A PROOF OF MAIN RESULTS

A.1 PROOF OF THEOREM 1

Theorem 1. *Let the joint policy be the product of the individual policy* $\{\pi_i\}_{i=1}^k$*, where* π_i *with respect to the individual flow function* $F_i(o_i, a_i)$*, i.e.,*

$$\pi_i(a_i|o_i) = \frac{F_i(o_i, a_i)}{F_i(o_i)}, \ \forall i = 1, \cdots, k. \tag{16}$$

Assume that the individual flow $F_i(o_i, a_i)$ *satisfies the condition in Definition 2. Define a flow function* $\hat{F}$*, if all agents generate trajectories using independent policies* $\pi_i, \ i = 1, ..., k$ *and the matching conditions*

$$\forall s' > s_0, \ \hat{F}\left(s'\right) = \sum_{s \in \mathcal{P}(s')} \hat{F}\left(s \rightarrow s'\right) \text{ and} \, \forall s' < s_f, \ \hat{F}\left(s'\right) = \sum_{s'' \in \mathcal{C}(s')} \hat{F}\left(s' \rightarrow s''\right) \tag{17}$$

are satisfied. Then, we have:
1) $\pi(s_f) \propto R(s_f)$*;*
2) $\hat{F}$ *uniquely defines a Markovian flow* F *matching* $\hat{F}$ *such that*

$$F(\tau) = \frac{\prod_{t=1}^{n+1} \hat{F}\left(s_{t-1} \rightarrow s_t\right)}{\prod_{t=1}^{n} \hat{F}\left(s_t\right)}. \tag{18}$$

Proof: We first prove the part 1). Since

$$F(s_t, \boldsymbol{a}_t) = \prod_i F_i(o_t^i, a_t^i),$$

then we have the global state flow as

$$F(s_t) = \sum_{a_t \in \mathcal{A}} F(s_t, \boldsymbol{a}_t) = \sum_{\boldsymbol{a}_t \in \mathcal{A}} \prod_i F_i(o_t^i, a_t^i). \tag{19}$$

According to the flow definitions, the observation flow $F_i(o_t^i)$ and individual observation flows have the relationship:

$$F_i(o_t^i) = \sum_{a_t^i \in \mathcal{A}^i} F_i(o_t^i, a_t^i). \tag{20}$$

Hence, we have

$$\begin{aligned}
\prod_{i=1}^k F_i(o_t^i) &= \prod_{i=1}^k \left\{ \sum_{a_t^i \in \mathcal{A}^i} F_i(o_t^i, a_t^i) \right\} && (21)\\
&= \sum_{a_t^1 \in \mathcal{A}^1} F_i(o_t^1, a_t^1) \cdots \sum_{a_t^k \in \mathcal{A}^k} F_i(o_t^k, a_t^k) && (22)\\
&= \sum_{a_t^1, \cdots, a_t^k \in \mathcal{A}^1 \times \cdots \times \mathcal{A}^k} F_i(o_t^1, a_t^1) \cdots F_i(o_t^k, a_t^k) && (23)\\
&= \sum_{a_t \in \mathcal{A}} \prod_{i=1}^k F_i(o_t^i, a_t^i), && (24)
\end{aligned}$$

yielding $F(s_t) = \prod_i F_i(o_t^i)$. Therefore, the joint policy

$$\begin{aligned}
\pi(\boldsymbol{a}|s) &= \frac{F(s_t, \boldsymbol{a}_t)}{F(s_t)} = \frac{\prod_i F_i(o_t^i, a_t^i)}{F(s_t)} \\
&= \frac{\prod_i F_i(o_t^i, a_t^i)}{\prod_i F_i(o_t^i)} = \prod_i \pi_i(a_i|o_i).
\end{aligned} \tag{25}$$

Equation 25 indicates that if the conditions in Definition 2 is satisfied, we can establish the consistency of joint and individual policies. Based on Lemma 1, we can conclude that the reward of the generated state satisfies $\pi(s_f) \propto R(s_f)$ using the individual policy $\pi_i(a_i|o_i)$ of each agent.

Next, we prove the part 2). We first prove the necessity part. According to Definition 2 and Bengio et al. (2021b) we have

$$
\begin{aligned}
F(s') &= \prod_i F_i(o^{i,\prime}) = \prod_i \sum_{o^i \in \mathcal{P}(o^{i,\prime})} F_i(o^i \to o^{i,\prime}) = \sum_{\boldsymbol{o} \in \mathcal{P}(\boldsymbol{o}')} \prod_i F_i(o^i \to o^{i,\prime}), \\
F(s') &= \prod_i F_i(o^{i,\prime}) = \prod_i \sum_{o^{i,\prime\prime} \in \mathcal{C}(o^{i,\prime})} F_i(o^{i,\prime} \to o^{i,\prime\prime}) = \sum_{\boldsymbol{o}'' \in \mathcal{C}(\boldsymbol{o}')} \prod_i F_i(o^{i,\prime} \to o^{i,\prime\prime}).
\end{aligned}
$$

Then we prove the sufficiency part. We first present Lemma 3, which shows that

$$
\sum_{\tau \in \mathcal{T}_{0,s}} P_B(\tau) = \sum_{\tau \in \mathcal{T}_{0,s}} \prod_{s_t \to s_{t+1} \in \tau} P_B(s_t|s_{t+1}) = 1.
$$

Lemma 3 (Independent Transition Probability) *Define the independent forward and backward transition respectively as*

$$
P_F\left(o_{t+1}^i|o_t^i\right) := P_i\left(o_t^i \to o_{t+1}^i|o_t^i\right) = \frac{F_i\left(o_t^i \to o_{t+1}^i\right)}{F_i\left(o_t^i\right)}, \tag{26}
$$

and

$$
P_B\left(o_t^i|o_{t+1}^i\right) := P_i\left(o_{t+1}^i \to o_t^i|o_{t+1}^i\right) = \frac{F_i\left(o_{t+1}^i \to o_t^i\right)}{F_i\left(o_{t+1}^i\right)}. \tag{27}
$$

Then we have

$$
\begin{aligned}
&\sum_{\tau \in \mathcal{T}_{s,f}} P_F(\tau) = 1, \forall s \in \mathcal{S} \backslash \{s_f\}, \\
&\sum_{\tau \in \mathcal{T}_{0,s}} P_B(\tau) = 1, \forall s \in \mathcal{S} \backslash \{s_0\},
\end{aligned} \tag{28}
$$

where $\mathcal{T}_{s,f}$ is the set of trajectories starting in s and ending in s_f and $\mathcal{T}_{0,s}$ is the set of trajectories starting in s_0 and ending in s.

Define $\hat{Z} = \hat{F}(s_0)$ as the partition function and $\hat{P}_F$ as the forward probability function. Then, according to Proposition 18 in Bengio et al. (2021b), we have there exists a unique Markovian flow F with forward transition probability function $P_F = \hat{P}_F$ and partition function Z, and such that

$$
F(\tau) = \hat{Z} \prod_{t=1}^{n+1} \hat{P}_F(s_t|s_{t-1}) = \frac{\prod_{t=1}^{n+1} \hat{F}(s_{t-1} \to s_t)}{\prod_{t=1}^{n} \hat{F}(s_t)}, \tag{29}
$$

where $s_{n+1} = s_f$. Thus, we have for $s' \neq s_0$:

$$
\begin{aligned}
F(s') &= \hat{Z} \sum_{\tau \in \mathcal{T}_{0,s'}} \prod_{(s_t \to s_{t+1}) \in \tau} \hat{P}_F(s_{t+1}|s_t) \\
&= \hat{Z} \frac{\hat{F}(s')}{\hat{F}(s_0)} \sum_{\tau \in \mathcal{T}_{0,s'}} \prod_{(s_t \to s_{t+1}) \in \tau} \hat{P}_B(s_t|s_{t+1}) = \hat{F}(s').
\end{aligned} \tag{30}
$$

Combining equation 30 with $P_F = \hat{P}_F$, we have $\forall s \to s' \in \mathcal{A}, F(s \to s')$. Finally, for any Markovian flow F' matching $\hat{F}$ on states and edges, we have $F'(\tau) = F(\tau)$ according to Proposition 16 in Bengio et al. (2021b), which shows the uniqueness property. Then we complete the proof.

A.2 PROOF OF LEMMA 2

Lemma 2. *Suppose Assumption 1 holds and the environment has a tree structure, based on the IGC and IGM conditions we have:*
1) $Q^{\mu}_{tot}(s, \boldsymbol{a}) = F(s, \boldsymbol{a})f(s)$*;*
2) $(\arg\max_{a_i} Q_i(o_i, a_i))_{i=1}^{k} = (\arg\max_{a_i} F_i(o_i, a_i))_{i=1}^{k}$.

Proof: The proof is an extension of that of Proposition 4 in Bengio et al. (2021a). For any $(s, \boldsymbol{a})$ satisfies $s_f = T(s, \boldsymbol{a})$, we have $Q^{\mu}_{\text{tot}}(s, \boldsymbol{a}) = R(s_f)f(s)$ and $F(s, \boldsymbol{a}) = R(s_f)$. Therefore, we have $Q^{\mu}_{\text{tot}}(s, \boldsymbol{a}) = F(s, \boldsymbol{a})f(s)$. Then, for each non-final node s', based on the action-value function in terms of the action-value at the next step, we have by induction:

$$\begin{aligned} Q^{\mu}_{\text{tot}}(s, \boldsymbol{a}) &= \hat{R}(s') + \mu(\boldsymbol{a}|s') \sum_{\boldsymbol{a}' \in \mathcal{A}(s')} Q^{\mu}_{\text{tot}}(s', \boldsymbol{a}'; \hat{R}) \\ &\overset{(a)}{=} 0 + \mu(\boldsymbol{a}|s') \sum_{\boldsymbol{a}' \in \mathcal{A}(s')} F(s', \boldsymbol{a}'; R) f(s'), \end{aligned} \tag{31}$$

where $\hat{R}(s')$ is the reward of $Q^{\mu}_{\text{tot}}(s, \boldsymbol{a})$ and (a) is due to that $\hat{R}(s') = 0$ if s' is not a final state. Since the environment has a tree structure, we have

$$F(s, \boldsymbol{a}) = \sum_{\boldsymbol{a}' \in \mathcal{A}(s')} F(s', \boldsymbol{a}'), \tag{32}$$

which yields

$$Q^{\mu}_{\text{tot}}(s, \boldsymbol{a}) = \mu(\boldsymbol{a}|s')F(s, \boldsymbol{a})f(s') = \mu(\boldsymbol{a}|s')F(s, \boldsymbol{a})f(s)\frac{1}{\mu(\boldsymbol{a}|s')} = F(s, \boldsymbol{a})f(s).$$

According to the IGC condition we have $F(s_t, \boldsymbol{a}_t) = \prod_i F_i(o^i_t, a^i_t)$, yielding

$$\begin{aligned} \arg\max_{\boldsymbol{a}} Q_{\text{tot}}(s, \boldsymbol{a}) &\overset{(a)}{=} \arg\max_{\boldsymbol{a}} \log F(s, \boldsymbol{a})f(s) \\ &\overset{(b)}{=} \arg\max_{\boldsymbol{a}} \sum_{i=1}^{k} \log F_i(o_i, a_i) \\ &\overset{(c)}{=} \left(\arg\max_{a_1 \in \mathcal{A}_i} F_1(o_1, a_1), \cdots, \arg\max_{a_k \in \mathcal{A}_k} F_k(o_k, a_k)\right), \end{aligned} \tag{33}$$

where (a) is based on the fact F and $f(s)$ are positive, (b) is due to the IGC condition. Combining with the IGM condition

$$\arg\max_{\boldsymbol{a} \in \mathcal{A}} Q_{\text{tot}}(s, \boldsymbol{a}) = \left(\arg\max_{a_1 \in \mathcal{A}_1} Q_1(o_1, a_1), \cdots, \arg\max_{a_k \in \mathcal{A}_k} Q_k(o_k, a_k)\right), \forall s \in \mathcal{S}. \tag{34}$$

we can conclude that

$$\left(\arg\max_{a_i \in \mathcal{A}_i} F_i(o_i, a_i)\right)_{i=1}^{k} = \left(\arg\max_{a_1 \in \mathcal{A}_1} Q_i(o_i, a_i)\right)_{i=1}^{k}.$$

Then we complete the proof.

A.3 PROOF OF LEMMA 3

Lemma 3 [Independent Transition Probability]. *Define the independent forward and backward transition respectively as*

$$P_F\left(o^i_{t+1}|o^i_t\right) := P_i\left(o^i_t \to o^i_{t+1}|o^i_t\right) = \frac{F_i\left(o^i_t \to o^i_{t+1}\right)}{F_i\left(o^i_t\right)}, \tag{35}$$

and

$$P_B\left(o^i_t|o^i_{t+1}\right) := P_i\left(o^i_{t+1} \to o^i_t|o^i_{t+1}\right) = \frac{F_i\left(o^i_{t+1} \to o^i_t\right)}{F_i\left(o^i_{t+1}\right)}. \tag{36}$$

Then we have

$$\begin{aligned} \sum_{\tau \in \mathcal{T}_{s,f}} P_F(\tau) = 1, \forall s \in \mathcal{S} \backslash \{s_f\}, \\ \sum_{\tau \in \mathcal{T}_{0,s}} P_B(\tau) = 1, \forall s \in \mathcal{S} \backslash \{s_0\}, \end{aligned} \tag{37}$$

where $\mathcal{T}_{s,f}$ *is the set of trajectories starting in* s *and ending in* s_f *and* $\mathcal{T}_{0,s}$ *is the set of trajectories starting in* s_0 *and ending in* s*.*

Proof: When the maximum length of trajectories is not more than 1, we have

$$\sum_{\tau \in \mathcal{T}_{s,f}} P_F(\tau) = 1. \tag{38}$$

Then we have the following results by induction:

$$\begin{aligned} \sum_{\tau \in \mathcal{T}_{s,f}} P_F(\tau) = \sum_{s' \in \mathcal{C}(s)} \sum_{\tau \in \mathcal{T}_{s \to s',f}} P_F(\tau) = \sum_{\boldsymbol{o}' \in \mathcal{C}(\boldsymbol{o})} P_F(\boldsymbol{o}'|\boldsymbol{o}) \sum_{\tau \in \mathcal{T}_{s',f}} P_F(\tau) \\ = \sum_{k} \sum_{o_i' \in \mathcal{C}(o_i)} P_F(o_i'|o_i) \sum_{\tau \in \mathcal{T}_{s',f}} P_F(\tau) = 1, \end{aligned} \tag{39}$$

where $\mathcal{C}(\cdot)$ is the children set of the current state or observation and the last equation is based on the fact $\sum_{o_i' \in \mathcal{C}(o_i)} P_F(o_i'|o_i) = 1$. Since the proof process of P_B is similar to that of P_F, it is omitted here.

B EXPERIMENTAL DETAILS

B.1 HYPER-GRID ENVIRONMENT

Here we present the experimental details on the Hyper-Grid environments. Figure 5 shows the curve of the flow matching loss function with the number of training steps. The loss of our proposed algorithm gradually decreases, ensuring the stability of the learning process. For some RL algorithms based on the state-action value function estimation, the loss usually oscillates. This may be because RL-based methods use experience replay buffer and the transition data distribution is not stable enough. The method we propose uses an on-policy based optimization method, and the data distribution changes with the current sampling policy, hence the loss function is relatively stable. We set the same number of training steps for all algorithms for a fair comparison. Moreover, we list the key hyperparameters of the different algorithms in Tables 2 3 4 5.

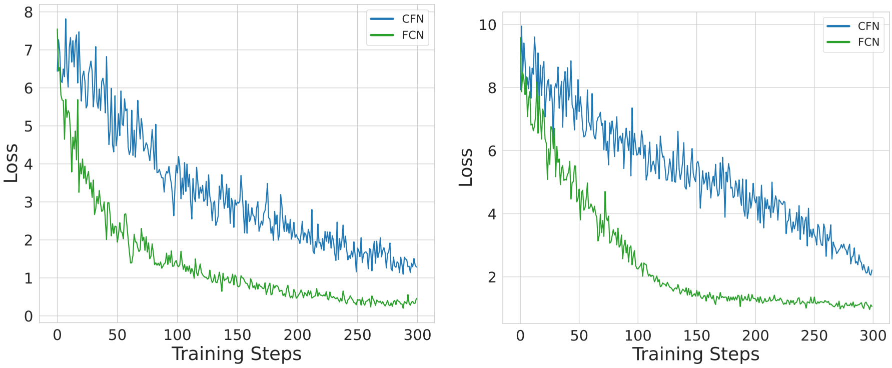

Figure 5: The flow matching loss of different algorithm.

We study the effect of different reward in Figure 6. In particular, we set $R_0 = \{10^{-1}, 10^{-2}, 10^{-4}\}$ for different task challenge. A smaller value of R_0 makes the reward function distribution more

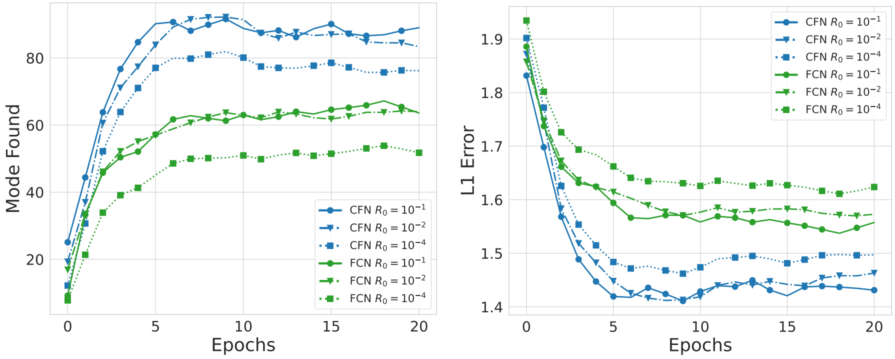

Figure 6: The effect of different reward R_0 on different algorithm according to L1 error and mode found.

sparse, which makes policy optimization more difficult Bengio et al. (2021a); Riedmiller et al. (2018); Trott et al. (2019). As shown in Figure 6, we found that our proposed method is robust with the cases $R_0 = 10^{-1}$ and $R_0 = 10^{-2}$. When the reward distribution becomes sparse, the performance of the proposed algorithm degrades slightly.

Table 2: Hyper-parameter of MAPPO under different environments

	Hyper-Grid-v1	Hyper-Grid-v2	Hyper-Grid-v3
Train Steps	20000	20000	20000
Agent	2	2	3
Grid Dim	2	3	3
Grid Size	[8,8]	[8,8]	[8,8]
Actor Network Hidden Layers	[256,256]	[256,256]	[256,256]
Optimizer	Adam	Adam	Adam
Learning Rate	0.0001	0.0001	0.0001
Batchsize	64	64	64
Discount Factor	0.99	0.99	0.99
PPO Entropy	1e-1	1e-1	1e-1

Table 3: Hyper-parameter of MASAC under different environments

	Hyper-Grid-v1	Hyper-Grid-v2	Hyper-Grid-v3
Train Steps	20000	20000	20000
Grid Dim	2	3	3
Grid Size	[8,8]	[8,8]	[8,8]
Actor Network Hidden Layers	[256,256]	[256,256]	[256,256]
Critic Network Hidden Layers	[256,256]	[256,256]	[256,256]
Optimizer	Adam	Adam	Adam
Learning Rate	0.0001	0.0001	0.0001
Batchsize	64	64	64
Discount Factor	0.99	0.99	0.99
SAC Alpha	0.98	0.98	0.98
Target Network Update	0.001	0.001	0.001

Table 4: Hyper-parameter of FCN under different environments

	Hyper-Grid-v1	Hyper-Grid-v2	Hyper-Grid-v3
Train Steps	20000	20000	20000
R_2	2	2	2
R_1	0.5	0.5	0.5
Grid Dim	2	3	3
Grid Size	[8,8]	[8,8]	[8,8]
Trajectories per steps	16	16	16
Flow Network Hidden Layers	[256,256]	[256,256]	[256,256]
Optimizer	Adam	Adam	Adam
Learning Rate	0.0001	0.0001	0.0001
ϵ	0.0005	0.0005	0.0005

Table 5: Hyper-parameter of CFN under different environments

	Hyper-Grid-v1	Hyper-Grid-v2	Hyper-Grid-v3
Train Steps	20000	20000	20000
Trajectories per steps	16	16	16
R_2	2	2	2
R_1	0.5	0.5	0.5
Grid Dim	2	3	3
Grid Size	[8,8]	[8,8]	[8,8]
Flow Network Hidden Layers	[256,256]	[256,256]	[256,256]
Optimizer	Adam	Adam	Adam
Learning Rate	0.0001	0.0001	0.0001
ϵ	0.0005	0.0005	0.0005