapses from the entire $\mathbb{R}^{d_k} \times \mathbb{R}^{d_k}$ space to the Cartesian product of two unit spheres in $\mathbb{R}^{d_k}$. This constrained space makes it difficult to draw fine-grained distinctions between two knowledge keys with subtle differences in their corresponding knowledge policies, especially when n is large. Optimizing $\mathcal{L}_{trade}$ can worsen the situation, as it deliberately separates keys of distinct knowledge policies, leading to an even smaller feasible input space for `softmax` in Eq. 1. Specifically, $\mathcal{L}_{trade}$ creates a discrete-like behavior in the continuous representation space (Tuan et al., 2022), making fewer available spots for knowledge keys.

We address the restricted input space of `softmax` in Eq. 1 while keeping KIAN's inverse-temperature design for mitigating the entropy imbalance issue (Chiu et al., 2023). We propose the *curated policy fusion*:

$$[\hat{w}_{t,in}, \hat{w}_{t,g_1}, \ldots, \hat{w}_{t,g_n}] = \texttt{softmax}\left(\left[\mathbf{u}_t \cdot \hat{\mathbf{k}}_{in}, \mathbf{u}_t \cdot \hat{\mathbf{k}}_{g_1}, \ldots, \mathbf{u}_t \cdot \hat{\mathbf{k}}_{g_n}\right]\right) \tag{6}$$

$$\pi(\cdot|\mathbf{s}_t) = \hat{w}_{t,in}\texttt{softmax}(\mathbf{z}_{in}) + \sum_{j=1}^{n} \hat{w}_{t,g_j}\texttt{softmax}\left(\|\mathbf{k}_{g_j}\|\mathbf{z}_{g_j}\right). \tag{7}$$

Note that $\mathbf{u}_t = \|\mathbf{u}_t\|\hat{\mathbf{u}}_t$, so the norm $\|\mathbf{u}_t\|$ acts as the inverse temperature for $\texttt{softmax}([\hat{\mathbf{u}}_t \cdot \hat{\mathbf{k}}_{in}, \hat{\mathbf{u}}_t \cdot \hat{\mathbf{k}}_{g_1}, ..., \hat{\mathbf{u}}_t \cdot \hat{\mathbf{k}}_{g_n}])$. The norm $\|\mathbf{k}_{g_j}\|$ acts as the inverse temperature for $\texttt{softmax}(\mathbf{z}_{g_j})$. Compared

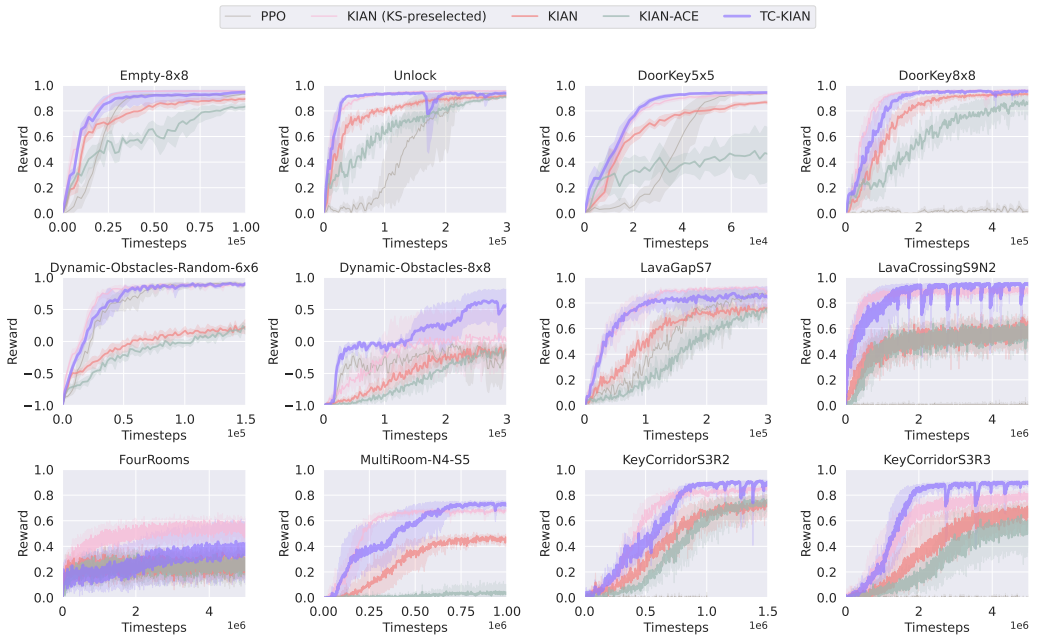


Figure 3: Learning curves of TC-KIAN, KIAN-ACE, and KIAN under the un-preselected knowledge setting. The KIAN (KS-preselected) is a presumed upper bound, and PPO is a reference for learning without knowledge. TC-KIAN shows results closest to, or better than, the upper bound.

to Eq. 1 and 2, this new policy fusion expands the effective input space of `softmax` for weight calculation while retaining the ability to adjust the entropy of knowledge policies. Figure 2 (right) illustrates the effects of the inverse temperatures $\|\mathbf{k}_{g_j}\|$ and $\|\mathbf{u}_t\|$. $\|\mathbf{k}_{g_j}\|$ controls the decisiveness of the knowledge policy π_{g_j}, with $\|\mathbf{k}_{g_j}\| > 1$ leading to a more deterministic policy and vice versa. $\|\mathbf{u}_t\|$ controls the decisiveness of the weight distribution, with $\|\mathbf{u}_t\| > 1$ making the weights concentrate on fewer knowledge policies and vice versa. Our trade mechanism separates knowledge keys by policy functionality, and the curated policy fusion uses $\|\mathbf{u}_t\|$ to focus efficiently on the knowledge policies most useful to the task.

The trade loss $\mathcal{L}_{trade}$ and curated policy fusion should be applied in combination for effective learning. Without curated policy fusion, the trade loss can hurt performance by restricting the effective input space for weight calculation. Without $\mathcal{L}_{trade}$, the sharper weight distribution from curated policy fusion can cause gradient vanishing for a knowledge key if its weight is set to zero. Poorly learned knowledge keys can cause high variance in training, as a query vector $\mathbf{u}_t$ may attend to a mix of well-suited and inappropriate keys simply because they happen to be nearby. As a result, selected knowledge policies can occasionally behave inconsistently, leading to high performance variance.

We mathematically show in Appendix A.3 (Property 2) that, under curated policy fusion, the linear-combination relationships of knowledge policies can be transferred to their knowledge keys. This property implies that functional knowledge relationships are maintained in the representation space, enabling structural understanding of the fused policy and knowledge reuse across tasks.

5 Experiments and Results

Following prior work, we evaluate TC-KIAN on MiniGrid (Chevalier-Boisvert et al., 2023) under prior MiniGrid knowledge set (Chiu et al., 2023) across 12 environments, including [Easy] Empty-8x8, Unlock, DoorKey-5x5, DoorKey-8x8, [Safety/Regulation] Dynamic-Obstacles-Random-6x6, Dynamic-Obstacles-8x8, LavaGap7, LavaCrossingS9N2, [Exploration] FourRooms, MultiRoom-N4-S5, KeyCorridorS3R2, KeyCorridorS3R3, covering different goals and challenges (Figure 9).

Unlike prior work that experiments with a human-preselected knowledge set for each environment (named KS-preselected), we use an **un-preselected knowledge set** (named KS-all) for all environ-

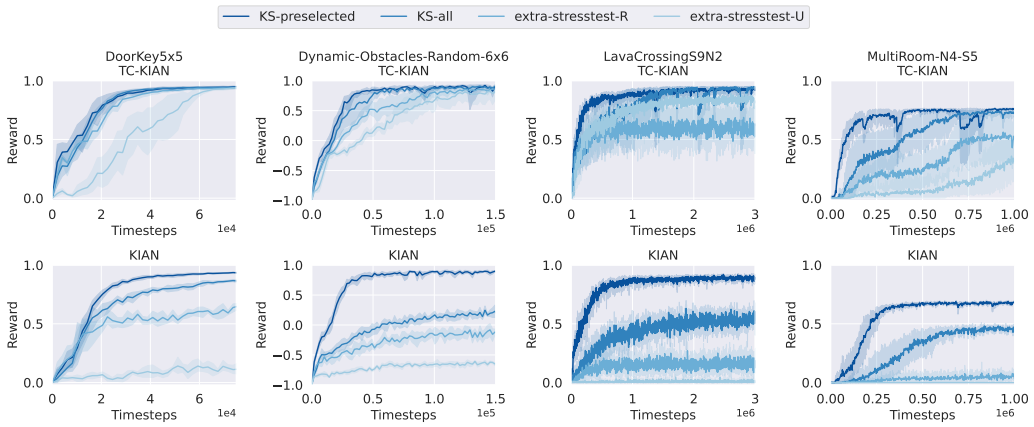


Figure 4: Stress test of KIAN and TC-KIAN by introducing extra knowledge to KS-all. The learning curves show that KIAN's results drop significantly under duplicative and noisy knowledge. TC-KIAN is more resilient when provided with a non-ideal knowledge set across all environments.

ments. All knowledge policies are suboptimal or irrelevant for each environment, so none of them alone or their simple ensembles can solve the tasks. This setting is more realistic because, in real-world applications, we often cannot know beforehand which knowledge policies might be helpful.

We compare TC-KIAN with two state-of-the-art KGRL algorithms, KIAN (Chiu et al., 2023) and KIAN-ACE Ho et al. (2025). In addition, we use KIAN with KS-preselected as a performance upper bound and study how TC-KIAN with KS-all closes the performance gap to the upper bound. All methods are updated with proximal policy optimization (PPO) (Schulman et al., 2017). More experimental details are in Appendixes C and D.

5.1 Performance Gap between Using KS-Preselected and KS-All

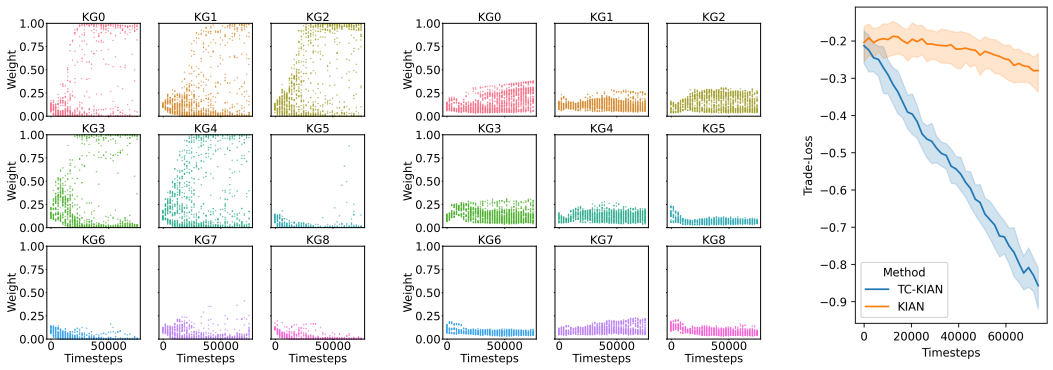
Figure 3 shows the reward learning curves across 12 environments. PPO results provide a reference for learning without access to knowledge, and KIAN (KS-preselected) results serve as a presumed performance upper bound since their knowledge sets are preselected by an expert. We observe that when using KS-all, KIAN's performance drops significantly, converging more slowly and to a worse average reward. KIAN-ACE guides agents to try every knowledge policy during training, so when the knowledge set contains noisy and irrelevant policies, this exploration can impair performance.

TC-KIAN consistently outperforms the baselines in bridging the performance gap between using KS-preselected and KS-all. In environments that are more difficult to explore, such as Dynamic-Obstacles-Random-6x6, LavaCrossingS9N2, MultiRoom-N4-S5, KeyCorridorS3R2, and KeyCorridorS3R3, TC-KIAN even achieves a higher converged average reward than KIAN (KS-preselected), the presumed upper bound. Specifically, in challenging environments, the improvement of TC-KIAN over KIAN (KS-all) is more pronounced: Dynamic-Obstacles-Random-6x6 (58.8%), Dynamic-Obstacles-8x8 (89.6%), LavaCrossingS9N2 (66.1%), FourRooms (59.0%), MultiRoom-N4-S5 (63.2%), KeyCorridorS3R3 (32.1%). These results suggest that our trade-and-curate mechanism is more effective in choosing useful knowledge policies from a noisy set.

5.2 Stress Test beyond Preselection-Free

In light of the real-world challenge that a knowledge set can contain duplicates and random noise, we define two knowledge sets to stress test different methods: **Extra-stresstest-R** expands KS-all to include 100 knowledge policies, each randomly sampled from KS-all, and **Extra-stresstest-U** expands KS-all to include 100 knowledge policies, with the additional policies being uniform. We evaluate KIAN and TC-KIAN using these two knowledge sets on four MiniGrid environments, DoorKey-5x5, Dynamic-Obstacles-Random-6x6, LavaCrossingS9N2, and MultiRoom-N4-S5, to test whether our method makes KGRL agents more resilient to duplicative and noisy information.

The results in Figure 4 demonstrate TC-KIAN's superiority in knowledge scalability and robustness to knowledge redundancy and noisiness. The final average reward of TC-KIAN does not drop



(a) TC-KIAN (b) KIAN (c)

Figure 5: (a,b) Knowledge weight distributions during training. (c) Trade loss curves. TC-KIAN identifies useful knowledge more decisively and explicitly optimizes an implicit objective in KIAN.

compared to the presumed upper bound (KS-preselected) in DoorKey5x5 and Dynamic-Obstacles-Random-6x6, and it takes only at most twice as many training steps to converge. In LavaCrossingS9N2 and MultiRoom-N4-S5, the final average reward drops by around 25%-58%, as expected, because the knowledge sets are far from ideal. In contrast, KIAN's final average reward drops by 33%-92% across all environments when using Extra-stesstest-R and 81%-100% when using Extra-stesstest-U. These results show that our trade-and-curate mechanism makes KGRL training robust to the challenges posed by large, redundant, and noisy knowledge sets.

5.3 Behavioral Analysis

Policy Fusion Weight Distribution. To empirically verify the sources of TC-KIAN's improvements, we study the chronological dynamics of knowledge weights during training, which exhibit different patterns between KIAN and TC-KIAN. Both methods use the same knowledge set, including 1 inner actor (KG0) and 8 external knowledge policies (KG1-8). Figures 5a and 5b plot the weights used in policy fusion along training iterations. KIAN's results indicate a lukewarm policy fusion that uniformly uses every knowledge available. In contrast, TC-KIAN is more decisive in selecting knowledge, assigning irrelevant knowledge policies (KG5-8) only 1/5-1/60 of KIAN's assigned weights. These results validate that our curated policy fusion assigns sharper weights to knowledge policies, leading to greater resilience to non-helpful knowledge.

Progression of the Learned Knowledge Representation Space. We study how the keys change in the knowledge representation space across training iterations. Figure 6's heatmaps show the progression of knowledge-key similarities for TC-KIAN trained with the Extra-stresstest-R knowledge set. Figure 10 in Appendix E shows KIAN's results, which resemble the left heatmap of Figure 6. As training progresses, TC-KIAN learns to assign similar keys to similar knowledge policies and vice versa, helping the agents distinguish the functionality of knowledge policies. On the other hand, KIAN continues to see all keys as being similar. These results show that our trade-and-curate mechanism rearranges the knowledge representation space to easily identify helpful knowledge policies.

Trade Loss Dynamics. In addition to the mathematical properties of $\mathcal{L}_{trade}$ discussed in Section 4 and Appendix A, we empirically show that $\mathcal{L}_{trade}$ decays stably during training in Figure 5c. This indicates that aligning the knowledge representation space with the policy information space while optimizing the expected reward does not lead to unstable $\mathcal{L}_{trade}$. Moreover, Figure 5c shows that $\mathcal{L}_{trade}$ naturally decays for KIAN without explicit minimization. This implies that minimizing the trade loss can facilitate the decay process and explains TC-KIAN's improvement in Figures 3 and 4.

5.4 Ablation Study

While Section 4.2 has discussed the necessity of combining the trade loss and curated policy fusion, we verify their impact individually. We compare the performance of TC-KIAN, TC-KIAN without the trade loss, and TC-KIAN without curated policy fusion, across 4 environments under the KS-

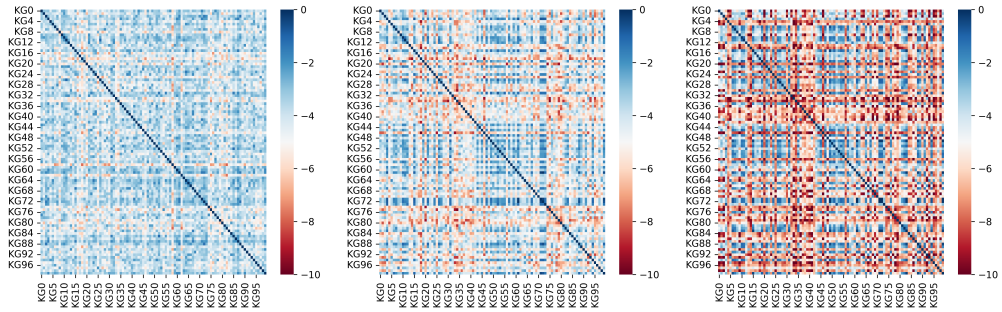


Figure 6: Knowledge key similarities, $S_{\mathbf{k}}^{ij}$, during TC-KIAN training (beginning, middle, end). The trade loss effectively distinguishes knowledge keys based on functional similarities of their policies.

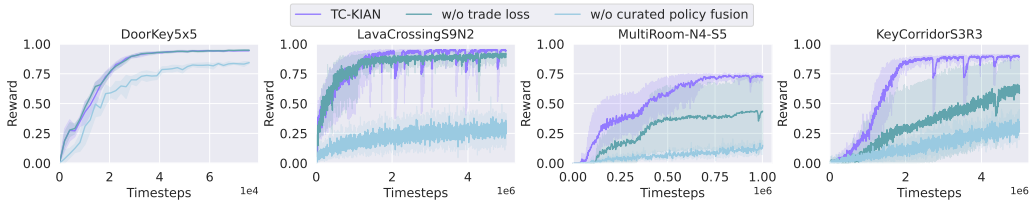


Figure 7: Ablation study of the trade loss and curated policy fusion. Without policy curation, the performance degrades since the feasible input space for weight calculation is restricted, as explained in Section 4.2. Without the trade loss, training variance becomes high. Only by combining the trade and curate mechanisms can we simultaneously achieve high rewards and low-variance training.

all setting (Figure 7). Without the trade loss, the curated policy fusion's decisiveness in choosing knowledge policies occasionally yields higher rewards but also induces high variance. The trade loss stabilizes the training by reducing the variance, especially in more challenging environments. Without curated policy fusion, performance is worse because the effective input space of `softmax` for weight calculation is largely restricted, and adding the trade loss further worsens the situation as the knowledge keys are forced to separate from one another in the already restricted space. These results prove the importance of combining the trade and curate mechanisms.

The tSNE visualization of the knowledge representation space under the Extra-stresstest-R setting in Figure 8 also supports the importance of optimizing the trade loss. With Extra-stresstest-R, the agent has to identify multiple duplicated knowledge policies. The trade loss groups the keys more meaningfully, whereas they are completely mixed without the trade mechanism. Such knowledge key distribution in the representation space explains some of TC-KIAN's performance gain and suggests potential future reuse of knowledge keys across tasks.

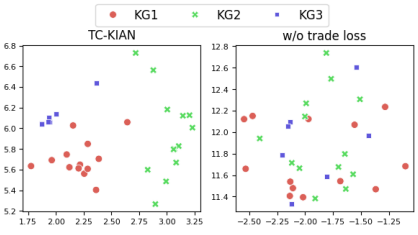


Figure 8: tSNE plot of keys. $\mathcal{L}_{trade}$ helps meaningful key separation.

6 Conclusion

We propose the trade-and-curate mechanism, which includes a novel trade loss and a policy fusion method, to achieve knowledge preselection-free KGRL training. Experiments show that our approach, TC-KIAN, improves the average reward by up to 89.6% over existing methods in challenging environments without an expert-picked knowledge set. The analyses suggest that our method opens a viable path to unleashing KGRL's effectiveness in broader applications when only raw, unfiltered knowledge is available. Future work will extend our framework to more knowledge modalities and investigate the scenarios when knowledge suggests exploratory or regulatory behavior.

AI USE STATEMENT

We have not used generative AI tools for any part of this research.

REPRODUCIBILITY STATEMENT

We provide our source code through the following link: `https://anonymous.4open.science/r/TC-KIAN`. In addition, the detailed algorithm, hyperparameters used, and environmental setup are reported in Appendix B, C, and D.

REFERENCES

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

J Bagnell, Joel Chestnutt, David Bradley, and Nathan Ratliff. Boosting structured prediction for imitation learning. *Advances in Neural Information Processing Systems*, 19, 2006.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and J K Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *Advances in Neural Information Processing Systems*, 36:73383–73394, 2023.

Zih-Yun Chiu, Yi-Lin Tuan, William Yang Wang, and Michael Yip. Flexible attention-based multi-policy fusion for efficient deep reinforcement learning. *Advances in Neural Information Processing Systems*, 36:13590–13612, 2023.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Guillaume Garrigos and Robert M Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

Milica Gašić, Filip Jurčíček, Blaise Thomson, Kai Yu, and Steve Young. On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pp. 312–317. IEEE, 2011.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. A knowledge-grounded neural conversation model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Vinicius G Goecks, Gregory M Gremillion, Vernon J Lawhern, John Valasek, and Nicholas R Waytowich. Integrating behavior cloning and reinforcement learning for improved performance in dense and sparse reward environments. *arXiv preprint arXiv:1910.04281*, 2019.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Yun-Jie Ho, Zih-Yun Chiu, Yuheng Zhi, and Michael C Yip. Surgirl: Toward life-long learning for surgical automation by incremental reinforcement learning. *IEEE Robotics and Automation Letters*, 10(12):13145–13152, 2025.

Xixi Hu, Bo Liu, Xingchao Liu, and Qiang Liu. Adaflow: Imitation learning with variance-adaptive flow-based policies. *Advances in Neural Information Processing Systems*, 37:138836–138858, 2024.

Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021.

Harold Jeffreys. *The theory of probability*. OuP Oxford, 1998.

Leslie Pack Kaelbling. The foundation of efficient robot learning. *Science*, 369(6506):915–916, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202, 2016.

Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10(105):eads5033, 2025.

Jun Morimoto and Kenji Doya. Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. *Robotics and Autonomous Systems*, 36(1):37–51, 2001.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6292–6299. IEEE, 2018.

Nicolás Navarro-Guerrero, Cornelius Weber, Pascal Schroeter, and Stefan Wermter. Real-world reinforcement learning for autonomous humanoid robot docking. *Robotics and Autonomous Systems*, 60(11):1400–1407, 2012.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3419–3448, 2022.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2463–2473, 2019.

Mark Pfeiffer, Samarth Shukla, Matteo Turchetta, Cesar Cadena, Andreas Krause, Roland Siegwart, and Juan Nieto. Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations. *IEEE Robotics and Automation Letters*, 3(4): 4423–4430, 2018.

Ahmed Hussain Qureshi, Yutaka Nakamura, Yuichiro Yoshikawa, and Hiroshi Ishiguro. Intrinsically motivated reinforcement learning for human–robot interaction in the real-world. *Neural Networks*, 107:23–33, 2018.

Janarthanan Rajendran, Aravind Lakshminarayanan, Mitesh M. Khapra, Prasanna P, and Balaraman Ravindran. Attend, adapt and transfer: Attentive deep architecture for adaptive transfer from multiple sources in the same domain. In *International Conference on Learning Representations*, 2017.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

Martin Riedmiller, Thomas Gabel, Roland Hafner, and Sascha Lange. Reinforcement learning for robot soccer. *Autonomous Robots*, 27(1):55–73, 2009.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Yi-Lin Tuan, Yun-Nung Chen, and Hung-yi Lee. Dykgchat: Benchmarking dialogue generation grounding on dynamic knowledge graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1855–1865, 2019.

Yi-Lin Tuan, Wei Wei, and William Yang Wang. Knowledge injection into dialogue generation via language models. *arXiv preprint arXiv:2004.14614*, 2020.

Yi-Lin Tuan, Connor Pryor, Wenhu Chen, Lise Getoor, and William Yang Wang. Local explanation of dialogue response generation. *Advances in Neural Information Processing Systems*, 34:404–416, 2021.

Yi-Lin Tuan, Zih-Yun Chiu, and William Yang Wang. Dynamic latent separation for deep learning. *arXiv preprint arXiv:2210.03728*, 2022.

Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. In *Conference on Robot Learning*, pp. 201–221. PMLR, 2023.

Daniel M Wegner, Toni Giuliano, and Paula T Hertel. Cognitive interdependence in close relationships. In *Compatible and incompatible relationships*, pp. 253–276. Springer, 1985.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. In *International Conference on Machine Learning*, pp. 6818–6827. PMLR, 2019.

Peng Zhang, Jianye Hao, Weixun Wang, Hongyao Tang, Yi Ma, Yihai Duan, and Yan Zheng. Kogun: Accelerating deep reinforcement learning via integrating human suboptimal knowledge. In *International Joint Conference on Artificial Intelligence*, 2020.

A PROPERTIES, THEOREM, AND PROOFS

A.1 PROPERTY 1

Property 1 (Feasibility of Trade Loss)**.** *Given n policies, $\pi_1, \ldots, \pi_n$, and their corresponding keys to be optimized, $\mathbf{k}_1, \ldots, \mathbf{k}_n$, assume the following condition holds*

$$\arg\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} \sum_{\substack{(i,j)\\ i\neq j\\ D_\pi^{ij}<th}} -\|\mathbf{k}_i-\mathbf{k}_j\|_2 \left(D_\pi^{ij}-th\right) \bigcap \arg\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} \sum_{\substack{(i,j)\\ i\neq j\\ D_\pi^{ij}>th}} -\|\mathbf{k}_i-\mathbf{k}_j\|_2 \left(D_\pi^{ij}-th\right) \neq \emptyset. \quad (8)$$

Then, if there exists a unique set of keys, $\mathcal{K}^* = \{\mathbf{k}_1^*, \ldots, \mathbf{k}_n^*\}$, such that $\mathcal{K}^*$ and its image $T(\mathcal{K}^*)$ under a Euclidean isometry T minimize the loss, $\mathcal{L}_{trade}$, they will assign the same key to policies that are sufficiently close and different keys to policies that are sufficiently different, i.e., for any pair of policies, (π_i, π_j), where $i \neq j$,

$$\begin{cases} \mathbf{k}_i^* = \mathbf{k}_j^*, & \text{if } D_\pi^{ij} < th \\ \mathbf{k}_i^* \neq \mathbf{k}_j^*, & \text{if } D_\pi^{ij} > th \end{cases}. \quad (9)$$

Proof. Define the following two functions, $f(\mathcal{K})$ and $g(\mathcal{K})$, with $\mathcal{K} = \{\mathbf{k}_1, \ldots, \mathbf{k}_n\}$:

$$f(\mathcal{K}) = \sum_{\substack{(i,j)\\ i\neq j\\ D_\pi^{ij}<th}} -\|\mathbf{k}_i-\mathbf{k}_j\|_2 \left(D_\pi^{ij}-th\right), \; g(\mathcal{K}) = \sum_{\substack{(i,j)\\ i\neq j\\ D_\pi^{ij}>th}} -\|\mathbf{k}_i-\mathbf{k}_j\|_2 \left(D_\pi^{ij}-th\right). \quad (10)$$

Then, we can write

$$\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} \mathcal{L}_{trade} = \min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} \left(f(\mathcal{K}) + g(\mathcal{K})\right) \geq \min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} f(\mathcal{K}) + \min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} g(\mathcal{K}). \quad (11)$$

The inequality in (11) can be proved by contradiction, and we omit the details of this proof here.

The assumption (8) suggests that $f(\mathcal{K})$ and $g(\mathcal{K})$ has the same minimizer, i.e., there exists $\mathcal{K}^* = \{\mathbf{k}_1^*, \ldots, \mathbf{k}_n^*\}$ such that

$$\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} f(\mathcal{K}) + \min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} g(\mathcal{K}) = f(\mathcal{K}^*) + g(\mathcal{K}^*) \geq \min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} \left(f(\mathcal{K}) + g(\mathcal{K})\right). \quad (12)$$

Combining the inequalities in (11) and (12), we have

$$\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} \left(f(\mathcal{K}) + g(\mathcal{K})\right) = \min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} f(\mathcal{K}) + \min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} g(\mathcal{K}) = f(\mathcal{K}^*) + g(\mathcal{K}^*). \quad (13)$$

Since we assume that $\mathcal{L}_{trade}$ has a unique solution for $\mathcal{K}$ and that $f(\mathcal{K})$ and $g(\mathcal{K})$ shares the same minimizer, we also have

$$\mathcal{K}^* = \arg\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} f(\mathcal{K}) = \arg\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} g(\mathcal{K}) = \arg\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} \left(f(\mathcal{K}) + g(\mathcal{K})\right). \quad (14)$$

Therefore, we can use the minimizer of $f(\mathcal{K})$ and $g(\mathcal{K})$ to infer the minimizer of $f(\mathcal{K}) + g(\mathcal{K})$.

Minimizing $f(\mathcal{K})$ will assign the same key to policies that are sufficiently close, shown as follows:

$$\arg\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} f(\mathcal{K}) = \arg\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} \sum_{\substack{(i,j)\\ i\neq j\\ D_\pi^{ij}<th}} -\|\mathbf{k}_i-\mathbf{k}_j\|_2 \left(D_\pi^{ij}-th\right) \quad (15)$$

$$= \arg\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} \sum_{\substack{(i,j)\\ i\neq j\\ D_\pi^{ij}<th}} \|\mathbf{k}_i-\mathbf{k}_j\|_2. \quad (16)$$

Equation (16) holds since $-\left(D_\pi^{ij}-th\right) > 0$ and $\|\mathbf{k}_i-\mathbf{k}_j\|_2 \geq 0$ for all (i,j) considered by $f(\mathcal{K})$. We therefore conclude that, for all (π_i, π_j) such that $D_\pi^{ij} < th$, minimizing $f(\mathcal{K})$ gives $\mathbf{k}_i = \mathbf{k}_j$.

Minimizing $g(\mathcal{K})$ will assign different keys to sufficiently different policies, shown as follows:

$$\arg\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} g(\mathcal{K}) = \arg\min_{\mathbf{k}_1,\ldots,\mathbf{k}_n} \sum_{\substack{(i,j)\\ i\neq j\\ D_\pi^{ij}>th}} -\|\mathbf{k}_i-\mathbf{k}_j\|_2 \left(D_\pi^{ij}-th\right) \tag{17}$$

$$= \arg\max_{\mathbf{k}_1,\ldots,\mathbf{k}_n} \sum_{\substack{(i,j)\\ i\neq j\\ D_\pi^{ij}>th}} \|\mathbf{k}_i-\mathbf{k}_j\|_2. \tag{18}$$

Equation (18) holds since $-\left(D_\pi^{ij}-th\right)<0$ and $\|\mathbf{k}_i-\mathbf{k}_j\|_2\geq 0$ for all (i,j) considered by $g(\mathcal{K})$. We therefore conclude that, for all (π_i,π_j) such that $D_\pi^{ij}>th$, minimizing $g(\mathcal{K})$ separates $\mathbf{k}_i$ and $\mathbf{k}_j$ as much as possible.

With equation (14), minimizing $\mathcal{L}_{trade}$ leads to that, for any pair of policies, (π_i,π_j), where $i\neq j$,

$$\begin{cases} \mathbf{k}_i^*=\mathbf{k}_j^*, & \text{if } D_\pi^{ij}<th \\ \mathbf{k}_i^*\neq\mathbf{k}_j^*, & \text{if } D_\pi^{ij}>th \end{cases}. \tag{19}$$

□

A.2 THEOREM 1

Theorem 1 (Structural Convergence of Trade Loss)**.** *Consider any two knowledge policies π_i and π_j fused in $\mathcal{L}_{trade}$:*

$$\mathcal{L}_{trade} = \mathop{\mathbb{E}}_{\mathbf{s}'\sim\mathcal{T}^\pi(\cdot|\mathbf{s})}\left[-sg\left(D_\pi^{i,j}(\mathbf{s}')-th\right)\|\mathbf{k}_i-\mathbf{k}_j\|_2\right]. \tag{20}$$

If $\mathbf{k}_i,\mathbf{k}_j$ is bounded by a r_k-sphere, i.e., $\|\mathbf{k}_i\|_2\leq r_k, \|\mathbf{k}_j\|_2\leq r_k, \forall i,j,\ \|\mathbf{k}_i-\mathbf{k}_j\|_2\geq\delta$, and $\inf|D_\pi^{i,j}(\mathbf{s}')-th|>0$, then solving $\arg\min_{\mathbf{k}_i,\mathbf{k}_j}\mathcal{L}_{trade}$ with stochastic gradient descent (SGD) will converge.

To prove Theorem 1, we will first prove Lemma 1-3.

Lemma 1. *If $\|\mathbf{k}_i\|_2\leq r_k, \|\mathbf{k}_j\|_2\leq r_k, \forall i,j$, then $l_{\mathbf{s}_t}(\mathbf{k}_i,\mathbf{k}_j)\triangleq-\left(D_\pi^{ij}(\mathbf{s}_t)-th\right)\|\mathbf{k}_i-\mathbf{k}_j\|_2$ is bounded from below.*

Proof. We consider the cases where $D_\pi^{ij}(\mathbf{s}_t)<th$ and $D_\pi^{ij}(\mathbf{s}_t)>th$.

If $D_\pi^{i,j}(\mathbf{s}_t)<th$, then

$$l_{\mathbf{s}_t}(\mathbf{k}_i,\mathbf{k}_j) = -\left(D_\pi^{ij}(\mathbf{s}_t)-th\right)\|\mathbf{k}_i-\mathbf{k}_j\|_2\geq 0. \tag{21}$$

If $D_\pi^{i,j}(\mathbf{s}_t)>th$, then by triangle inequality, we have

$$\|\mathbf{k}_i-\mathbf{k}_j\|_2\leq\|\mathbf{k}_i\|_2+\|\mathbf{k}_j\|_2\leq 2r_k \tag{22}$$

$$l_{\mathbf{s}_t}(\mathbf{k}_i,\mathbf{k}_j) = -\left(D_\pi^{ij}(\mathbf{s}_t)-th\right)\|\mathbf{k}_i-\mathbf{k}_j\|_2\geq -2r_k\left(D_\pi^{ij}(\mathbf{s}_t)-th\right). \tag{23}$$

Hence, $l_{\mathbf{s}_t}(\mathbf{k}_i,\mathbf{k}_j)$ is bounded from below. □

Lemma 2. *If $\|\mathbf{k}_i-\mathbf{k}_j\|_2\geq\delta$, then $S_{\mathbf{k}}^{ij}$ is $\frac{4}{\delta}$-smooth, i.e.,*

$$\forall\,\mathbf{k},\mathbf{k}'\in\mathbb{R}^{2\times d_k},\quad \left\|\nabla S_{\mathbf{k}}^{ij}-\nabla S_{\mathbf{k}'}^{ij}\right\|_2\leq\frac{4}{\delta}\|\mathbf{k}-\mathbf{k}'\|_2, \tag{24}$$

where $\mathbf{k}\triangleq[\mathbf{k}_i^\top\ \mathbf{k}_j^\top]^\top$ and $\mathbf{k}'\triangleq[\mathbf{k}_i'^\top\ \mathbf{k}_j'^\top]^\top$.

Proof. The gradients of $S_{\mathbf{k}}^{ij}$ with respect to $\mathbf{k}_i$ and $\mathbf{k}_j$ are

$$\nabla_{\mathbf{k}_i}S_{\mathbf{k}}^{ij}=\frac{-(\mathbf{k}_i-\mathbf{k}_j)}{\|\mathbf{k}_i-\mathbf{k}_j\|_2},\quad \nabla_{\mathbf{k}_j}S_{\mathbf{k}}^{ij}=\frac{\mathbf{k}_i-\mathbf{k}_j}{\|\mathbf{k}_i-\mathbf{k}_j\|_2}. \tag{25}$$

Let $\mathbf{v} = \mathbf{k}_i - \mathbf{k}_j$. We can rewrite equation (25) as

$$\nabla_{\mathbf{k}_i} S^{ij}_{\mathbf{k}} = \frac{-\mathbf{v}}{\|\mathbf{v}\|_2} = -\psi(\mathbf{v}), \quad \nabla_{\mathbf{k}_j} S^{ij}_{\mathbf{k}} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2} = \psi(\mathbf{v}). \tag{26}$$

If $\|\mathbf{v}\|_2 \geq \delta$, then $\psi(\mathbf{v})$ is Lipschitz with the local Lipschitz constant being $\frac{2}{\delta}$ by the definition of Lipschitz continuity.

The left-hand side of equation (24) becomes

$$\left\| \nabla S^{ij}_{\mathbf{k}} - \nabla S^{ij}_{\mathbf{k}'} \right\|_2 = \sqrt{2} \left\| \psi(\mathbf{v}) - \psi(\mathbf{v}') \right\|_2 \leq \frac{2\sqrt{2}}{\delta} \left\| \mathbf{v} - \mathbf{v}' \right\|_2, \tag{27}$$

where $\mathbf{v}' = \mathbf{k}'_i - \mathbf{k}'_j$. In addition, with the triangle and AM-QM inequalities, we have

$$\left\| \mathbf{v} - \mathbf{v}' \right\|_2 \leq \left\| \mathbf{k}_i - \mathbf{k}'_i \right\|_2 + \left\| \mathbf{k}_j - \mathbf{k}'_j \right\|_2 \leq \sqrt{2} \left\| \mathbf{k} - \mathbf{k}' \right\|_2. \tag{28}$$

Combining equations (27) and (28), we get

$$\left\| \nabla S^{ij}_{\mathbf{k}} - \nabla S^{ij}_{\mathbf{k}'} \right\|_2 \leq \frac{4}{\delta} \left\| \mathbf{k} - \mathbf{k}' \right\|_2. \tag{29}$$

□

Lemma 3. *If* $\|\mathbf{k}_i\|_2 \leq r_k$, $\|\mathbf{k}_j\|_2 \leq r_k, \forall i, j$, *and* $\|\mathbf{k}_i - \mathbf{k}_j\|_2 \geq \delta$, *then* $l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j) = -\left(D^{ij}_\pi(\mathbf{s}_t) - th\right) \|\mathbf{k}_i - \mathbf{k}_j\|_2$ *is* ν*-Polyak-Łojasiewicz, i.e., there exists* $\nu > 0$ *such that*

$$\forall (\mathbf{k}_i, \mathbf{k}_j), \quad l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j) - \inf\ l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j) \leq \frac{1}{2\nu} \left\| \nabla_{\mathbf{k}} l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j) \right\|_2^2, \tag{30}$$

where $\mathbf{k} = [\mathbf{k}_i^\top \mathbf{k}_j^\top]^\top$.

Proof. According to equation (25), we have

$$\left\| \nabla_{\mathbf{k}} l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j) \right\|_2^2 = 2 \left(D^{ij}_\pi(\mathbf{s}_t) - th \right)^2. \tag{31}$$

Next, we consider the cases where $D^{ij}_\pi(\mathbf{s}_t) < th$ and $D^{ij}_\pi(\mathbf{s}_t) > th$. If $D^{ij}_\pi(\mathbf{s}_t) < th$, then according to equation (21), $\inf\ l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j) = 0$. By the triangle inequality,

$$-\left(D^{ij}_\pi(\mathbf{s}_t) - th\right) \|\mathbf{k}_i - \mathbf{k}_j\|_2 - \inf\ l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j) \leq -\left(D^{ij}_\pi(\mathbf{s}_t) - th\right) (\|\mathbf{k}_i\|_2 + \|\mathbf{k}_j\|_2) \tag{32}$$

$$\leq -2r_k \left(D^{ij}_\pi(\mathbf{s}_t) - th\right). \tag{33}$$

Hence, if $D^{ij}_\pi(\mathbf{s}_t) < th$, $l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j)$ is ν-Polyak-Łojasiewicz with $\nu = \frac{-1}{2r_k}\left(D^{ij}_\pi(\mathbf{s}_t) - th\right)$.

If $D^{ij}_\pi(\mathbf{s}_t) > th$, then according to equation (23), $\inf\ l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j) = -2r_k \left(D^{ij}_\pi(\mathbf{s}_t) - th\right)$, and

$$-\left(D^{ij}_\pi(\mathbf{s}_t) - th\right) \|\mathbf{k}_i - \mathbf{k}_j\|_2 - \inf\ l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j) = \left(D^{ij}_\pi(\mathbf{s}_t) - th\right) (2r_k - \|\mathbf{k}_i - \mathbf{k}_j\|_2) \tag{34}$$

$$\leq 2r_k \left(D^{ij}_\pi(\mathbf{s}_t) - th\right). \tag{35}$$

Hence, if $D^{ij}_\pi(\mathbf{s}_t) > th$, $l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j)$ is ν-Polyak-Łojasiewicz with $\nu = \frac{1}{2r_k}\left(D^{ij}_\pi(\mathbf{s}_t) - th\right)$. □

Theorem 1 (Structural Convergence of Trade Loss)**.** *Consider any two knowledge policies* π_i *and* π_j *fused in* $\mathcal{L}_{trade}$*:*

$$\mathcal{L}_{trade} = \mathop{\mathbb{E}}_{\mathbf{s}' \sim \mathcal{T}^\pi(\cdot|\mathbf{s})} \left[-sg\left(D^{i,j}_\pi(\mathbf{s}') - th\right) \|\mathbf{k}_i - \mathbf{k}_j\|_2 \right]. \tag{36}$$

If $\mathbf{k}_i, \mathbf{k}_j$ *is bounded by a* r_k*-sphere, i.e.,* $\|\mathbf{k}_i\|_2 \leq r_k, \|\mathbf{k}_j\|_2 \leq r_k, \forall i, j$, $\|\mathbf{k}_i - \mathbf{k}_j\|_2 \geq \delta$, *and* $\inf |D^{i,j}_\pi(\mathbf{s}') - th| > 0$, *then solving* $\arg\min_{\mathbf{k}_i, \mathbf{k}_j} \mathcal{L}_{trade}$ *with stochastic gradient descent (SGD) will converge.*

Proof. We use Theorem 5.10 (Convergence for Polyak-Łojasiewicz and Smooth Functions) in (Garrigos & Gower, 2023) to prove the convergence of $\mathcal{L}_{trade}$.

With Lemma 1 and 2, we can show that $\forall \mathbf{s}_t \in \mathcal{S}$, $l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j) = -\left(D^{ij}_\pi(\mathbf{s}_t) - th\right) \|\mathbf{k}_i - \mathbf{k}_j\|_2$ is bounded from below and $L_{\mathbf{s}_t}$-smooth, with $L_{\mathbf{s}_t} = |D^{ij}_\pi(\mathbf{s}_t) - th| \frac{4}{\delta}$.

With Lemma 3, we can show that $\forall \mathbf{s}_t \in \mathcal{S}$, $l_{\mathbf{s}_t}(\mathbf{k}_i, \mathbf{k}_j)$ is ν-Polyak-Łojasiewicz, with $\nu = |\frac{1}{2r_k}(D^{ij}_\pi(\mathbf{s}_t) - th)|$. Therefore, according to Theorem 5.10 in (Garrigos & Gower, 2023), we can conclude that solving $\arg\min_{\mathbf{k_i}, \mathbf{k}_j} \mathcal{L}_{trade}$ with SGD will converge. □

A.3 PROPERTY 2

Property 2. *(Knowledge Compositionality) The linear-combination relationships of knowledge policies are transferable to knowledge keys under Curated Policy Fusion. Without loss of generality, for any three knowledge policies, π_i, π_j, and π_c, if the knowledge policy π_c is a state-dependent combination of policies π_i and π_j, i.e.:*

$$\pi_c(\mathbf{a}_t|\mathbf{s}_t) := \hat{w}_i(\mathbf{s}_t)\pi_i(\mathbf{a}_t|\mathbf{s}_t) + \hat{w}_j(\mathbf{s}_t)\pi_j(\mathbf{a}_t|\mathbf{s}_t),\ 0 \le \hat{w}_i(\mathbf{s}_t), \hat{w}_j(\mathbf{s}_t) \le 1,\ \hat{w}_i(\mathbf{s}_t) + \hat{w}_j(\mathbf{s}_t) = 1, \forall t, \tag{37}$$

then there exists a $\mathbf{u}_t = \Phi(\mathbf{s}_t)$ and a $\hat{\mathbf{k}} \triangleq [\hat{\mathbf{k}}_i^\top\ \hat{\mathbf{k}}_j^\top\ \hat{\mathbf{k}}_c^\top]^\top$, such that:

$$\hat{\mathbf{k}}_c = c_i\hat{\mathbf{k}}_i + c_j\hat{\mathbf{k}}_j, \quad c_i, c_j \in \mathbb{R}. \tag{38}$$

Proof. Since $\pi_c(\mathbf{a}_t|\mathbf{s}_t) = \hat{w}_i(\mathbf{s}_t)\pi_i(\mathbf{a}_t|\mathbf{s}_t) + \hat{w}_j(\mathbf{s}_t)\pi_j(\mathbf{a}_t|\mathbf{s}_t)$, the probability of π_c generating a trajectory $\tau = (\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \ldots, \mathbf{a}_{T-1}, \mathbf{s}_T)$ becomes

$$P(\tau) = \rho(\mathbf{s}_0) \prod_{t=0}^{T-1} \pi_c(\mathbf{a}_t|\mathbf{s}_t)\mathcal{T}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \tag{39}$$

$$= \rho(\mathbf{s}_0) \prod_{t=0}^{T-1} \left(\hat{w}_i(\mathbf{s}_t)\pi_i(\mathbf{a}_t|\mathbf{s}_t) + \hat{w}_j(\mathbf{s}_t)\pi_j(\mathbf{a}_t|\mathbf{s}_t)\right) \mathcal{T}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t). \tag{40}$$

Equations (39) and (40) imply that to generate the trajectory τ with curated policy fusion, the query vector $\mathbf{u}_t$'s attention to (π_i, π_j) and to π_c satisfies the relationship in equation (41):

$$\frac{\texttt{softmax}(\mathbf{u}_t \cdot \hat{\mathbf{k}}_i)}{\texttt{softmax}(\mathbf{u}_t \cdot \hat{\mathbf{k}}_c)} = \hat{w}_i(\mathbf{s}_t) \quad , \frac{\texttt{softmax}(\mathbf{u}_t \cdot \hat{\mathbf{k}}_j)}{\texttt{softmax}(\mathbf{u}_t \cdot \hat{\mathbf{k}}_c)} = \hat{w}_j(\mathbf{s}_t) \tag{41}$$

By the definition of $\texttt{softmax}(\cdot)$, we can rewrite it as:

$$\mathbf{u}_t^\top \mathbf{k}_i - \mathbf{u}_t^\top \mathbf{k}_c = \log \hat{w}_i(\mathbf{s}_t) \quad , \mathbf{u}_t^\top \mathbf{k}_j - \mathbf{u}_t^\top \mathbf{k}_c = \log \hat{w}_j(\mathbf{s}_t) \tag{42}$$

Next, we show that there exists $\mathbf{u}_t, \forall t, c_i, c_j$, such that $\hat{\mathbf{k}}_c = c_i\hat{\mathbf{k}}_i + c_j\hat{\mathbf{k}}_j$ and equation (42) hold. By rewriting equation (42), we have

$$\begin{cases} \mathbf{u}_t^\top \left(\hat{\mathbf{k}}_i - \hat{\mathbf{k}}_c\right) = \log \hat{w}_i \\ \mathbf{u}_t^\top \left(\hat{\mathbf{k}}_j - \hat{\mathbf{k}}_c\right) = \log \hat{w}_j \end{cases} \tag{43}$$

$$= \begin{cases} \mathbf{u}_t^\top \left[(1-c_i)\hat{\mathbf{k}}_i - c_j\hat{\mathbf{k}}_j\right] = \log \hat{w}_i \\ \mathbf{u}_t^\top \left[-c_i\hat{\mathbf{k}}_i + (1-c_j)\hat{\mathbf{k}}_j\right] = \log \hat{w}_j \end{cases}. \tag{44}$$

$$\tag{45}$$

The above equations can be written as:

$$\begin{bmatrix} 1-c_i & -c_j \\ -c_i & 1-c_j \end{bmatrix} \begin{bmatrix} \mathbf{u}_t^\top \hat{\mathbf{k}}_i \\ \mathbf{u}_t^\top \hat{\mathbf{k}}_j \end{bmatrix} = \begin{bmatrix} \log \hat{w}_i \\ \log \hat{w}_j \end{bmatrix}. \tag{46}$$

Solution exists for $\mathbf{u}_t^\top \hat{\mathbf{k}}_i$ and $\mathbf{u}_t^\top \hat{\mathbf{k}}_j$ in equation (46) under two cases. First, if $c_i + c_j \neq 1$, a unique solution exists, since:

$$\det\left(\begin{bmatrix} 1-c_i & -c_j \\ -c_i & 1-c_j \end{bmatrix}\right) = 1 - c_i - c_j \neq 0. \tag{47}$$

Second, if the determinant is 0, c_i, c_j exist for:

$$\frac{1-c_i}{-c_i} = \frac{-c_j}{1-c_j} = \frac{\log w_i}{\log w_j}, \tag{48}$$

and infinite solutions of $\mathbf{u}_t^\top \hat{\mathbf{k}}_i$ and $\mathbf{u}_t^\top \hat{\mathbf{k}}_j$ exist for equation (46). □

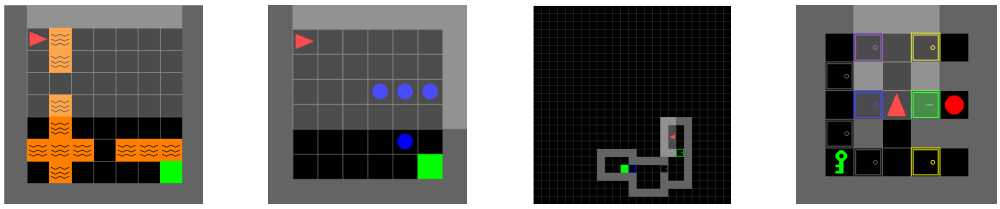
B ALGORITHM OF TC-KIAN

Algorithm 1: TC-KIAN

Given an environment $\mathcal{E} = \texttt{KGMDP}(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{G}, \mathcal{R}, \rho, \gamma)$, where $\mathcal{G} = \{\pi_{g_1}, \pi_{g_2}, \ldots, \pi_{g_n}\}$
Initialize the inner actor $\pi_{in}(\cdot|\cdot;\boldsymbol{\theta})$, query $\Phi(\cdot;\boldsymbol{\phi})$, and policy keys $\mathbf{k}_i, \forall i \in \{in, g_1, \ldots, g_n\}$
Observe a state $\mathbf{s}_0 \in \mathcal{S}$ from $\mathcal{E}$
for *each time step* t **do**
    Calculate the trade loss $\mathcal{L}_{trade}(\mathbf{s}_t)$ in Eq 3.
    Obtain the policy $\pi(\mathbf{s}_t)$ by curated policy fusion in Eq 7.
    Sample an action $\mathbf{a}_t \sim \pi(\cdot|\mathbf{s}_t)$.
    Apply $\mathbf{a}_t \in \mathcal{A}$ to $\mathcal{E}$ and observe the reward and next state $(R_t, \mathbf{s}_{t+1})$.
    Update $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, and $\mathbf{k}_{in}, \mathbf{k}_{g_1}, \ldots, \mathbf{k}_{g_n}$ by $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi},\mathbf{k}_i}(\mathcal{L}_{RL} + \mathcal{L}_{trade})$.

C EXPERIMENT DETAILS

MiniGrid Environments. Our experiments are conducted on MiniGrid environments (`https://github.com/maximecb/gym-minigrid` and `https://minigrid.farama.org/`). In MiniGrid, a state $\mathbf{s}_t$ is a directed first-person view represented as a 5x5 grid. An action $\mathbf{a}_t$ in each task is one of seven discrete actions: left, right, forward, pickup, drop, toggle, and done.



LavaCrossingS9N2 | Dynamic-Obstacles-8x8 | MultiRoom-N4-S5 | KeyCorridorS3R3

Figure 9: Examples of MiniGrid environments.

Implementation and Training Details. We implement the algorithms in PyTorch (Paszke et al., 2019). All experiments in Section 5 are run with five random seeds on a single NVIDIA GeForce RTX 4090 and reported with a 95% confidence interval. The analyses in Sections 5.3 and 5.4, if not specified, are run on DoorKey-5x5.

Model Architecture Details. We use the same model architecture (convolutional neural networks followed by multi-layer perceptrons) for all experiments as reported in prior work and used in the public repositories `https://github.com/lcswillems/rl-starter-files` and `https://github.com/Pascalson/KGRL`. The architecture includes an image encoder, actor, and critic. **[Image Encoder]** The image encoder is a three-layer convolutional neural network that maps an input image to an image embedding, which is used as the state by the actor and critic networks. **[Critic]** The critic network consists of one 64x64 hidden linear layer with Tanh activation and outputs a scalar. **[Actor]** The actor network begins with a shared 64x64 linear layer. The inner actor adds additional Tanh activation and one linear output layer for projection onto the action space. The query adds additional Tanh activation and one linear output layer for projection onto the d_k-dimensional embedding space. Each knowledge key, including $\mathbf{k}_{in}$ and $\mathbf{k}_{g_j}$, is a learnable d_K-dimensional vector implemented by `nn.Embedding` in Pytorch.

Hyperparameters. Following prior work (Chiu et al., 2023), the hyperparameters are: learning rates 1×10^{-3}, batch sizes 256, and discount factors $\gamma = 0.99$. The learnable parameters are updated with the Adam optimizer (Kingma & Ba, 2014). We search the entropy coefficient among $\{0, 0.001, 0.01\}$.

D KGRL MINIGRID KNOWLEDGE SET

We use the knowledge set $\mathcal{G}$ defined in (Chiu et al., 2023) and `https://github.com/Pascalson/KGRL` for KGRL MiniGrid tasks. It comprises eight suboptimal programs that cannot complete any MiniGrid tasks on their own. We use the same notation as in our experiments, especially Figures 5a-5b and Table 2, for clear reference. The inner actor and all knowledge policies take first-person views as input, covering only a small part of the environment.

- KG1 `pick_up_the_key (PK)`: If there exists a key in $\mathbf{s}_t$, move to the key; if the key is in front of the agent, $\pi_{g_j}(pickup|\mathbf{s}_t) = 1$.
- KG2 `open_the_door (OD)`: If there exists a door in $\mathbf{s}_t$, move to the door; if the door is in front of the agent, $\pi_{g_j}(toggle|\mathbf{s}_t) = 1$.
- KG3 `go_to_the_goal (G)`: If there exists a goal in $\mathbf{s}_t$, move to the goal.
- KG4 `do_not_hit`: If there are walls or lava around the agent, do not choose that direction.
- KG5 `do_not_hit_balls`: If there exist balls around the agent, do not choose that direction.
- KG6 `open_the_unlocked_door`: If there exists a unlocked door in $\mathbf{s}_t$, move to the door; if the door is in front of the agent, $\pi_{g_j}(toggle|\mathbf{s}_t) = 1$.
- KG7 `open_the_locked_door`: If there exists a locked door in $\mathbf{s}_t$, move to the door; if the door is in front of the agent, $\pi_{g_j}(toggle|\mathbf{s}_t) = 1$.
- KG8 `pick_up_the_ball`: If there exists a ball in $\mathbf{s}_t$, move to the key; if the key is in front of the agent, $\pi_{g_j}(pickup|\mathbf{s}_t) = 1$.

In the above policies, the "move to" is decided by $\mathbf{p}_{obj} - \mathbf{p}_{self}$, where $\mathbf{p}_{self} \in \mathbb{R}^3$ is the position of the agent and $\mathbf{p}_{obj} \in \mathbb{R}^3$ is the position of the object. The actions right, left, and forward in π_{g_j} can be written as:

$$\begin{cases} \pi_{g_j}(right|\mathbf{s}_t) = 1, & \text{if } \mathbf{p}_{obj,x} - \mathbf{p}_{self,x} > 0 \\ \pi_{g_j}(left|\mathbf{s}_t) = 1, & \text{if } \mathbf{p}_{obj,x} - \mathbf{p}_{self,x} < 0 \\ \pi_{g_j}(forward|\mathbf{s}_t) = 1, & \text{if } \mathbf{p}_{obj,y} - \mathbf{p}_{self,y} > 0 \end{cases} \tag{49}$$

The preselected knowledge policies (KS-preselected) for each environment are:

- Empty Environments: KG3
- Unlock Environments: KG1, KG2
- DoorKey Environments: KG1, KG2, KG3
- Dynamic Obstacles Environments: KG3, KG5
- LavaCrossing and FourRoom Environments: KG3, KG4
- MultiRoom Environments: KG3, KG4, KG6
- KeyCorridor Environments: KG1, KG6, KG7, KG8

E THE LEARNED KNOWLEDGE REPRESENTATION SPACE OF KIAN

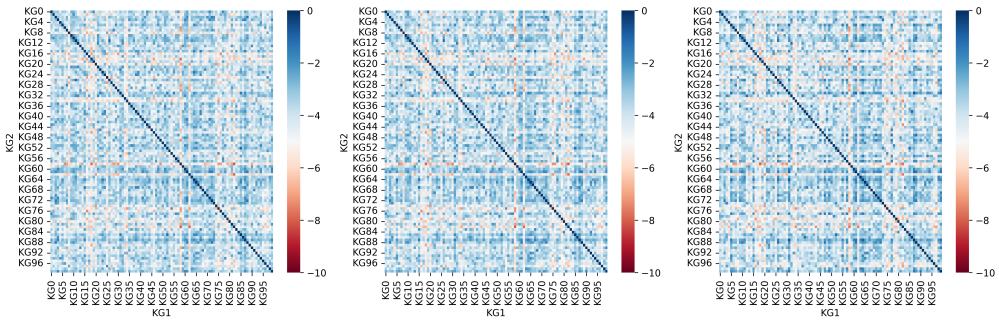


Figure 10: Knowledge key similarities, $S_{\mathbf{k}}^{ij}$, during KIAN training (beginning, middle, end).

F COMPUTATIONAL ANALYSES

Method	Knowledge Set	Average FPS ↑	wall-clock time (sec) ↓
KIAN	KS-all	1127.7±111.3	62.74
TC-KIAN	KS-all	1201.8±81.4	64.09
KIAN	Extra-stresstest-U	1207.2±139.2	63.87
TC-KIAN	Extra-stresstest-U	1193.1±122.3	64.68
KIAN	Extra-stresstest-R	1175.2±109.4	65.53
TC-KIAN	Extra-stresstest-R	1208.6±96.8	63.62

Table 1: Average frames per second (FPS) and wall-clock time. The wall-clock time is calculated for a training cycle that processes 75000 frames. The results show minimal computational overhead when optimizing $\mathcal{L}_{trade}$ simultaneously.

G ABLATION STUDY BY KNOWLEDGE PRUNING DURING INFERENCE

	w/o Trade Loss	TC-KIAN
N-PK	0.728	**0.794**
N-OD	**0.900**	0.840
N-G	0.804	**0.844**
N-Main	0.624	**0.688**
Sufficiency	0.720	**0.844**

Table 2: Necessity leave one knowledge out from the Pick Up Key set (PK), Open Door set (OD), and Go to Goal set (G), or the Main set (PK+OD+G). Sufficiency retains only one knowledge from each PK, OD, G set.

To further evaluate the properties of the knowledge keys learned with and without the trade loss, we study the knowledge's *Necessity* and *Sufficiency* based on concepts from Explainable AI (Tuan et al., 2021). Necessity and Sufficiency are respectively defined as: (1) *How is the agent affected after removing explanations?* and (2) *How does the agent perform when only the explanations are given?* We take each knowledge policy as an explanation unit.

Table 2 shows the Necessity and Sufficiency results for DoorKey-5x5 agents trained under the Extra-stresstest-R setting. We study the impacts of the following knowledge policies: PK, OD, and G. Please refer to Section D for the details of these knowledge policies. We note here that in Extra-stresstest-R, the same policy can appear multiple times. For Necessity, we remove one knowledge policy, either one of PK (N-PK), OD (N-OD), or G (N-G), from Extra-stresstest-R during inference and report the minimum average reward across 5 random seeds. In N-Main, we remove only one

knowledge policy, either PK, OD, or G, from Extra-stresstest-R during inference and report the minimum average reward across 5 random seeds. For Sufficiency, we keep only three knowledge policies, one from the group of PK, one from the group of OD, and one from the group of G, in Extra-stresstest-R during inference and report the average reward across 5 random seeds.

The results show that trade loss distributes weights more uniformly across similar knowledge policies, so removing one does not significantly affect performance. This is evident in its Necessity results, which generally outperform those without adding the trade loss. The trade loss also helps the weights concentrate on the useful knowledge policies, as indicated by the Sufficiency results. Reaching an average reward of 0.844 using only 3 knowledge policies, compared to 0.94 using all 100 knowledge policies in Extra-stresstest-R, suggests an effective way to reduce the number of knowledge policies during agent deployment.

H LIMITATION

This work presents TC-KIAN, a more efficient method to address the degraded performance of prior KGRL agents when facing a noisier knowledge set. While TC-KIAN performs better across various knowledge setups (KS-all, Extra-stresstest-R, Extra-stresstest-U), it still leaves room for improvement. TC-KIAN's learning-curve variance is sometimes higher than that of KIAN, the state-of-the-art KGRL method, showing that while it converges faster and achieves higher rewards, its stability can be improved in future work. Moreover, while we empirically verify TC-KIAN in diverse decision-making environments, future work can study more modalities of states, actions, or knowledge to investigate TC-KIAN's generalizability in broad real-world cases.