CONTRASTIVE REPRESENTATIONS FOR COMBINATORIAL REASONING

Alicja Ziarko
IDEAS NCBR
University of Warsaw
IMPAN

Michał Bortkiewicz
Warsaw University of Technology

Michał Zawalski
University of Warsaw

Benjamin Eysenbach †
Princeton University

Piotr Miłoś †
IDEAS NCBR
University of Warsaw
IMPAN, deepsense.ai

ABSTRACT

Contrastive learning (CL) has emerged as a powerful framework for learning structured representations that enable a wide range of downstream tasks. Its applications span sample-efficient reinforcement learning (RL), retrieval-augmented generation, and improved selection of model-generated samples, among others. Despite these successes, its potential for combinatorial reasoning problems remains largely untapped. In this paper, we take a step in this direction by using temporal contrastive learning to learn representations conducive to solving planning problems, which will reduce our reliance on planning. Our analysis reveals that standard CL approaches struggle to capture temporal dependencies over complex trajectories. To address this, we introduce a novel method that leverages negatives from the same trajectories. Across two complex reasoning tasks, our approach outperforms traditional supervised learning.

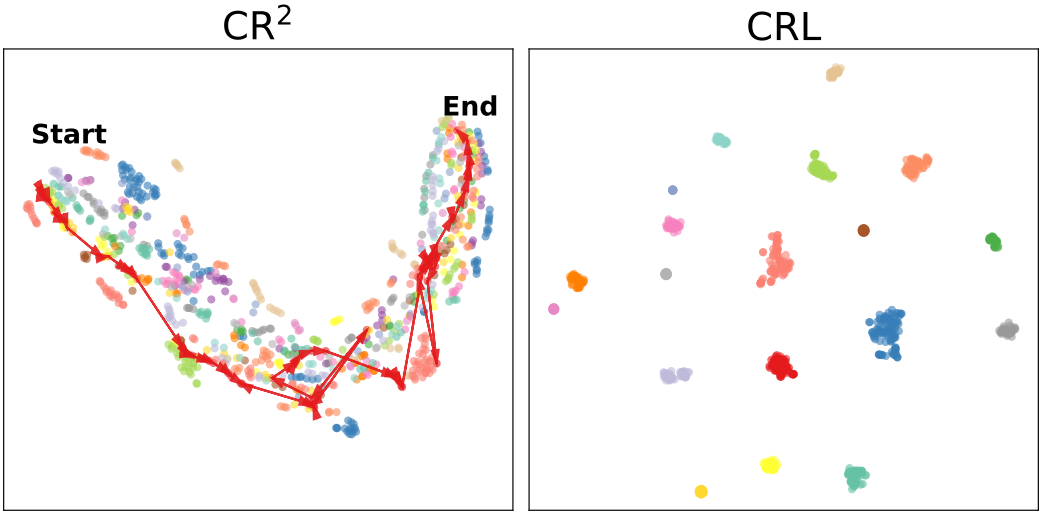

Figure 1: **CR^2 makes representations reflect the structure of the combinatorial task.** t-SNE visualization of representations learned by CR^2 (left) and naive CRL (right) for Sokoban, with one trajectory highlighted using arrows connecting consecutive points. Colors correspond to trajectories. Baseline representations (right) cluster within trajectories, making them useless for planning.

† Equal advising contribution.

1 INTRODUCTION

Representation learning has driven advances across AI, particularly in vision and language Bengio et al. (2013); Devlin et al. (2019); Huang et al. (2019); Grill et al. (2020). In combinatorial reasoning problems (e.g., Chess, Go, TSP), search remains a core component, often combined with learned value functions or representations Silver et al. (2016); Brown et al. (2020); Yonetani et al. (2021); Silver et al. (2017).

While learning-based approaches have improved performance in these domains, they still rely heavily on search. This makes combinatorial reasoning an ideal setting to explore how learned representations can better support search-based methods. Specifically, we investigate how learned representations can induce a metric space that facilitates search, aiming to shift more of the reasoning workload onto representations. Our core hypothesis is that deep learning is not only effective for capturing high-dimensional data patterns but also for learning representations that facilitate geometric reasoning.

Our work builds on a long line of prior methods that integrate learned representations into reinforcement learning and control (Lange & Riedmiller, 2010; Watter et al., 2015; Hafner et al., 2019). We focus on recent work (Eysenbach et al., 2022b; 2024) using temporal contrastive learning (CL) to learn representations where distances correspond to value functions. These methods optimize states from the same trajectory to have similar representations while differentiating states from different trajectories.

However, applying this approach to combinatorial reasoning presents unique challenges, which we illustrate by the Sokoban puzzle (Fig.10). In this task, walls are randomized at the start of each episode, making a naive CL objective trivial: it learns to encode superficial scene features (e.g., wall positions) while ignoring agent and block positions. Although this minimizes loss, the learned representations are useless for decision-making, as they lack temporal structure – within a trajectory they do not indicate which moves advance towards solution (Fig.1). We hypothesize that this issue is more general: combinatorial reasoning tasks naturally decompose into semi-independent subtasks, rendering standard CL objectives ineffective.

To address this problem, we propose Contrastive Representations for Combinatorial Reasoning (CR^2). Our key idea is to modify how temporal contrastive learning selects samples. Instead of a single (start, goal) pair per trajectory, CR^2 samples two such pairs. Intuitively, this provides a "repulsive force" that prevents trivial solutions (Fig. 1). As a result, CR^2 learns representations that reflect the problem's underlying structure.

We validate CR^2 on three challenging combinatorial reasoning tasks: Sokoban, Rubik's Cube, and N-puzzle. Our results show that CR^2 significantly improves search efficiency over standard CL and other baselines. Ablation studies further confirm the importance of our negative sampling strategy in learning high-quality representations.

Our main contributions are the following:

- We show that standard CL fails to capture temporal dependencies in combinatorial problems.
- We introduce Contrastive Representations for Combinatorial Reasoning (CR^2), an algorithm that uses in-trajectory negative sampling to learn high-quality representations for combinatorial reasoning tasks.
- We empirically show that CR^2 improves search efficiency compared to other approaches.

2 RELATED WORK

We build upon recent advances in self-supervised RL and contrastive representation learning, showing that they can be applied successfully to complex combinatorial problems.

Contrastive Learning Contrastive learning has emerged as a widely adopted approach for model pretraining (Jaiswal et al., 2020; Rethmeier & Augenstein, 2023). It facilitates the discovery of rich representations (Chuang et al., 2020; Chen et al., 2020) from unlabeled data that improve learning downstream tasks (Xiao et al., 2021), thereby reducing dependence on human annotations. Importantly, contrastive learning enabled effective learning of large-scale models in fields such as

computer vision (Zhang et al., 2022; Caron et al., 2021), VLMs (Radford et al., 2021; Liu et al., 2024), NLP (Srivastava et al., 2023) and real-world applications including RAG (Gao et al., 2023). The foundational idea of contrastive learning is to learn representations by pulling similar data points, i.e. ones that belong to the same underlying concept, closer together and pushing dissimilar ones further apart in the representation space (Wang & Isola, 2020). This idea is reflected in various contrastive objectives, including Triplet Loss (Hoffer & Ailon, 2014), NCE (Gutmann & Hyvärinen, 2012), or InfoNCE (Sohn, 2016). It has been shown that representations learned in this way demonstrate discriminative power for downstream tasks and exhibit properties such as generalization, robustness (Tian et al., 2020), and transferability (Islam et al., 2021).

Contrastive Representations for Sequential Problems Recently, self-supervised contrastive learning has been also applied to sequential (or temporal) problems, including goal-conditioned RL (Eysenbach et al., 2022b; Venkattaramanujam et al., 2019; Myers et al., 2024), skill-learning algorithms (Park et al., 2023; Zheng et al., 2024b; Eysenbach et al., 2018), or exploration methods (Guo et al., 2022). It has also been tested in symbolic reasoning for simple mathematical problems (Poesia et al., 2021). Most temporal-based contrastive algorithms are based on optimizing InfoNCE objective (Sohn, 2016) to distinguish real future states in the trajectory from random states. Interestingly, Eysenbach et al. (2024) demonstrate that inferring intermediate state representations can be performed by linear interpolation between the initial and final representations. Based on these findings, we hypothesize that such representations might facilitate planning in combinatorial reasoning problems.

Combinatorial Problems Combinatorial environments are characterized by discrete, compact observations that represent exponentially large configuration spaces, often associated with NP-complete problems Karp (1972). Recent RL advancements address these challenges using neural networks to learn efficient strategies, including policy-based heuristics Mazyavkina et al. (2021); Bello et al. (2016), graph neural networks for structural exploitation Cappart et al. (2021); Kool et al. (2019), and imitation learning with expert demonstrations Silver et al. (2016).

Planning in latent space. Planning in complex environments can be made more efficient by leveraging learned state representations that capture the underlying structure of the problem. Techniques such as autoencoders have been employed to reduce the dimensionality of the state space and learn compact world models Ha & Schmidhuber (2018); Hafner et al. (2023). Some approaches focus on learning representations that preserve only the features relevant for planning Schrittwieser et al. (2020); Fang et al. (2022). For robotic applications, latent representations are trained to guide movement and decision-making Ichter & Pavone (2019); Fang et al. (2022). Furthermore, Eysenbach et al. (2022a) frames goal-conditioned planning as a representation learning problem.

3 PRELIMINARIES

We focus on combinatorial problems, which can be formulated as deterministic goal-conditioned controlled Markov processes $(\mathcal{S}, \mathcal{A}, p, p_0, r_g, \gamma)$. In particular, at every timestep t, the agent observes both state $s_t \in \mathcal{S}$ and goal $g \in \mathcal{S}$, and performs action $a_t \in \mathcal{A}$. We assume that the transition function $p : \mathcal{A} \times \mathcal{S} \to \mathcal{S}$ is known and deterministic, while the initial states might differ as they are sampled from the distribution $p_0(s_0)$. We define reward function $r_g = 1$ for $s_t = g$ and $r_g = 0$ otherwise. The objective is to learn goal-conditioned policy $\pi(a \mid s, g)$ that maximize the expected reward:

$$\max_{\pi} \mathbb{E}_{p_0(s_0), p_g(g)} \left[\sum_{t=0}^{\infty} \gamma^t r_g\left(s_t, a_t\right) \right]. \tag{1}$$

Dataset properties. We study an offline learning setup with a dataset of successful yet suboptimal trajectories $\tau_i = ((s_1, a_1), (s_2, a_2), \ldots (g, -))$. We define the distance function $d : \mathcal{S}^2 \to \mathbb{R}$ as follows: for $s_1, s_2 \in \mathcal{S}$ $d(s_1, s_2) = n$ if s_2 is reachable from s_1 in n actions, and there does not exist shorter path between s_1 and s_2. Formally, s_2 is reachable from s_1 if there exist a path $a_1, a_2, \ldots, a_n$, such that $s_2 = p(a_n, p(a_{n-1}, p(\ldots, p(a_1, s_1))))$.

3.1 CONTRASTIVE REINFORCEMENT LEARNING

We employ a contrastive reinforcement learning (CRL) method (Eysenbach et al., 2022b) to train a critic, $f(s, a, g)$, which estimates the correlation between the current state-action pair and future states. The critic consists of two embedding networks: one for state-action pairs, ϕ, and another for goals, ψ. These networks generate representations $\phi(s, a)$ and $\psi(g)$, respectively. The energy function, $f_{\phi,\psi}(s, a, g)$, then measures a form of similarity between these representations that reflects the structure of the task.

For training the critic, we use the InfoNCE objective (Sohn, 2016) as in previous CRL works (Eysenbach et al., 2022b; 2021; Zheng et al., 2023; 2024a; Myers et al., 2024; Bortkiewicz et al., 2024). Specifically, we construct every batch $\mathcal{B}$, by sampling n random trajectories from the dataset. For each trajectory, we select a state-action pair (s_i, a_i) uniformly and draw goal g_i, using a $\mathrm{Geom}(1-\gamma)$ distribution over future states. Negative pairs consist of state-action pairs (s_i, a_i) and goals g_j from different trajectories. Thus, the InfoNCE loss is applied to rows of the batch matrix $\mathcal{B}$ where positive pairs are on the diagonal:

$$\min_{\phi,\psi} \mathbb{E}_{\mathcal{B}} \left[-\sum\nolimits_{i=1}^{|\mathcal{B}|} \log \left(\frac{e^{f_{\phi,\psi}(s_i,a_i,g_i)}}{\sum_{j=1}^{K} e^{f_{\phi,\psi}(s_i,a_i,g_j)}} \right) \right]. \tag{2}$$

Adjusting Contrastive RL for Combinatorial Problems In this work, we adapt the standard CRL setup to address combinatorial problems and learn a proxy of distance function d using the critic f. We use a shared neural network, ϕ, to learn embeddings for both states and goals. The critic output is defined as the squared l_2-norm between these embeddings: $f = ||\phi(s) - \phi(g)||_2$. Because combinatorial problems are deterministic and we are interested in state similarities, we omit actions (a) in the embeddings.

3.2 SEARCH

In reinforcement learning, search-based planning is a widely used approach for solving complex environments Silver et al. (2016); Brown et al. (2020); Yonetani et al. (2021); Orseau et al. (2018). In our study, we focus on the Best-First Search (BestFS) Pearl (1984) planner. BestFS builds the search tree by greedily expanding nodes with the highest heuristic estimates, hence targeting paths that are most likely to lead to the goal. While not ensuring optimality, BestFS provides a simple yet effective strategy for navigating complex search spaces. The pseudocode for BestFS is outlined in Algorithm 2. In our work, we use distances in the latent space as the heuristic, as detailed in Section 3.1.

4 METHOD

The main contribution of this paper is a method for learning representations that facilitate search. We will use an off-the-shelf search algorithm (BestFS) and focus on how learned representations can serve as an effective value function for guiding the search. The rough design for our representations will follow the contrastive approach described in Section 3.1. However, the following example will highlight a key limitation of these representations, which will be addressed in the subsequent sections. Section 4.4 will summarize our complete method, CR^2, which combines this improved representation learning with BestFS search.

4.1 A MOTIVATING EXAMPLE

Consider the Sokoban game, where each problem instance is a maze with a random wall pattern. Directly applying CRL (outlined in 3.1) to this problem does not yield meaningful results. For the CRL objective, positive pairs are sampled from the same trajectory and are close in time, while negative are sampled from different trajectories. CRL samples a batch of pairs. In practice, each batch element will have a different wall pattern due to a huge number of possibilities. Therefore, the objective has a local minimum, where the network learns to tell samples apart by only looking at the wall patterns, completely ignoring the temporal

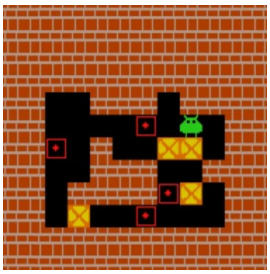

Figure 2: Example Sokoban Board

Algorithm 1 Contrastive Representations for Combinatorial Reasoning (CR^2) algorithm.

Input: Dataset $\mathcal{D}$, Batch Size B, Repetition Factor R.
Output: Batch of pairs for contrastive learning.
T_0 := a sample of $\frac{B}{R}$ trajectories from $\mathcal{D}$
$T := repeat(T_0, R, axis = 0)$
x^i := a state sampled uniformly from T_i for $i \in \{1, \ldots, B\}$
x^i_{t+} := a state sampled from the distribution $p_{t+}(\cdot|x^i)$ from T_i for $i \in \{1, \ldots, B\}$
return (x, x_{t+})

aspect. Indeed, in practice, the standard objective is prone to that issue, as demonstrated in Section 5.3.

4.2 CR^2

We found that there are two types of negatives: *global* that capture the high-level manifold of the environment, and *local* that capture temporal properties. CRL relies only on global negatives, neglecting local structure crucial for planning. As a solution, we propose Contrastive Representations for Combinatorial Reasoning (CR^2), which ensures each batch includes both global and local negatives by sampling $\frac{B}{R}$ trajectories and drawing R pairs per trajectory. Our proposed algorithm solves the problem and is a simple change on top of the usual CRL and hence preserves its theoretical and practical properties. The approach is detailed in Algorithm 1, with alternative strategies explored in Section 5.4. There, we also outline the intuition, for why our approach works well, in contrast to more straightforward methods of incorporating in-trajectory negatives. The impact of the repeat factor R is analyzed in Section 5.6. Our implementation uses $R = 2$.

4.3 Other Combinatorial Reasoning Problems

The example of Sokoban is extreme because the global negatives are trivially separable. This issue also impacts other environments. Consider the example of the Rubik's cube, in which all the trajectories $\tau \in \mathcal{D}$ share the same solved state and all the states are reachable from one another. The last (shuffled) state in the trajectory determines a large portion of the trajectory. When limiting the view to the more shuffled half of the trajectory, using the Hamming distance for deciding whether two states form a positive or negative pair results in 90% accuracy. This shows that the shuffled states in the Rubik's cube are also disjoint to a certain extent. Similar patterns can be found in other domains as well.

4.4 Algorithm Summary

Our complete method (CR^2) works as follows. We take as input an offline dataset of trajectories that solve the given problem, $\mathcal{D} = \{\tau = (s_0, s_1, \cdots)\}$. We train an embedding network ϕ using the InfoNCE loss (Eq. 2.) For a given problem instance, we embed the goal state g getting the representation $z_g = \phi(g)$. For a given state s, we embed it as $z_s = \phi(s)$ and calculate the value $v = ||z_g - z_s||_2$. Values computed in this way (with the embedding network) are used in combination with the BestFS search algorithm 2 to navigate from s to g. We stop the search when we arrive at the solved state or when we exceed a set computational budget.

5 Experiments

5.1 Experimental Setup

Environments. We evaluate the methods across three challenging combinatorial reasoning problems: Sokoban, Rubik's Cube, and N-Puzzle. Since they are known to be NP-hard Demaine et al. (2018); Culberson (1997); Ratner & Warmuth (1986), they are widely used for benchmarking RL algorithms Agostinelli et al. (2019); Racanière et al. (2017); Zawalski et al. (2024). *Sokoban* is a classic grid-based puzzle where an agent must push boxes to designated target locations, avoiding

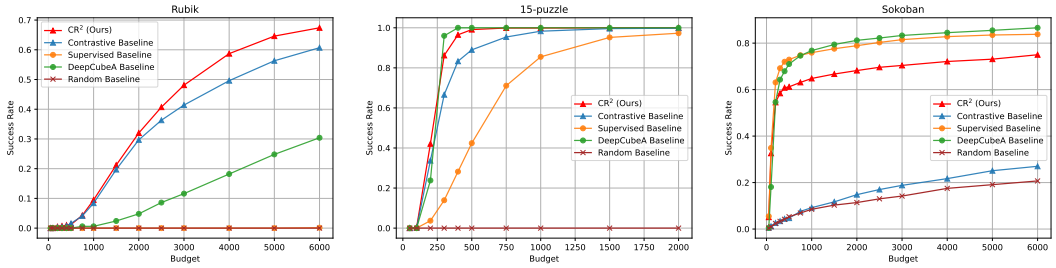

Figure 3: **CR^2 performs well in all the evaluated domains.** Performance of CR^2 compared to baselines.

irreversible states. *Rubik's Cube* is a 3D permutation puzzle where the goal is to align all faces of the cube to a uniform color configuration. *N- puzzle* (or Sliding Tile Puzzle) involves sliding numbered tiles within a 4×4 grid to reach a specific order. Detailed descriptions of our environments can be found in Appendix A.

Baselines. We evaluate CR^2 against three baselines. The *contrastive baseline* follows a standard contrastive reinforcement learning (CRL) approach Eysenbach et al. (2022b), training representations for search without utilizing in-trajectory negatives. The *supervised baseline* directly predicts the distance between two states using a value network trained on demonstration data through imitation. We also compare against *DeepCubeA* Agostinelli et al. (2019), a well-established method for combinatorial reasoning that learns a value function through iterative one-step lookahead updates on increasingly difficult states.

For a fair comparison, both the CRL-based and supervised baselines share the same architecture as CR^2. All methods, including DeepCubeA, use BestFS as the planner. When constructing the search tree for Rubik's Cube, N-Puzzle, and Sokoban, all available actions are considered during node expansion. To measure efficiency, we define the search budget as the number of unique states visited by the planner when solving a given problem instance.

Code to reproduce our results is available online.[1] The training details are specified in Appendix C.

5.2 Main Results

As shown in Figure 3, our CR^2 method demonstrates consistently strong performance across all evaluated domains. It ranks among the top-performing methods in each environment and is strictly the best one in two cases. In contrast, each baseline fails to solve at least one task.

In Sokoban, CR^2 significantly outperforms the CRL baseline. This aligns with our analysis in Section 4.1, where we identified the issue of separable trajectories as a major limitation for naive CL. By addressing this issue, CR^2 achieves substantially better results, confirming the importance of enforcing both global structure and local consistency in learned representations.

In other environments, CR^2 maintains a smaller but consistent advantage over the CRL baseline. This supports our conjecture that the issue of trivial separability is a common challenge. Even in cases where it is less pronounced, addressing it brings benefits.

Furthermore, CR^2 outperforms the supervised baseline and DeepCubeA, particularly in Rubik's Cube. In our evaluations, each method uses the same planning algorithm – BestFS. This suggests that the advantage of CR^2 stems from the structure of its learned representations, which provide more effective guidance for planning compared to the direct value estimation approach used by those baselines. While in Sokoban they achieve higher scores, the difference is small.

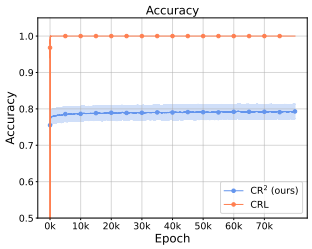

Figure 4: Accuracy of training objectives. CRL quickly acquires near-perfect accuracy, however this is due to relying only on superficial features, like walls.

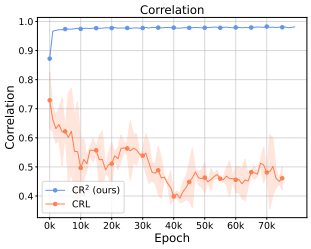

Figure 5: Correlation (Spearman's ρ) between the distance induced by learned embeddings and actual distance. CR^2 quickly acquired a reasonable values, thus allowing for effective planning.

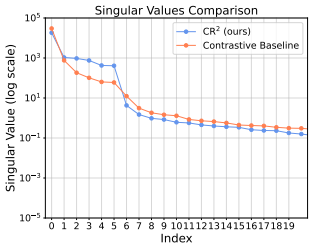

Figure 6: Sorted singular values for Sokoban embeddings. The CR^2 representations have move eigenvectors with high values, hinting more expressive representations.

5.3 Sokoban Case Study

As discussed in Section 4.1, when training demonstrations are easily separable, the CRL baseline collapses to a trivial local minimum, making its learned representations ineffective for planning. Our experiments confirm that prediction. As shown in Figure 1, vanilla CRL produces densely clustered representations, allowing easy trajectory distinction but failing to encode any temporal structure within solutions. This is further reflected in its instant 100% accuracy during training (Figure 4), despite the representations lacking correlation with actual state distances (Figure 5).

In contrast, CR^2 successfully leverages local negatives to enforce temporal consistency in its representations (Figure 1). This augmentation prevents trivial solutions (Figure 4), enabling the model to capture the correct state-space geometry (Figure 5).

This improvement is also evident in the effective rank of learned representations (Figure 6). CR^2 maintains a rank of 6. CRL collapses to just two significant singular values, while the remaining ones are less pronounced. Such a dimensional collapse is often caused by excessive data augmentation or implicit regularization in deep networks Jing et al. (2021).

Finally, this difference directly impacts performance. As shown in Figure 3 (Sokoban), CRL performs only marginally better than random representations, while CR^2 succeeds, confirming the importance of learning structured, temporally consistent representations for effective planning.

5.4 Different Approaches to In-trajectory Negatives

We explore alternative ways to add local negatives. The first approach mirrors the standard addition of hard negatives: given a batch $\mathcal{B} = (x_i, x_{i+})_{i \in \{1..B\}}$, we sample additional negatives, $(x_{i-})_{i \in \{1..B\}}$, and compute the loss as

$$\mathcal{L} = \frac{1}{B} \sum_i \log \left(\frac{\exp\left(f(x_i, x_{i+})\right)}{\sum_{j \neq i} f(x_i, x_{j+}) + f(x_i, x_{i-})} \right),$$

for $f = ||\cdot||_2$.
The second approach modifies the denominator:

$$\mathcal{L} = \frac{1}{B} \sum_i \log \left(\frac{\exp\left(f(x_i, x_{i+})\right)}{\exp(f(x_i, x_{i+})) + \exp(f(x_i, x_{i-}))} \right).$$

We consider three strategies for sourcing in-trajectory negatives: sampling uniformly, selecting the first state, or selecting the last state—excluding the last state for the Rubik's cube since it is the same for all trajectories. All these approaches perform significantly worse than CRL, yielding a success rate below 0.1. We hypothesize that in-trajectory negatives start to dominate the in-batch negatives - the gradient update coming from the in-batch negatives becomes very small early in the training,

[1] Code: https://github.com/combinatorialreasoning/crcr

while the update from the in-trajectory negative remains large. This causes the temporal structure to be lost.

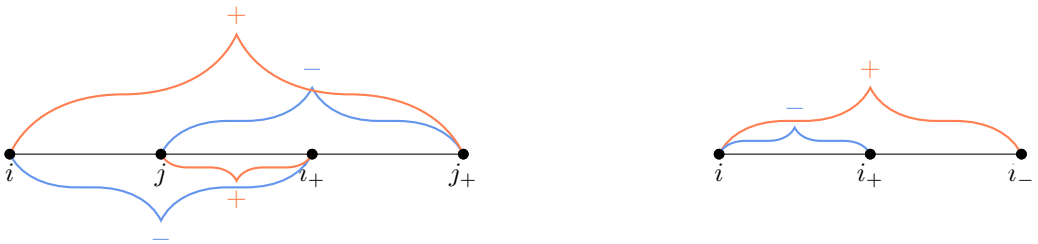

Figure 7: **CR^2 prevents the points in a trajectory from drifting apart.** Embeddings of a single trajectory during a gradient update in CR^2 (on the left) and when using normal in-trajectory negatives (on the right). By $+$, we denote increasing the distance and by $-$, decreasing the distance between embeddings.

How is CR^2 different? Approach 1 is in principle similar to CR^2, so why does Approach 1 fail? The key difference lies in having at least two positive samples from the same trajectory per gradient update. Figure 7 illustrates how our method preserves the trajectory's structure compared to using only in-trajectory negatives. In the standard setting, the loss encourages i and i_+ to move together while pushing i and i_- apart–allowing i_- to drift arbitrarily far. In contrast, CR^2 ensures that j_+ moves away from i, and i_+ moves away from j. while simultaneously pulling i and i_+, as well as j and j_+, together. preventing the structure of the trajectory from being lost. This interplay preserves trajectory coherence, eliminating the need for additional regularization or gradient clipping.

5.5 Is search still necessary?

One of our main questions was whether having good representations allows to use no search, or at least, decrease the amount of search needed. We test the approach, where we always only consider one action, predicted to be the best by our heuristic. We do this until we arrive at the same state for the second time, or exceed the budget of 6000 nodes. Table 1 demonstrates the results of not using search in CR^2, our contrastive baseline supervised baseline. While our approach improves the performance without search, for Rubik's Cube and 15-puzzle it essentially solved no of the boards. In Figure 8 we demonstrate how the performance for the no-search, for the Rubik's cube that is increasingly shuffled. All the methods' performance decreases exponentially, as the number of shuffles is increased. This is expected, as the amount of states reachable within n shuffles follows an exponential trend, for $n \leq 18$. Rokicki et al. (2014) We therefore conclude, that search still is necessary for achieving the optimal performance.

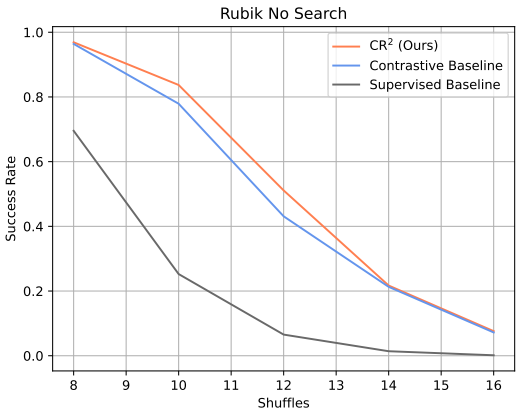

Figure 8: Contrastive representations must be used alongside search.

5.6 Balancing Global and Local Negatives for Effective Representation Learning

As discussed in Section 4.2, CR^2 leverages both global negatives, which capture the overall structure of the environment, and local negatives, which induce the correct temporal consistency of state sequences. Our experiments show that balancing these two components is essential for achieving strong performance.

Table 1: Performance of the supervised baseline, contrastive baseline and CR^2 on Rubik's Cube, 15-puzzle and Sokoban without using search.

Problem	CR^2	Contrastive Baseline	Supervised Baseline
Rubik's Cube	**0.03**	0.02	0.0
15-puzzle	0.0	0.0	0.0
Sokoban	**0.30**	0.0	0.23

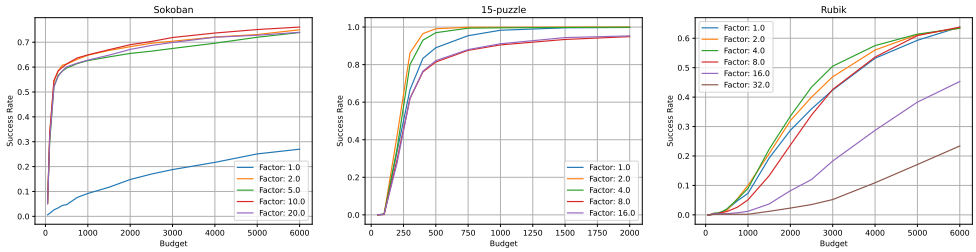

Figure 9: **Influence of the repetition factor depends on the environment type.** Increasing the repetition factor for Sokoban, N-Puzzle, and Rubik's Cube, respectively. Factor 2.0 corresponds to our CR^2, while factor 1.0 corresponds to the CRL baseline.

Figure 9 shows the effect of increasing the fraction of local negatives in the training objective. In case of Sokoban, where trajectories are fully separated, learning progresses only through local negatives, making this parameter irrelevant. In contrast, increasing the fraction of local negatives in other tasks degrades performance. However, relying exclusively on global negatives is also suboptimal across all domains.

In complex problems, effective representations must find a balance between capturing the global structure and maintaining local consistency, which is achieved by CR^2 as shown in our experiments.

6 Limitations and Future Work

Theoretical analysis We empirically demonstrate that in-trajectory negatives improve the performance. In future work, we plan to build a theoretical framework explaining these benefits.

Real-world problems While solving combinatorial reasoning problems is interesting and showcases the potential of our method, considering more impactful problems, such as proving mathematical equations or solving the problem of retrosythesis would give our method the opportunity to be tested in a broader setting.

Multitask Reinforcement Learning Intuitively, in multitask reinforcement learning, CRL should have similar performance issues as it has in combinatorial reasoning problems. Exploring that could have an impact on practical RL algorithms.

7 Conclusions

In our work, we introduced CR^2, an algorithm for learning high-quality representations in combinatorial reasoning tasks. Our analysis revealed a critical limitation of prior approaches: when training demonstrations are separable, their learned representations become trivial and ineffective for planning. CR^2 addresses this by balancing global negatives, which capture overall task structure, with local negatives, which enforce temporal consistency. Experimental results across four challenging domains demonstrate that CR^2 consistently outperforms baselines, highlighting its effectiveness and broad applicability. We share the code for reproducibility.

ACKNOWLEDGEMENTS

PM was supported by National Science Center Poland under the grant agreement 2019/35/O/ST6/03464. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2024/017382.

REFERENCES

Forest Agostinelli, Stephen McAleer, Alexander Shmakov, and Pierre Baldi. Solving the rubik's cube with deep reinforcement learning and search. *Nat. Mach. Intell.*, 1(8):356–363, 2019. doi: 10.1038/S42256-019-0070-Z. URL https://doi.org/10.1038/s42256-019-0070-z.

Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *CoRR*, abs/1611.09940, 2016. URL http://arxiv.org/abs/1611.09940.

Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50. URL https://doi.org/10.1109/TPAMI.2013.50.

Michał Bortkiewicz, Władek Pałucki, Vivek Myers, Tadeusz Dziarmaga, Tomasz Arczewski, Łukasz Kuciński, and Benjamin Eysenbach. Accelerating goal-conditioned rl algorithms and research. *arXiv preprint arXiv:2408.11052*, 2024.

Noam Brown, Anton Bakhtin, Adam Lerer, and Qucheng Gong. Combining deep reinforcement learning and search for imperfect-information games. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c61f571dbd2fb949d3fe5ae1608dd48b-Abstract.html.

Quentin Cappart, Didier Chételat, Elias B. Khalil, Andrea Lodi, Christopher Morris, and Petar Velickovic. Combinatorial optimization and reasoning with graph neural networks. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pp. 4348–4355. ijcai.org, 2021. doi: 10.24963/IJCAI.2021/595. URL https://doi.org/10.24963/ijcai.2021/595.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv: 2104.14294*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

Joseph C. Culberson. Sokoban is pspace-complete. 1997. URL https://api.semanticscholar.org/CorpusID:61114368.

Konrad Czechowski, Tomasz Odrzygózdz, Marek Zbysinski, Michal Zawalski, Krzysztof Olejnik, Yuhuai Wu, Lukasz Kucinski, and Piotr Milos. Subgoal search for complex reasoning tasks. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 624–638, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/05d8cccb5f47e5072f0a05b5f514941a-Abstract.html.

Erik D. Demaine, Sarah Eisenstat, and Mikhail Rudoy. Solving the rubik's cube optimally is np-complete. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi: 10.4230/LIPICS.STACS.2018.24. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.STACS.2018.24.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Dorit Dor and Uri Zwick. SOKOBAN and other motion planning problems. *Comput. Geom.*, 13(4): 215–228, 1999. doi: 10.1016/S0925-7721(99)00017-6. URL https://doi.org/10.1016/S0925-7721(99)00017-6.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and S. Levine. Diversity is all you need: Learning skills without a reward function. *International Conference on Learning Representations*, 2018.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-Learning: Learning to Achieve Goals via Recursive Classification. In *International Conference on Learning Representations*. arXiv, 2021.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022a. URL http://papers.nips.cc/paper_files/paper/2022/hash/e7663e974c4ee7a2b475a4775201ce1f-Abstract-Conference.html.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022b.

Benjamin Eysenbach, Vivek Myers, Ruslan Salakhutdinov, and Sergey Levine. Inference via interpolation: Contrastive representations provably enable planning and inference. 2024.

Kuan Fang, Patrick Yin, Ashvin Nair, Homer Walke, Gengchen Yan, and Sergey Levine. Generalization with lossy affordances: Leveraging broad offline data for learning visuomotor tasks. In Karen Liu, Dana Kulic, and Jeffrey Ichnowski (eds.), *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pp. 106–117. PMLR, 2022. URL https://proceedings.mlr.press/v205/fang23a.html.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv: 2312.10997*, 2023.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html.

Zhaohan Guo, Shantanu Thakoor, Miruna Pislar, Bernardo Ávila Pires, Florent Altché, Corentin Tallec, Alaa Saade, Daniele Calandriello, Jean-Bastien Grill, Yunhao Tang, Michal Valko, Rémi

Munos, Mohammad Gheshlaghi Azar, and Bilal Piot. Byol-explore: Exploration by bootstrapped prediction. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/ced0d3b92bb83b15c43ee32c7f57d867-Abstract-Conference.html.

Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research*, 13(2), 2012.

David Ha and Jürgen Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018. URL http://arxiv.org/abs/1803.10122.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering diverse domains through world models. *CoRR*, abs/2301.04104, 2023. doi: 10.48550/ARXIV.2301.04104. URL https://doi.org/10.48550/arXiv.2301.04104.

Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. *arXiv preprint arXiv: 1412.6622*, 2014.

Haoshuo Huang, Vihan Jain, Harsh Mehta, Alexander Ku, Gabriel Magalhães, Jason Baldridge, and Eugene Ie. Transferable representation learning in vision-and-language navigation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 7403–7412. IEEE, 2019. doi: 10.1109/ICCV.2019.00750. URL https://doi.org/10.1109/ICCV.2019.00750.

Brian Ichter and Marco Pavone. Robot motion planning in learned latent spaces. *IEEE Robotics Autom. Lett.*, 4(3):2407–2414, 2019. doi: 10.1109/LRA.2019.2901898. URL https://doi.org/10.1109/LRA.2019.2901898.

Ashraful Islam, Chun-Fu Richard Chen, Rameswar Panda, Leonid Karlinsky, Richard Radke, and Rogerio Feris. A broad study on the transferability of visual representations with contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8845–8855, 2021.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and F. Makedon. A survey on contrastive self-supervised learning. *TECHNOLOGIES*, 2020. doi: 10.3390/technologies9010002.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *ICLR*, 2021.

Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher (eds.), *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pp. 85–103. Plenum Press, New York, 1972. doi: 10.1007/978-1-4684-2001-2_9. URL https://doi.org/10.1007/978-1-4684-2001-2_9.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ByxBFsRqYm.

Sascha Lange and Martin Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *The 2010 international joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2010.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

Nina Mazyavkina, Sergey Sviridov, Sergei Ivanov, and Evgeny Burnaev. Reinforcement learning for combinatorial optimization: A survey. *Comput. Oper. Res.*, 134:105400, 2021. doi: 10.1016/J.COR.2021.105400. URL https://doi.org/10.1016/j.cor.2021.105400.

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. *International Conference on Machine Learning*, 2024. doi: 10.48550/arXiv.2406.17098.

Laurent Orseau, Levi Lelis, Tor Lattimore, and Theophane Weber. Single-agent policy tree search with guarantees. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 3205–3215, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/52c5189391854c93e8a0e1326e56c14f-Abstract.html.

Seohong Park, Oleh Rybkin, and Sergey Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. *International Conference on Learning Representations*, 2023. doi: 10.48550/arXiv.2310.08887.

Judea Pearl. *Heuristics - intelligent search strategies for computer problem solving*. Addison-Wesley series in artificial intelligence. Addison-Wesley, 1984. ISBN 978-0-201-05594-8.

Gabriel Poesia, WenXin Dong, and Noah Goodman. Contrastive reinforcement learning of symbolic reasoning domains. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 15946–15956. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/859555c74e9afd45ab771c615c1e49a6-Paper.pdf.

Sébastien Racanière, Theophane Weber, David P. Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adrià Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, Razvan Pascanu, Peter W. Battaglia, Demis Hassabis, David Silver, and Daan Wierstra. Imagination-augmented agents for deep reinforcement learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5690–5701, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/9e82757e9a1c12cb710ad680db11f6f1-Abstract.html.

Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, 2021.

Daniel Ratner and Manfred K. Warmuth. Finding a shortest solution for the N × N extension of the 15-puzzle is intractable. In Tom Kehler (ed.), *Proceedings of the 5th National Conference on Artificial Intelligence. Philadelphia, PA, USA, August 11-15, 1986. Volume 1: Science*, pp. 168–172. Morgan Kaufmann, 1986. URL http://www.aaai.org/Library/AAAI/1986/aaai86-027.php.

Nils Rethmeier and Isabelle Augenstein. A primer on contrastive pretraining in language processing: Methods, lessons learned, and perspectives. *ACM Computing Surveys*, 55(10):1–17, 2023.

Tomas Rokicki, Herbert Kociemba, Morley Davidson, and John Dethridge. The diameter of the rubik's cube group is twenty. *SIAM Rev.*, 56(4):645–670, 2014. doi: 10.1137/140973499. URL https://doi.org/10.1137/140973499.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy P. Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nat.*, 588(7839):604–609, 2020. doi: 10.1038/S41586-020-03051-4. URL https://doi.org/10.1038/s41586-020-03051-4.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489, 2016. doi: 10.1038/NATURE16961. URL https://doi.org/10.1038/nature16961.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Kihyuk Sohn. Improved Deep Metric Learning With Multi-Class N-Pair Loss Objective. In *Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, A. Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew K. Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, A. Tabassum, Arul Menezes, Arun Kirubarajan, A. Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakas, B. R. Roberts, B. S. Loe, Barret Zoph, Bartlomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, B. Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, Cèsar Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Christopher Callison-Burch, Christian Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Daniel H Garrette, Dan Hendrycks, D. Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, D. Gilboa, David Dohan, D. Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, E. D. Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, E. Donoway, Ellie Pavlick, Emanuele Rodolà, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, E. Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, F. Siar, Fernando Martínez-Plumed, Francesca Happé, François Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Xinyue Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hanna Hajishirzi, Harsh Mehta, H. Bogar, Henry Shevlin, Hinrich Schütze, H. Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, John Kernion, Jacob Hilton, Jaehoon Lee, J. Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Narain Sohl-Dickstein, Jason Phang, Jason Wei, J. Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Oluwadara Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Jane W Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jorg Frohberg, Jos Rozen, J. Hernández-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Josh Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, K. Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, K. Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras Ochando, Louis-Philippe Morency, Luca Moschella, Luca Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Senel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, M. J. Ramírez-Quintana, M. Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, M. Schubert, Medina Baitemirova, Melody Arnaud, M. McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy,

Michael Starritt, M. Strube, Michal Swedrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mohit Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, T. MukundVarma, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, N. Keskar, Niveditha Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, P. Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, P. Hwang, P. Milkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphael Milliere, Rhythm Garg, Richard Barnes, R. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, R. L. Bras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Samuel Wiseman, Samuel Gruetter, Samuel R. Bowman, S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi S. Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, S. Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima Debnath, Siamak Shakeri, Simon Thormeyer, S. Melzi, Siva Reddy, S. Makini, Soo-Hwan Lee, Spencer Bradley Torene, Sriharsha Hatwar, S. Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T Piantadosi, Stuart M. Shieber, Summer Misherghi, S. Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsunori Hashimoto, Te-Lin Wu, T. Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, T. Kornev, T. Tunduny, Tobias Gerstenberg, T. Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, V. Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, W. Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yufang Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Trans. Mach. Learn. Res.*, 2023.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020.

Srinivas Venkattaramanujam, Eric Crawford, Thang Doan, and Doina Precup. Self-supervised learning of distance functions for goal-conditioned reinforcement learning. *arXiv preprint arXiv: 1907.02998*, 2019.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9929–9939. PMLR, 13-18 Jul 2020. URL `https://proceedings.mlr.press/v119/wang20k.html`.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28, 2015.

Tete Xiao, Xiaolong Wang, Alexei A. Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=CZ8Y3NzuVzO`.

Ryo Yonetani, Tatsunori Taniai, Mohammadamin Barekatain, Mai Nishimura, and Asako Kanezaki. Path planning using neural a* search. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12029–12039. PMLR, 2021. URL `http://proceedings.mlr.press/v139/yonetani21a.html`.

Michal Zawalski, Michal Tyrolski, Konrad Czechowski, Tomasz Odrzygozdz, Damian Stachura, Piotr Piekos, Yuhuai Wu, Łukasz Kucinski, and Piotr Milos. Fast and precise: Adjusting planning horizon with adaptive subgoal search, 2024. URL https://arxiv.org/abs/2206.00702.

Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D. Manning, and Curtis P. Langlotz. Contrastive learning of medical visual representations from paired images and text. In Zachary Lipton, Rajesh Ranganath, Mark Sendak, Michael Sjoding, and Serena Yeung (eds.), *Proceedings of the 7th Machine Learning for Healthcare Conference*, volume 182 of *Proceedings of Machine Learning Research*, pp. 2–25. PMLR, 05-06 Aug 2022. URL https://proceedings.mlr.press/v182/zhang22a.html.

Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive Difference Predictive Coding. In *Twelfth International Conference on Learning Representations*. arXiv, October 2023.

Chongyi Zheng, Benjamin Eysenbach, Homer Walke, Patrick Yin, Kuan Fang, Ruslan Salakhutdinov, and Sergey Levine. Stabilizing Contrastive RL: Techniques for Offline Goal Reaching. In *International Conference on Learning Representations*. arXiv, 2024a.

Chongyi Zheng, Jens Tuyls, Joanne Peng, and Benjamin Eysenbach. Can a misl fly? analysis and ingredients for mutual information skill learning. *arXiv preprint arXiv: 2412.08021*, 2024b.

A ENVIRONMENTS

Sokoban. Sokoban is a well-known puzzle where the player must push boxes onto target locations within a confined grid. Its high combinatorial complexity and PSPACE-hard nature Dor & Zwick (1999) make it a benchmark for both classical planning and deep learning methods. Sokoban challenges algorithms to balance search efficiency and long-term planning. In our experiments, we use 12×12 Sokoban boards with four boxes.

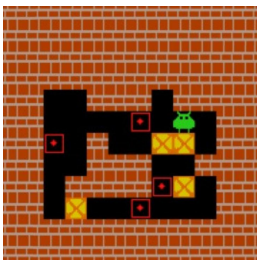

Figure 10: An example instance of Sokoban.

Figure 11: An example instance of Rubik's Cube.

Rubik's Cube. The Rubik's Cube is a 3D combinatorial puzzle with over 4.3×10^{19} possible configurations, making it an iconic testbed for algorithms tackling massive search spaces. Solving the Rubik's Cube requires sophisticated reasoning and planning, as well as the ability to navigate high-dimensional state spaces efficiently. Recent advances in using neural networks for solving this puzzle, such as Agostinelli et al. (2019), highlight the potential of deep learning in handling such computationally challenging tasks.

N-Puzzle. The N-Puzzle is a sliding-tile puzzle with variants like the 8-puzzle (3×3 grid), 15-puzzle (4×4 grid), and 24-puzzle (5×5 grid). The objective is to rearrange tiles into a predefined order by sliding them into an empty space. It serves as a classic benchmark for testing algorithms' planning and search efficiency. The problem's difficulty scales with puzzle size, requiring effective heuristics for solving larger instances.

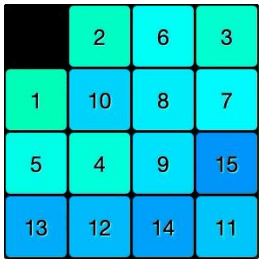

Figure 12: An example instance of N-Puzzle.

B BEST-FIRST SEARCH

Best-First Search greedily prioritizes node expansions with the highest heuristic estimates, aiming for paths that likely lead to the goal. While not ensuring optimality, BestFS provides a simple yet efficient strategy for navigating complex search spaces. The high-level pseudocode for BestFS is outlined in Algorithm 2.

Algorithm 2 Pseudocode for Best-First Search

```
while has nodes to expand do
    Take node N with the highest value
    Select children n_i of N
    Compute values v_i for the children
    Add (n_i, v_i) to the search tree
end while
```

C TRAINING DETAILS

Code to reproduce all our results can be found in the anonymous repository linked in the main text. We trained the supervised baseline, contrastive baseline and CR^2. For Sokoban we use trajectories following Czechowski et al. (2021) and we use 10^5 trajectories for training. For 15-puzzle and Rubik's Cube we generate trajectories by using a policy, performing n random actions, with n equal 150 and 21 correspondingly. In 15-puzzle we delete cycles of length one from the dataset. We work in a setup with an unlimited amount of data, training all the networks for two days. This resulted in seeing around $8 * 10^6$ trajectories for the Rubik's Cube and $7 * 10^6$ for 15-puzzle.