LLM-Assisted Semantically Diverse Teammate Generation for Efficient Multi-agent Coordination

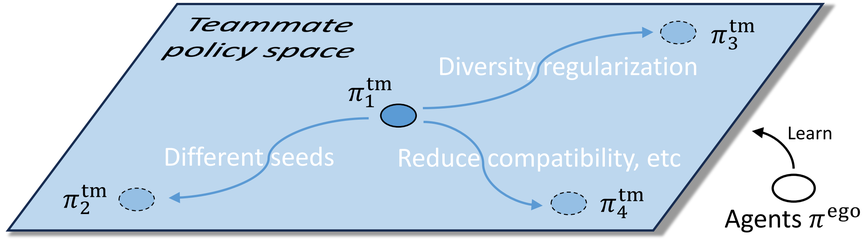
Lihe Li[1,2], Lei Yuan[1,2,3], Pengsen Liu[1,2], Tao Jiang[1,2,3], Yang Yu[1,2,3]

[1]National Key Laboratory for Novel Software Technology, Nanjing University; [2]School of Artificial Intelligence, Nanjing University; [3]Polixir Technologies

Introduction

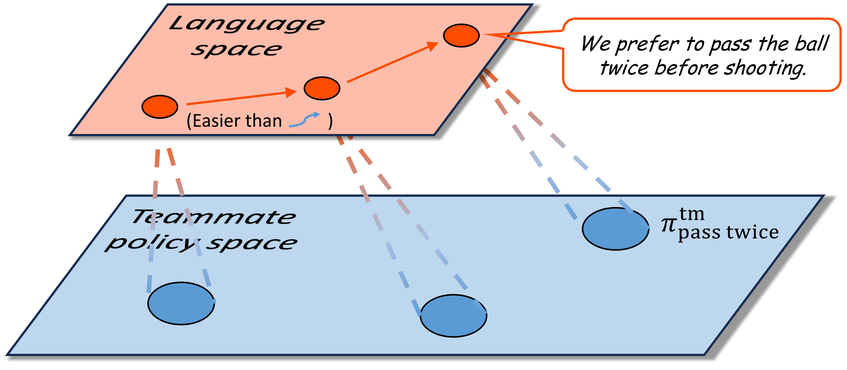
Training with diverse teammates is the key for learning generalizable agents.

Classic methods mainly explore on **policy-level** to generate teammates.



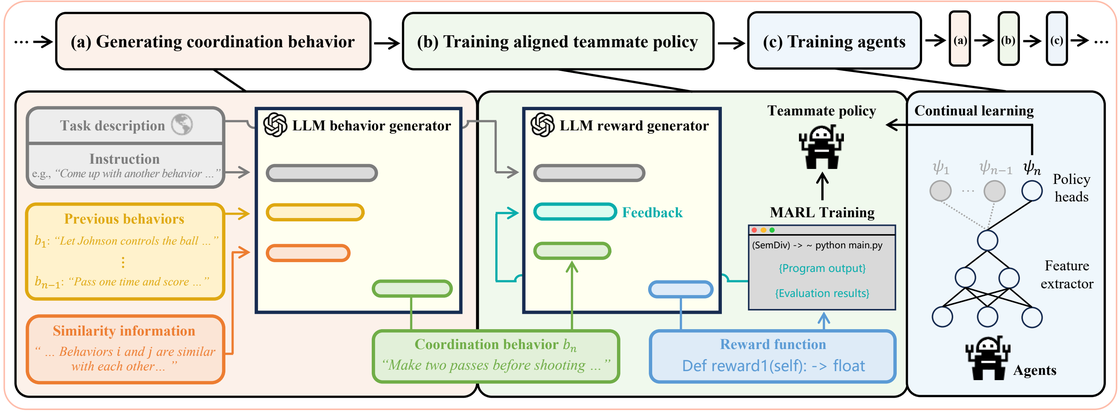
- **Low efficiency**: The more complex the space is, the harder to explore.
- **Lack semantic information**: We don't know teammates' behaviors.

*What if we explore on a higher **sematic-level**, first finding novel coordination-behaviors, and then generating the corresponding teammates?*



Method

We propose LLM-Assisted **Sem**antically **Div**erse Teammate Generation (**SEMDIV**), which *iteratively*:



(a) utilizes LLMs to propose a novel coordination behavior in natural language (the → step),

(b) translates this behavior into reward function code, and trains a grounded MARL teammate policy π^{tm} (the step),

(c) trains ego agents π^{ego} to coordinate with this teammate.

Finally, SEMDIV obtains a set of diverse grounded teammates + strong and adaptable coordination agents.

Experiments

Overall performance with unseen teammates (R1: Return, R2: Run teammates' behavior %)

Methods	LBF		PP		SMACv2		GRF		Average	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
Oracle	1.00	1.00	0.91	0.90	0.94	0.93	0.95	0.95	0.95	0.95
SEMDIV	**0.90** ±0.05	**0.90** ±0.05	**0.72** ±0.03	**0.54** ±0.10	**0.65** ±0.02	**0.64** ±0.02	**0.67** ±0.08	**0.62** ±0.07	**0.74**	**0.68**
SEMDIV-Dist	0.45 ±0.14	0.45 ±0.14	0.51 ±0.03	0.28 ±0.05	0.24 ±0.08	0.23 ±0.08	0.47 ±0.20	0.37 ±0.16	0.42	0.33
SEMDIV-R1	0.91 ±0.04	0.91 ±0.04	0.76 ±0.01	0.53 ±0.04	0.70 ±0.00	0.69 ±0.01	0.88 ±0.06	0.62 ±0.08	0.81	0.69
SEMDIV-R2	0.91 ±0.04	0.91 ±0.04	0.74 ±0.01	0.58 ±0.06	0.70 ±0.00	0.69 ±0.01	0.78 ±0.08	0.73 ±0.05	0.78	0.73
Macop-R1	0.82 ±0.10	0.81 ±0.11	0.58 ±0.02	0.23 ±0.00	0.48 ±0.03	0.45 ±0.03	0.59 ±0.15	0.44 ±0.04	0.62	0.48
Macop-R2	0.82 ±0.10	0.81 ±0.11	0.54 ±0.01	0.25 ±0.00	0.47 ±0.03	0.45 ±0.03	0.56 ±0.15	0.45 ±0.03	0.60	0.49
SEMDIV-PBT	0.64 ±0.02	0.64 ±0.02	0.70 ±0.01	0.31 ±0.01	0.61 ±0.01	0.61 ±0.01	0.57 ±0.30	0.39 ±0.12	0.63	0.49
Macop-PBT	0.61 ±0.00	0.60 ±0.02	**0.72** ±0.03	0.33 ±0.03	0.56 ±0.04	0.54 ±0.03	0.49 ±0.24	0.35 ±0.10	0.60	0.46
FCP	0.46 ±0.22	0.43 ±0.20	0.57 ±0.23	0.21 ±0.15	0.40 ±0.05	0.37 ±0.06	0.50 ±0.25	0.36 ±0.12	0.48	0.34
MEP	0.57 ±0.08	0.56 ±0.08	0.70 ±0.01	0.31 ±0.01	0.55 ±0.04	0.47 ±0.02	0.50 ±0.26	0.35 ±0.14	0.58	0.42
LIPO	0.54 ±0.00	0.51 ±0.02	0.69 ±0.02	0.31 ±0.01	0.45 ±0.10	0.38 ±0.06	0.51 ±0.25	0.37 ±0.12	0.55	0.39
LLM-Agent	0.88 ±0.05	0.88 ±0.05	0.71 ±0.09	0.53 ±0.08	0.35 ±0.10	0.35 ±0.10	0.14 ±0.09	0.12 ±0.09	0.52	0.47

A teammate example in Google Research Football (GRF):

LLM-output coordination behavior:
We prefer to pass the ball twice before shooting.

```
def reward1(self) -> float:
    if self.score:
        # Check if there are two passes in the history
        if len(self.pass_history) == 2:
            # Check if the passes are between the two players
            # ... (Codes omitted for clarity)
            # Large reward to reinforce the desired behavior
            return 100.0
    # Default return, no extra reward in other cases
    return 0.0
```

SEMDIV discovers behaviors that baselines cannot!

Paper

Home page