ICLR
International Conference On
Learning Representations



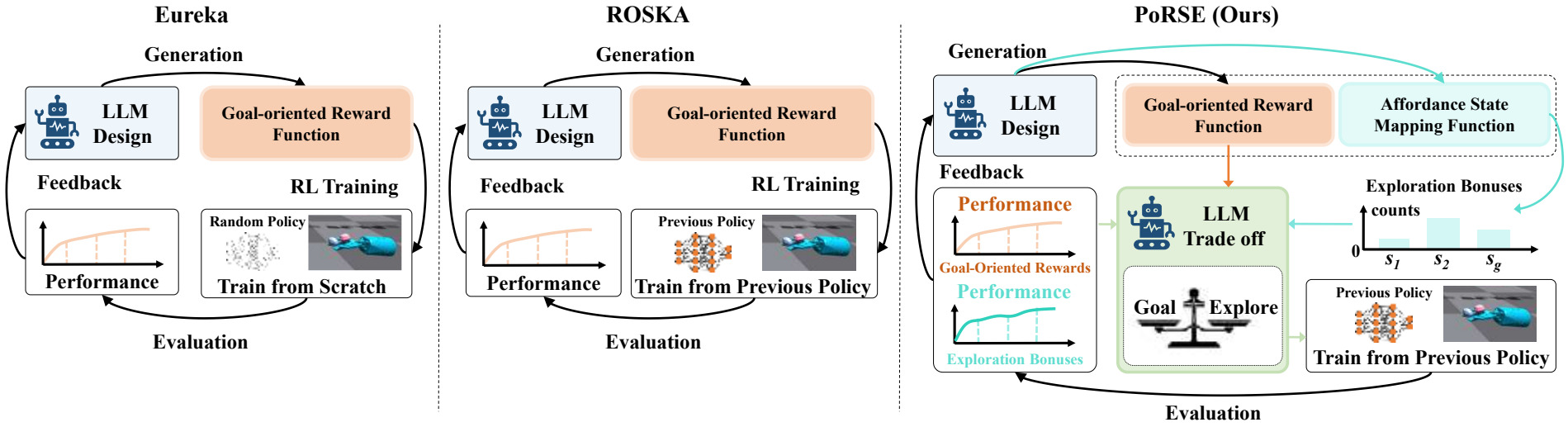
# Master Skill Learning with Policy-Grounded Synergy of LLM-based Reward Shaping and Exploring

**Yanbin Chang, Junfan Lin, Jie Jiang, Runhao Zeng, Changxin Huang, Jianqiang Li**

**The Fourteenth International Conference on Learning Representations, ICLR 2026**

# Background

- ✓ Reward functions are crucial for policy learning in embodied robotics.
- ✓ Designing high-quality reward signals typically requires **extensive domain-specific knowledge**, resulting in significant design costs.
- ✓ LLM-based methods (e.g., Eureka) generate **goal-oriented rewards** but **lack exploration mechanisms**, causing agents to get stuck in local optima.

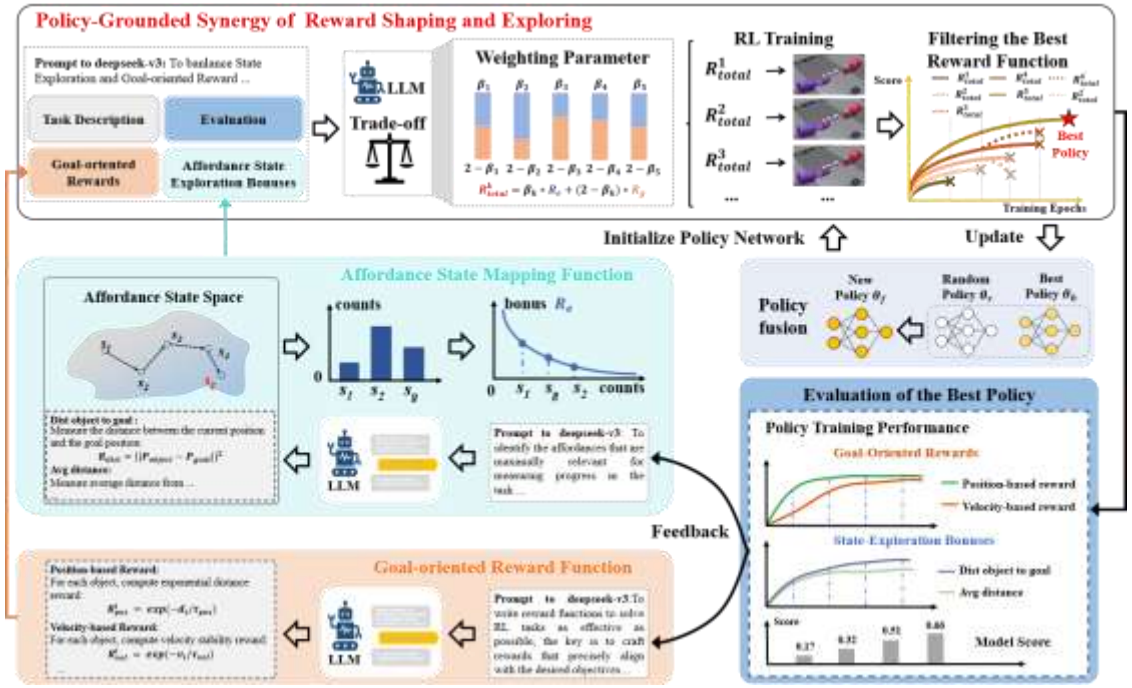



- ➢ Eureka: LLM Design → Goal-oriented Reward → **Train from Scratch**
- ➢ ROSKA: LLM Design → Goal-oriented Reward → **Train from Previous Policy**
- ➢ PoRSE (Ours): LLM Design → **Goal-oriented Reward** + **Affordance State Mapping** → Dynamic Balance → Train from Previous Policy

# Motivation

- ✓ Recently, Large Language Models (LLMs) have demonstrated **extensive domain knowledge and remarkable coding capabilities**.
- ✓ Existing LLM-driven approaches **rely solely on task-specific textual descriptions**, leading to overly goal-oriented rewards while neglecting state exploration.
- ✓ This limitation is particularly problematic in high-DoF robotic tasks (e.g., dexterous manipulation with sparse rewards), **causing agents to get stuck in local optimal**.
- ✓ Traditional exploration bonuses **do not link exploration to task objectives**, resulting in **inefficient exploration of irrelevant states**.
  - ➢ Current LLM-based methods face a critical dilemma: pure goal-oriented rewards **lack exploration** guidance, while **generic exploration** wastes samples on **task-irrelevant** states.
  - ➢ **Manually tuning** the trade-off ratio for each task is **time-consuming and suboptimal.**

# Method



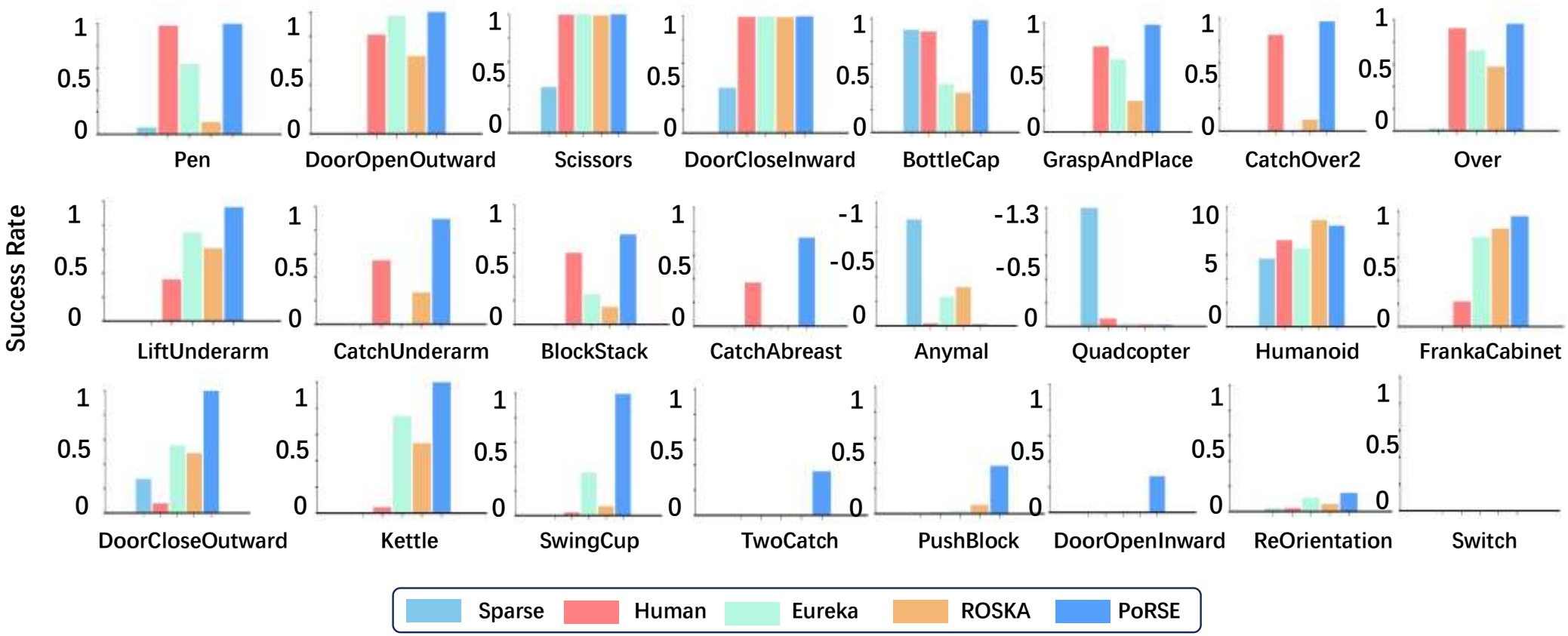

**Key Ideas**

PoRSE decouples robotic skill acquisition into **exploitation-exploration balance** and policy inheritance for **synergistic optimization.**

- ✓ **Exploitation-Exploration Balance:** PoRSE leverages LLMs to auto-set AFS mapping M and iteratively refine $R_g$ and $M$ via policy feedback, maintaining exploration-goal reward pairs $(\beta, R_e, R_g)$ and applying LEF to explore the design space of $\beta$.

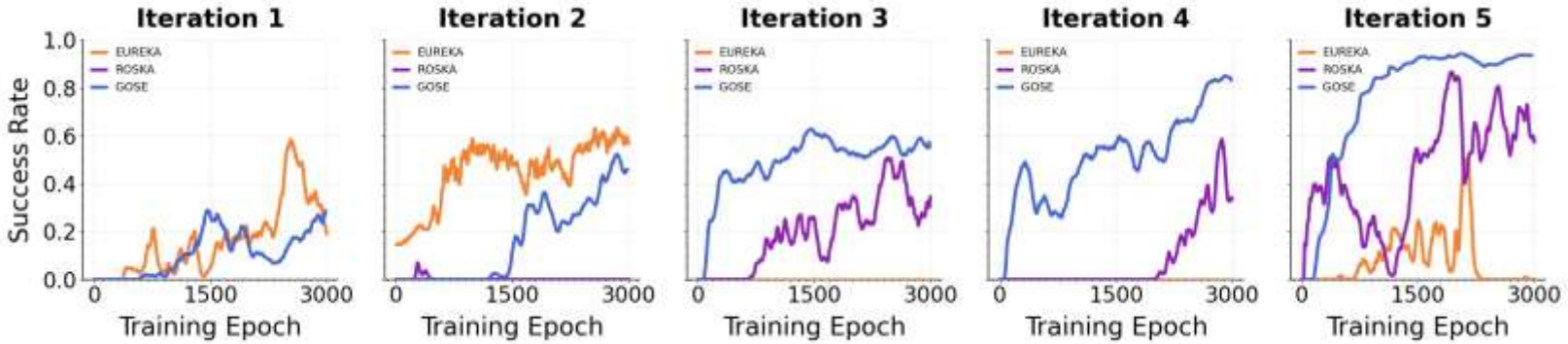
- ✓ **Policy Inheritance:** PoRSE blends $\theta_{best}$ with $\theta_{random}$ via fusion ratio $\alpha$, dynamically adjusting $\alpha$ through LEF to inherit prior knowledge and align policy performance with task completion preferences.

# Experiments

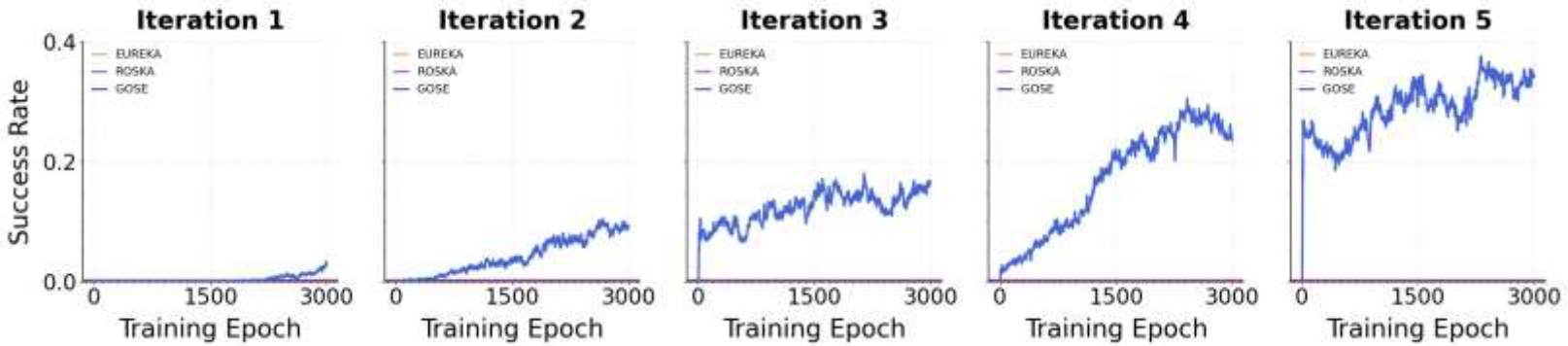



- ✓ **PoRSE achieves the highest or near-highest performance on most benchmarks**, outperforming Sparse, Human, Eureka, and ROSKA in the majority of evaluated tasks.

# Experiments

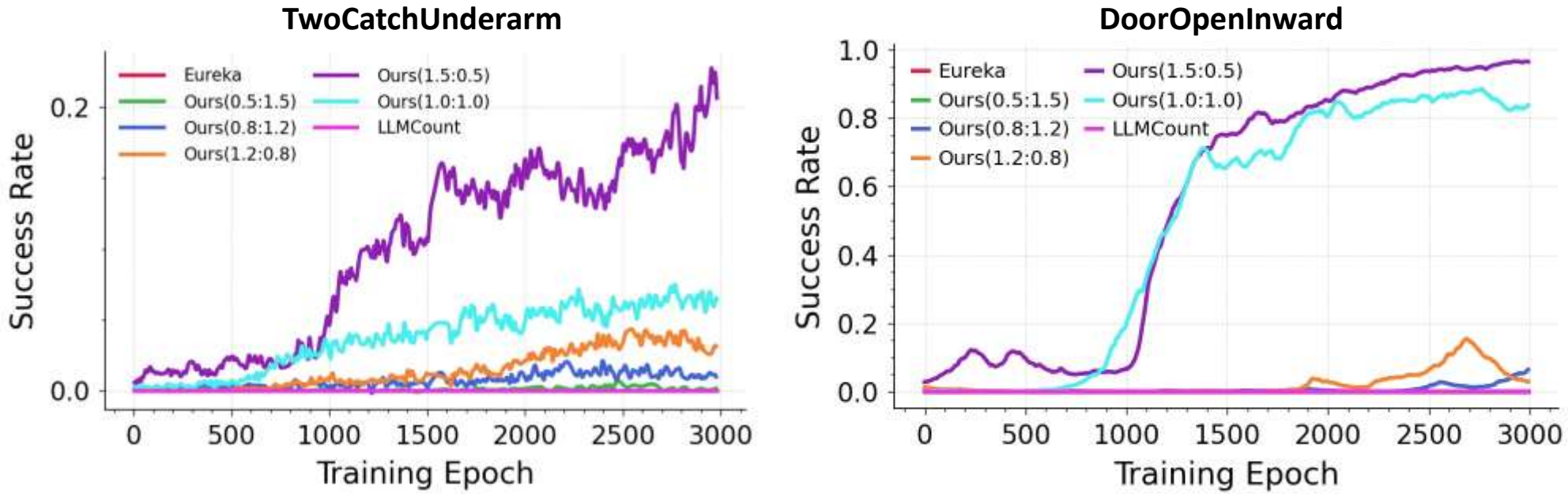








- ✓ On the LiftUnderarm task, PoRSE exhibits a **faster convergence rate** and **achieves higher final success rates** than EUREKA and ROSKA in later iterations, **with its success rate rising rapidly to outperform the baselines.**
- ✓ On the TwoCatch task, PoRSE **achieves success rate improvements earlier across all iterations** and **attains higher final success rates in later iterations** compared to EUREKA and ROSKA.

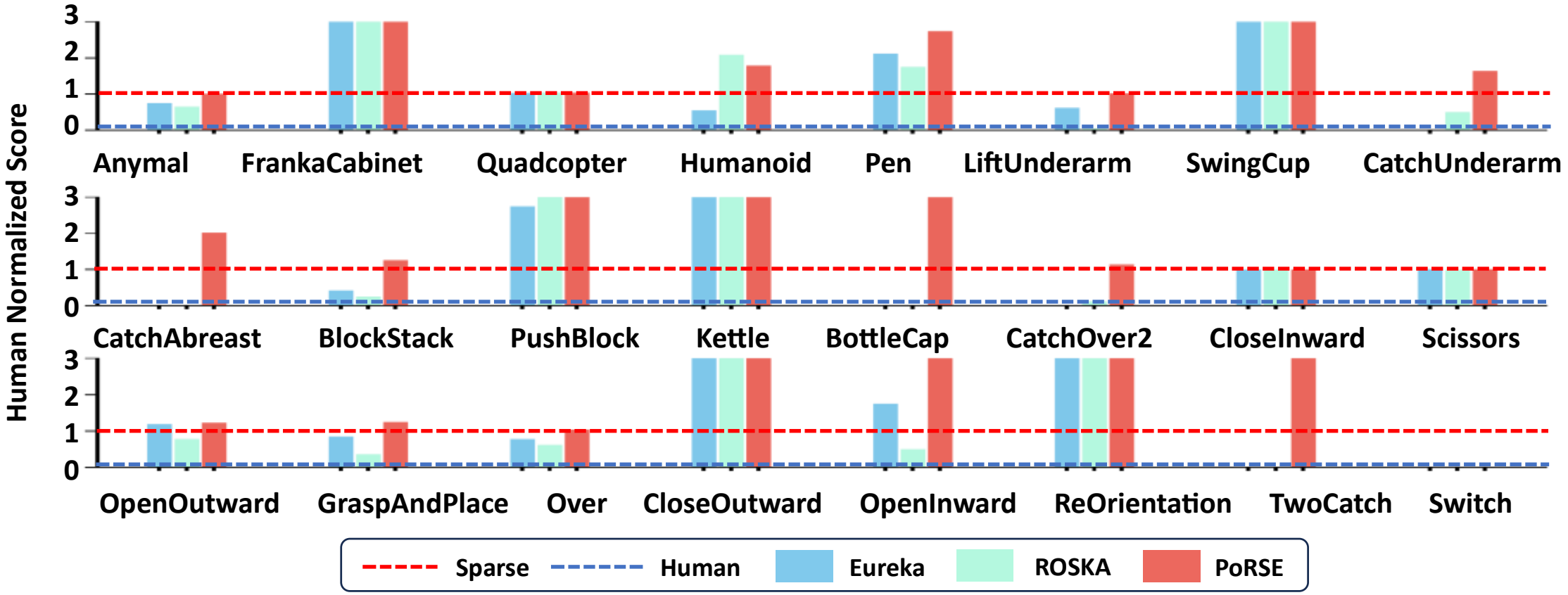
# Experiments




- ✓ For the DoorOpenInward and TwoCatch tasks, when using single-reward configurations (Eureka with Rg-only and LLMCount with Re-only) and training for 3000 epochs, the single-reward approaches failed completely with **0%** success rate.

- ✓ Among the tested PoRSE reward ratios (0.5:1.5, 0.8:1.2, 1.2:0.8, 1.5:0.5, 1.0:1.0), the optimal fusion ratio (1.5:0.5) achieved significant improvements, reaching 97% success rate on DoorOpenInward and 25% on TwoCatch, which outperformed prior methods (~0% success rate).

# Experiments




✓ **PoRSE significantly outperformed other methods in 23 tasks**, achieving the best overall results: it far exceeded human-level HNS in complex fine-grained control tasks, matched human or state-of-the-art performance in simpler tasks, and maintained positive HNS across all evaluated tasks.

# Experiments

| Method | Anymal | Franka | Quadcopter | Humanoid | LiftUnderarm | Pen |
|---|---|---|---|---|---|---|
| LLMCount | -1.226 | 0.706 | -0.092 | 7.737 | 0.649 | 0.412 |
| PoRSE | **-0.012** | **0.957** | **-0.014** | **8.454** | **0.952** | **1.000** |
| **Method** | **SwingCup** | **CatchUnder** | **CatchAbreast** | **BlockStack** | **PushBlock** | **Kettle** |
| LLMCount | 0.993 | 0.000 | 0.238 | 0.140 | 0.127 | **1.000** |
| PoRSE | **0.995** | **0.894** | **0.745** | **0.753** | **0.378** | **1.000** |
| **Method** | **BottleCap** | **CatchOver2** | **CloseIn** | **Scissors** | **OpenOut** | **GraspPlace** |
| LLMCount | 0.985 | 0.387 | **1.000** | **1.000** | 0.998 | 0.906 |
| PoRSE | **0.988** | **0.973** | **1.000** | **1.000** | **1.000** | **0.984** |
| **Method** | **Over** | **CloseOut** | **OpenIn** | **ReOrientation** | **TwoCatch** | **Switch** |
| LLMCount | 0.594 | 0.905 | 0.025 | 0.098 | 0.000 | 0.000 |
| PoRSE | **0.965** | **1.000** | **0.283** | **0.149** | **0.349** | 0.000 |

✓ MTS Comparison of LLMCount and PoRSE methods, PoRSE method **achieved higher MTS on 23 tasks**.

# Experiments

| Method | Anymal | | Quadcopter | | Franka | |
|---|---|---|---|---|---|---|
| | MTS | Gap | MTS | Gap | MTS | Gap |
| PoRSE w/o $\mathbf{R_g}$ | −0.128 | ↓ 0.116 | −0.040 | ↓ 0.026 | 0.883 | ↓ 0.074 |
| PoRSE w/o $\mathbf{R_e}$ | −0.097 | ↓ 0.085 | −0.038 | ↓ 0.024 | 0.912 | ↓ 0.045 |
| PoRSE w/o $\boldsymbol{\theta_{\text{fusion}}}$ | −0.346 | ↓ 0.334 | −0.034 | ↓ 0.020 | 0.671 | ↓ 0.286 |
| PoRSE w/o $\mathbf{R_{\text{ratio}}}$ | −0.066 | ↓ 0.054 | −0.019 | ↓ 0.005 | 0.946 | ↓ 0.011 |
| PoRSE w/o $\boldsymbol{\theta_{\text{ratio}}}$ | −0.020 | ↓ 0.008 | −0.015 | ↓ 0.001 | 0.940 | ↓ 0.017 |
| PoRSE | −0.012 | | −0.014 | | 0.957 | |

| Method | BlockStack | | PushBlock | | TwoCatch | |
|---|---|---|---|---|---|---|
| | MTS | Gap | MTS | Gap | MTS | Gap |
| PoRSE w/o $\mathbf{R_g}$ | 0.328 | ↓ 0.425 | 0.295 | ↓ 0.023 | 0.190 | ↓ 0.159 |
| PoRSE w/o $\mathbf{R_e}$ | 0.393 | ↓ 0.360 | 0.306 | ↓ 0.012 | 0.193 | ↓ 0.156 |
| PoRSE w/o $\boldsymbol{\theta_{\text{fusion}}}$ | 0.603 | ↓ 0.150 | 0.236 | ↓ 0.082 | 0.017 | ↓ 0.332 |
| PoRSE w/o $\mathbf{R_{\text{ratio}}}$ | 0.296 | ↓ 0.457 | 0.243 | ↓ 0.075 | 0.297 | ↓ 0.052 |
| PoRSE w/o $\boldsymbol{\theta_{\text{ratio}}}$ | 0.590 | ↓ 0.163 | 0.309 | ↓ 0.009 | 0.276 | ↓ 0.073 |
| PoRSE | 0.753 | | 0.318 | | 0.349 | |

- ✓ Removing θfusion caused the largest performance drops (Anymal ↓ 2783.33%, TwoCatch ↓ 95.13%).
- ✓ Eliminating Rg or Re led to massive declines on Anymal (Rg ↓ 966.67%, Re ↓ 708.33%).
- ✓ Removing Rratio or θratio brought moderate drops (BlockStack ↓ 60.70%, PushBlock ↓ 23.58%).

# Experiments

| Task | BlockStack | PushBlock | LiftUnderarm | CatchUnderarm |
|---|---|---|---|---|
| Sparse | 0.000 ± 0.001 | 0.003 ± 0.004 | 0.000 ± 0.000 | 0.000 ± 0.000 |
| Human | 0.600 ± 0.229 | 0.011 ± 0.003 | 0.348 ± 0.140 | 0.544 ± 0.250 |
| Eureka | 0.254 ± 0.119 | 0.025 ± 0.008 | 0.739 ± 0.166 | 0.000 ± 0.000 |
| Roska | 0.148 ± 0.154 | 0.069 ± 0.049 | 0.608 ± 0.211 | 0.271 ± 0.370 |
| PoRSE-GPT-4o-min | 0.618 ± 0.165 | 0.360 ± 0.051 | 0.869 ± 0.07 | 0.907 ± 0.015 |
| PoRSE | **0.753 ± 0.210** | **0.378 ± 0.022** | **0.952 ± 0.015** | **0.894 ± 0.038** |

- ✓ On BlockStack, PushBlock, LiftUnderarm, and CatchUnderarm, **PoRSE-GPT-4o-mini outperforms all baselines** while remaining below the full PoRSE.

| Task | BlockStack | PushBlock | LiftUnderarm |
|---|---|---|---|
| Sparse | 0.000 ± 0.001 | 0.003 ± 0.004 | 0.000 ± 0.000 |
| Human | 0.600 ± 0.229 | 0.011 ± 0.003 | 0.348 ± 0.140 |
| Eureka | 0.254 ± 0.119 | 0.025 ± 0.008 | 0.739 ± 0.166 |
| Roska | 0.148 ± 0.154 | 0.069 ± 0.049 | 0.608 ± 0.211 |
| PoRSE-AFS-Random | 0.680 ± 0.080 | 0.324 ± 0.034 | 0.802 ± 0.029 |
| PoRSE | **0.753 ± 0.210** | **0.378 ± 0.022** | **0.952 ± 0.015** |

- ✓ Even with **randomly assembled invalid AFS dimensions**, the framework **still outperforms all baseline methods** on BlockStack, PushBlock, and LiftUnderarm, while remaining below the full PoRSE.

# Experiments

| Method | FrankaCabinet | OpenInward | OpenOutward |
|---|---|---|---|
| PoRSE-Prompt | 0.951 ± 0.031 | 0.264 ± 0.044 | 1.000 ± 0.000 |
| PoRSE | **0.957 ± 0.049** | **0.283 ± 0.386** | **1.000 ± 0.000** |

| Method | BlockStack | PushBlock | LiftUnderarm |
|---|---|---|---|
| PoRSE (1000 bins) | 0.753 ± 0.210 | **0.378 ± 0.022** | **0.952 ± 0.015** |
| PoRSE (2000 bins) | 0.708 ± 0.056 | 0.335 ± 0.103 | 0.950 ± 0.023 |
| PoRSE (3000 bins) | **0.763 ± 0.266** | 0.321 ± 0.027 | 0.922 ± 0.079 |

| Method | BlockStack | PushBlock | LiftUnderarm |
|---|---|---|---|
| PoRSE (Normalized-to-1) | 0.744 ± 0.201 | 0.353 ± 0.077 | **0.957 ± 0.020** |
| PoRSE | **0.753 ± 0.210** | **0.378 ± 0.022** | 0.952 ± 0.015 |

- ✓ **No significant MTS difference** from the original PoRSE, even with redundant AFS dimensions introduced by guided prompts.
- ✓ **No significant MTS differences** across 1000, 2000, and 3000 bins on BlockStack, PushBlock, and LiftUnderarm.
- ✓ Changing the reward coefficient sum from 2 to 1 results in **minimal MTS differences**.

# Conclusion

- PoRSE proposes a unified framework integrating **LLM-generated goal-reward and dynamic curiosity-driven bonus**, enabling **task-aligned efficient exploration** without manual reward tuning.
- PoRSE's IPG process optimizes **reward configurations** via **real-time policy feedback**, adopting fast LLM-aided methods instead of cumbersome Bayesian optimization.
- Experiments on 24 robotic tasks show **PoRSE outperforms SOTA LLM-based reward-design methods and breaks through previously unsolved challenges**.
- Unifying **reward shaping, task-relevant exploration, and policy refinement** in a self-reinforcing cycle, PoRSE sets a new paradigm for **autonomous robotic skill acquisition**.
- PoRSE's synergistic optimization of goal-oriented rewards, AFS-guided exploration, and policy inheritance ensures robust cross-task performance, **effective in sparse reward and high-dimensional action spaces**.