"Think Before You Speak": Improving Multi-Action Dialog Policy by Planning Single-Action Dialogs

Shuo Zhang[1], Junzhou Zhao[1*], Pinghui Wang[1*], Yu Li[1], Yi Huang[2], Junlan Feng[2]
[1]MOE KLINNS Lab, Xi'an Jiaotong University, Xi'an 710049, P. R. China
[2]JIUTIAN Team, China Mobile Research, Beijing 100053, P. R. China
{zs412082986, liyu1998}@stu.xjtu.edu.cn, {junzhou.zhao, phwang}@mail.xjtu.edu.cn,
{huangyi, fengjunlan}@chinamobile.com

Code： https://github.com/ShuoZhangXJTU/PEDP.

（IJCAI-2022)

gesis Leibniz-Institut für Sozialwissenschaften

Reported by Zicong Dou

1. Introduction
2. Approach
3. Experiments

Introduction

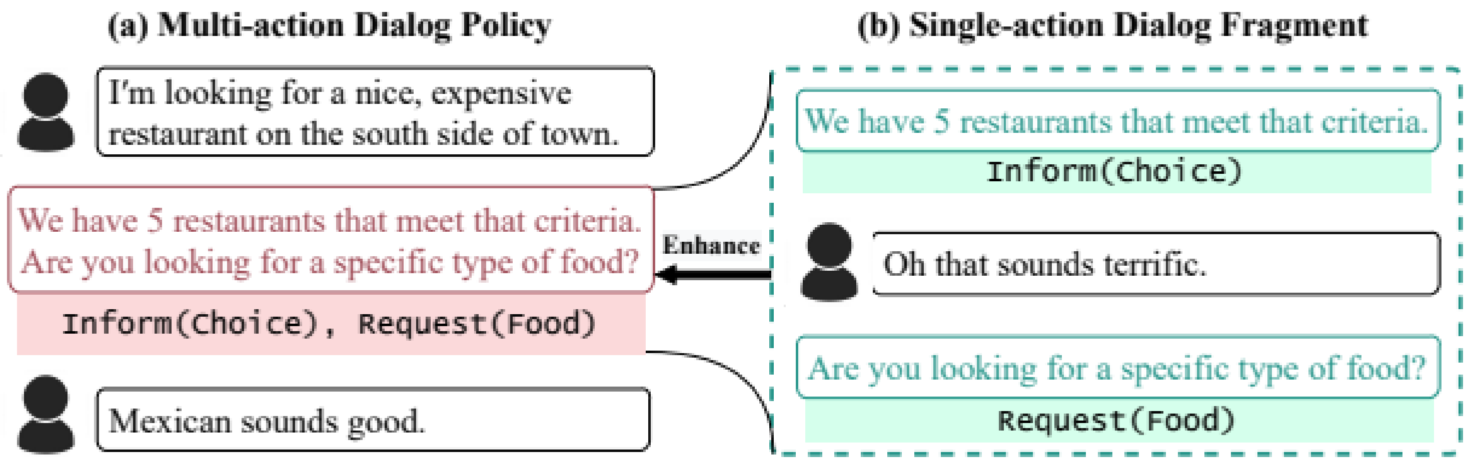


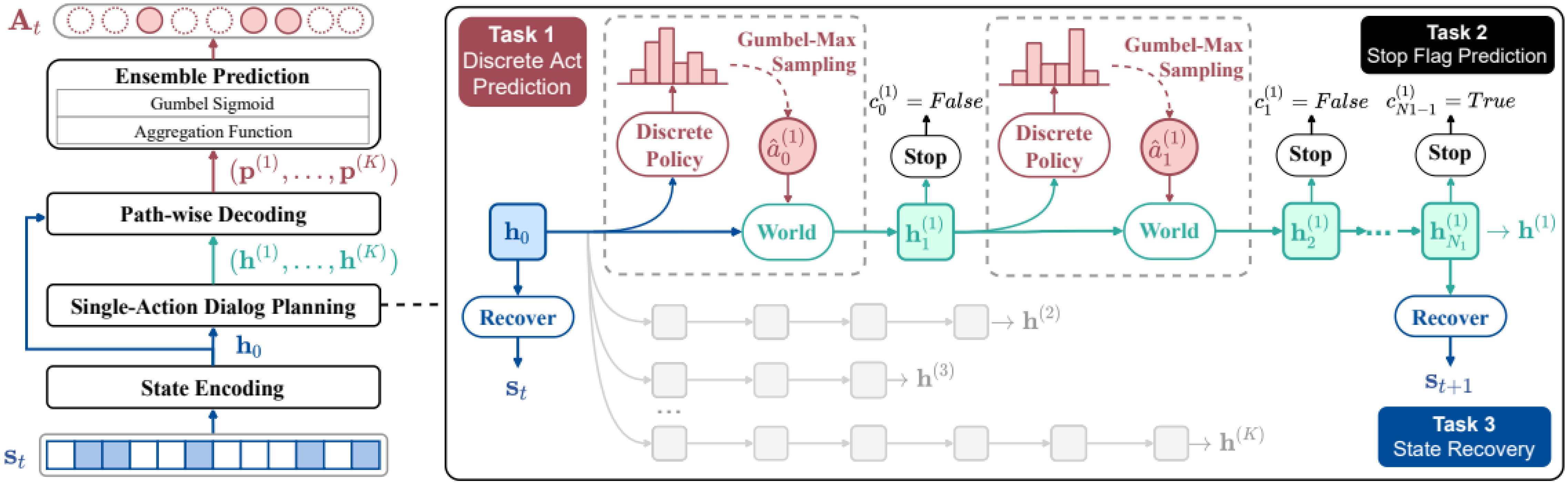
Figure 1: (a) Example dialog under multi-action dialog policy[2]. We propose to learn single-action dialog dynamics (b) to model conditional act combination patterns and enhance multi-action prediction.

[2]A dialog policy responses by predicting atomic dialog actions represented as "Domain-Intent(Slot)" phrases. We omit the domain ("restaurant") for clarity.

We propose Planning Enhanced Dialog Policy (PEDP), a novel multi-task learning framework that learns single action dialog dynamics to enhance multi-action prediction.

- 一个宏动作，它是一组独立的原子对话动作，用作当前系统响应。
- 每个原子对话动作都是域名、动作类型和插槽名称的串联，例如"hotel-inform-area"。

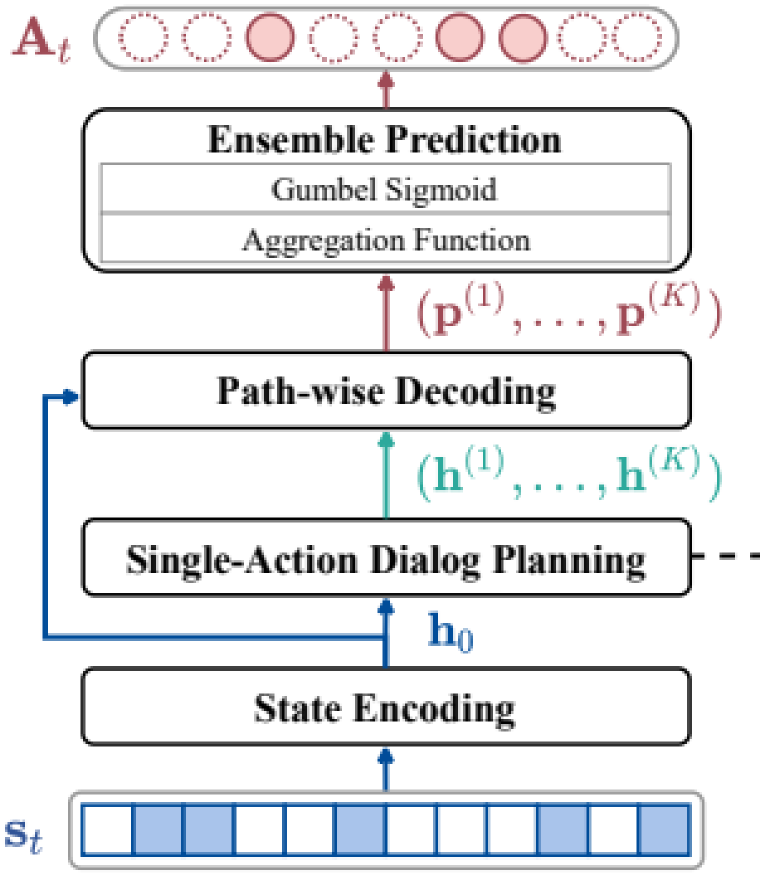
Approach



(a) (b)

Figure 2: (a) The Planning-Enhanced Dialog Policy (PEDP) framework. It utilizes a *single action dialog planning* module (b) to incorporate contextually relevant contents before multi-action prediction. A total of K single-action dialog procedures are planned, with the k-th path looking ahead N_k steps under single-action dialog dynamics. At each step, the discrete policy model predicts an atomic dialog action a_n given the previous dialog state embedding $\mathbf{h}_{n-1}$. The world model, which simulates user behavior, responds to the predicted action a_n and updates the dialog state embedding from $\mathbf{h}_{n-1}$ to $\mathbf{h}_n$.

Approach



(a)

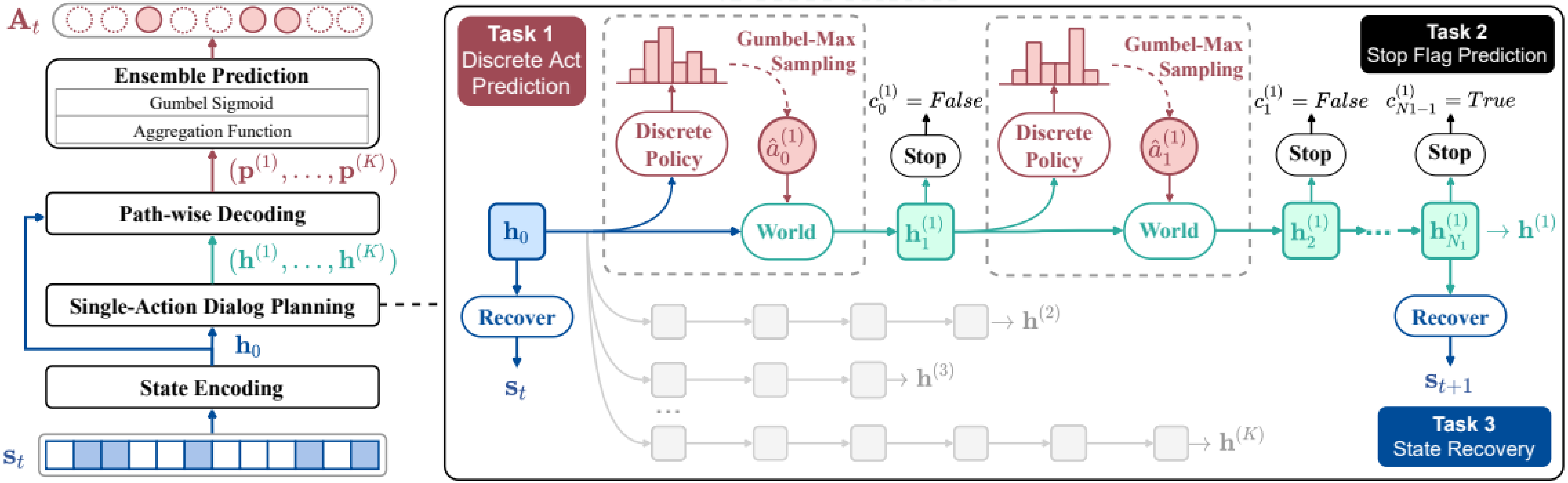
$\mathbf{s}_t$ into a dialog state embedding $\mathbf{h}_t$. Given the current state embedding $\mathbf{h}_t$, we plan K independent single-action dialog paths, and the k-th dialog path is represented by a vector $\mathbf{h}^{(k)}$, $k = 1, \dots, K$. Our model then decodes each dialog path to a probability distribution over atomic dialog actions, i.e., $\mathbf{p}^{(k)}$. Finally, these distributions $\{\mathbf{p}^{(k)}\}_{k=1}^{K}$ are aggregated to form a unified distribution, from which atomic dialog actions in the macro-action $\mathbf{A}_t$ are sampled.

State Encoding

$$\mathbf{h}_t = \mathtt{FFN}_{enc}(\mathbf{s}_t) = \mathtt{ReLU}(\mathbf{s}_t W_1 + b_1) W_2 + b_2. \qquad (1)$$

In what follows, this dialog state embedding $\mathbf{h}_t$ will serve as the initial dialog state embedding for planning.

Approach

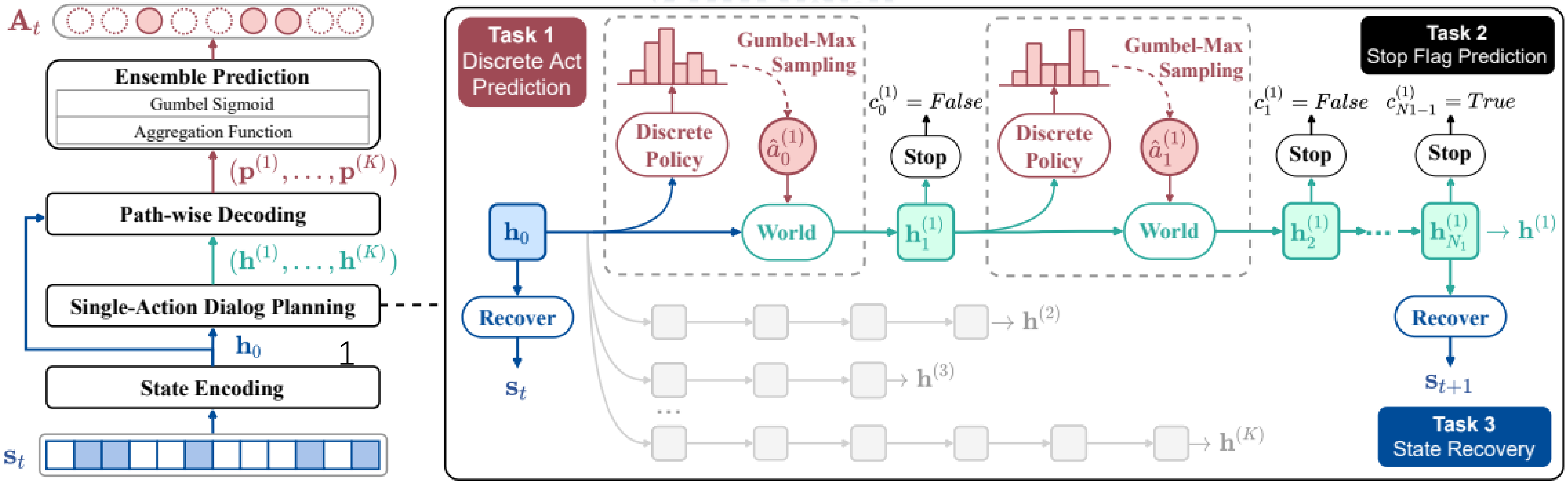


(a) (b)

Single-Action Dialog Planning

look ahead several steps. Let $\mathbf{h}_{t,n}^{(k)}$ denote the dialog state embedding at the n-th step in the k-th dialog for $n = 0, \dots, N_k$ where N_k is the length of the k-th dialog, and $\mathbf{h}_{t,0}^{(k)} = \mathbf{h}_t$, $\mathbf{h}_{t,N_k}^{(k)} = \mathbf{h}^{(k)}$. The last dialog state embedding $\mathbf{h}^{(k)}$ estimates the hidden vector of the future dialog state $\mathbf{s}_{t+1}$ and summarizes the planned single-action dialog. In what follows, we describe how to plan a single step from $\mathbf{h}_{t,n}^{(k)}$ to obtain $\mathbf{h}_{t,n+1}^{(k)}$.

Approach



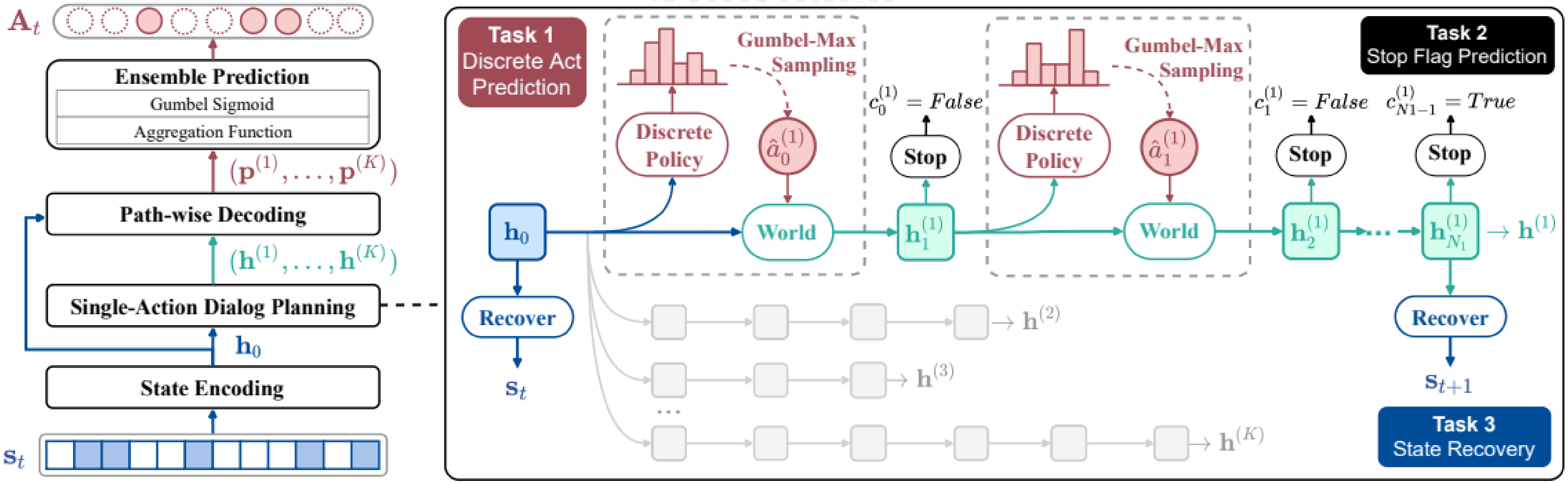
(a) **(b)**

Single-Action Dialog Planning

$$a_n = \mathtt{DP}(\mathbf{h}_n) \triangleq \mathrm{GumbelSoftmax}^{(\tau_d)}(\mathbf{h}_n W_d + b_d)$$
$$\mathbf{h}_{n+1} = \mathtt{World}(\mathbf{h}_n, a_n) \triangleq \mathtt{GRU}(\mathbf{h}_n, \mathtt{Emb}(a_n)). \quad (2)$$

Here, DP is implemented as a single linear layer followed by a Gumbel-Softmax function [Jang *et al.*, 2016] parameterized by τ_d. The Gumbel-Softmax function draws an atomic dialog action sample from a categorical distribution, diversifying the planned dialogs. τ_d is selected to balance the approximation bias and the magnitude of gradient variance. The world model is implemented using a GRU to model dialog state transitions, and $\mathtt{Emb}(a_n)$ denotes the embedding vector of atomic dialog action a_n.

Approach



(a) (b)

Single-Action Dialog Planning

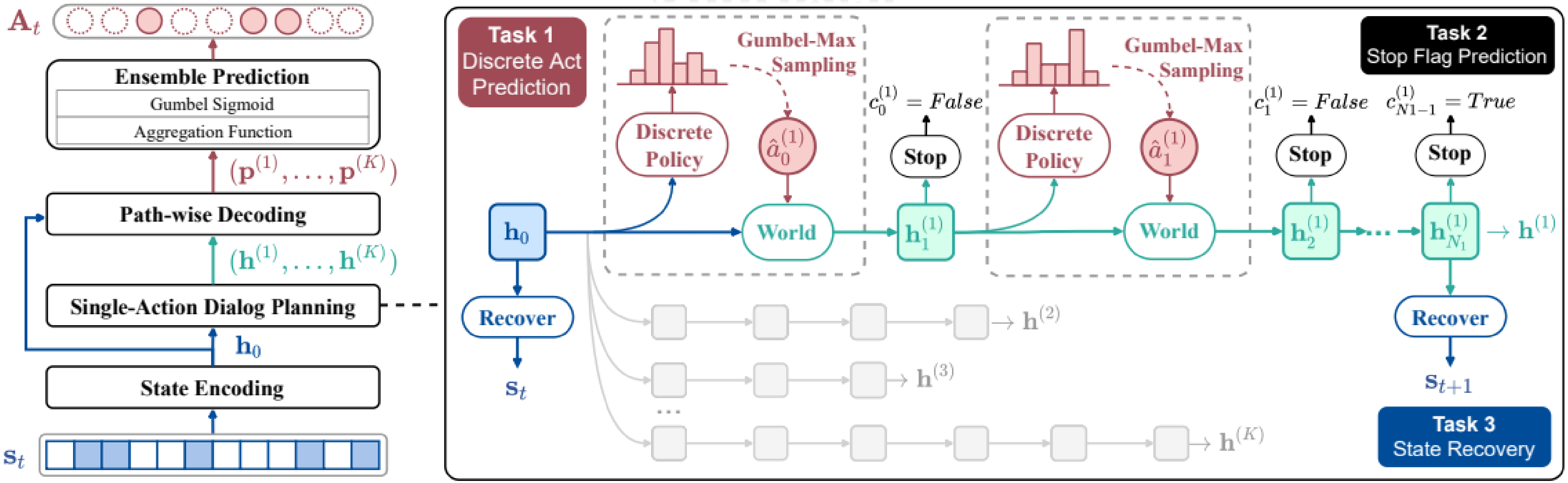
$$c_n = \text{GumbelSoftmax}^{(\tau_s)}(\texttt{FFN}_{st}([\mathbf{h}_0 : \mathbf{h}_{n+1}]))$$

where c_n is a binary variable, ":" denotes vector concatenation, and FFN is a 2-layer fully-connected feed-forward network using the ReLU activation function in the middle layer.

$$\mathbf{s}_t = \texttt{Recover}(\mathbf{h}_0)$$
$$\mathbf{s}_{t+1} = \texttt{Recover}(\mathbf{h}_N) \quad (3)$$

Here, Recover is implemented by a 2-layer FFN and is only used during the training stage.

Approach



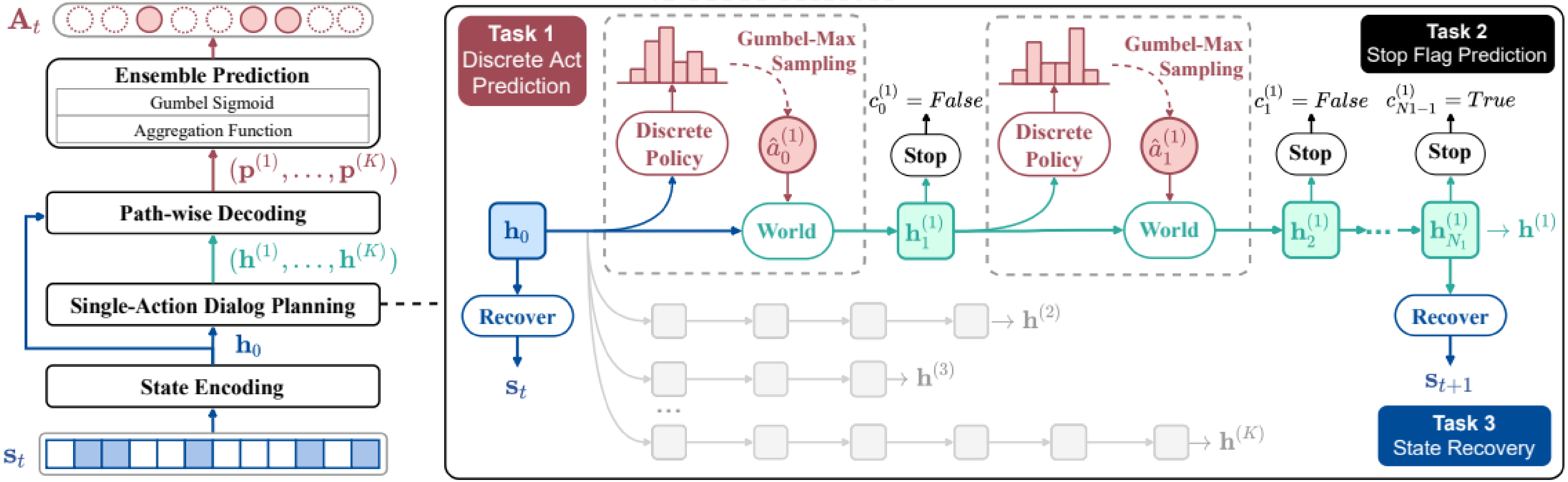
(a) (b)

Path-wise Decoding

Specifically, we instantiate the decoder $\mathbf{p}^{(k)} = [\mathbf{p}_1^{(k)} : \ldots : \mathbf{p}_M^{(k)}]$, where k refers to the planned path and M is the size of the action space. Each $\mathbf{p}_m^{(k)}, m = 1, \ldots, M$ is a vector computed as:

$$\mathbf{p}_m^{(k)} = \mathrm{FFN}_m^{dec}([\mathbf{h}_0 : \mathbf{h}^{(k)}]).$$

Approach



(a) (b)

Ensemble Prediction

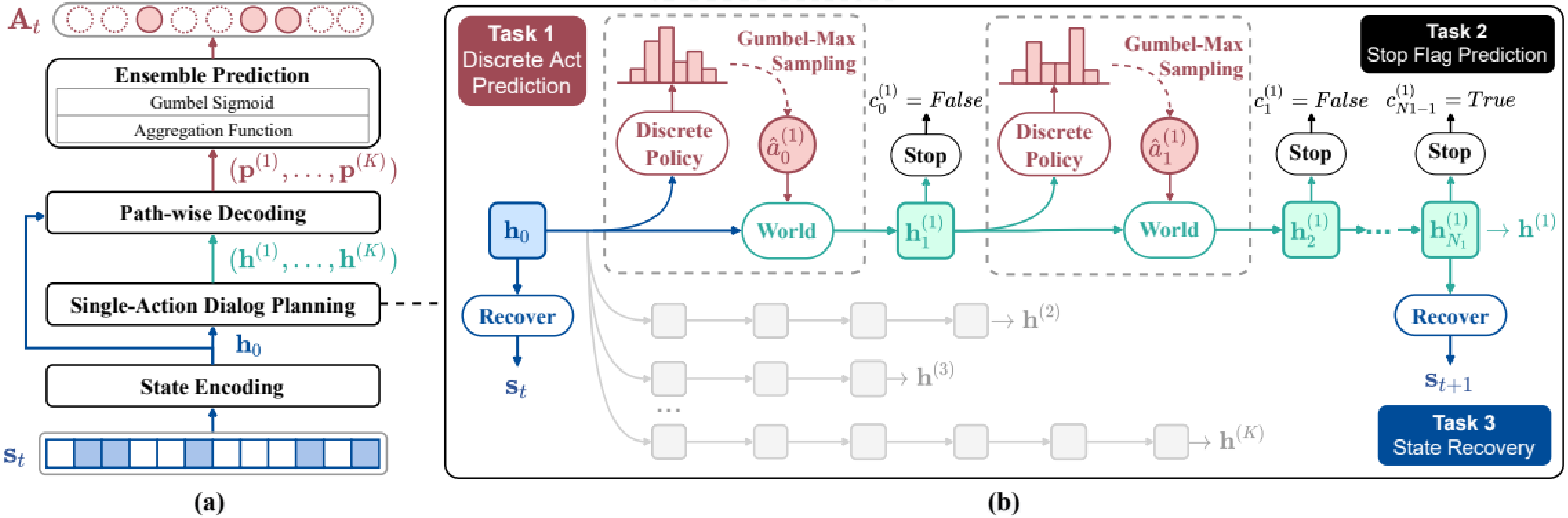
$$\mathbf{P}_t = \mathrm{Aggr}(\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(K)})$$

where $\mathrm{Aggr}(\cdot)$ is the mean average in our case.

$$\mathbf{A}_t = \mathrm{GumbelSigmoid}(\mathbf{P}_t) = \frac{e^{(\mathbf{P}_t + g_1)/\tau}}{e^{(\mathbf{P}_t + g_1)/\tau} + e^{(\mathbf{P}_t + g_2)/\tau}}$$

Here $\mathrm{GumbelSigmoid}(\cdot)$ is a modification of the Gumbel-Softmax function, regarding sigmoid as a softmax with two logits p and 0. τ denotes the temperature factor, g_1 and g_2 are Gumbel noises.

Approach



Training

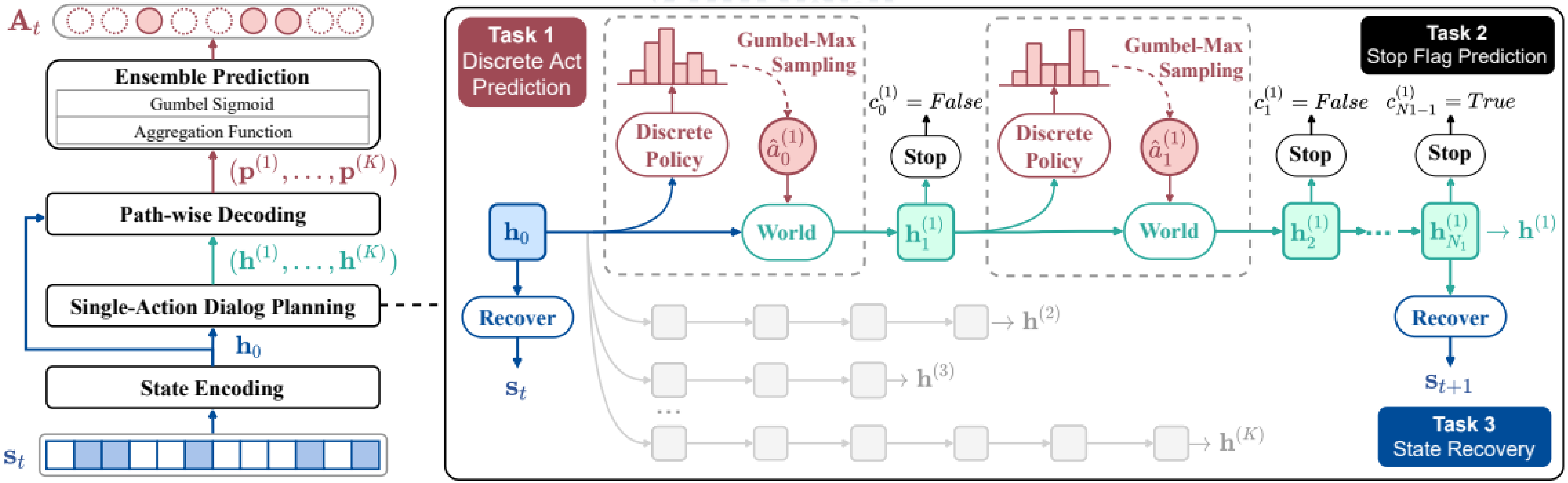
Task 1: Discrete Act Prediction (DAP)

$$p(\boldsymbol{a}|\mathbf{h}_0) = p_\theta(a_0|\mathbf{h}_0)\prod_{n=1}^{N-1}\underbrace{p_\theta(a_n|\mathbf{h}_n)}_{\text{DAP}}\underbrace{p_\phi(\mathbf{h}_n|a_{n-1},\mathbf{h}_{n-1})}_{\text{state transition}}$$

$$\boldsymbol{a} = (a_0, \ldots, a_{N-1})$$

where θ and ϕ denotes trainable parameters for the discrete dialog policy model and the world model, respectively.

Approach



(a) (b)

Training

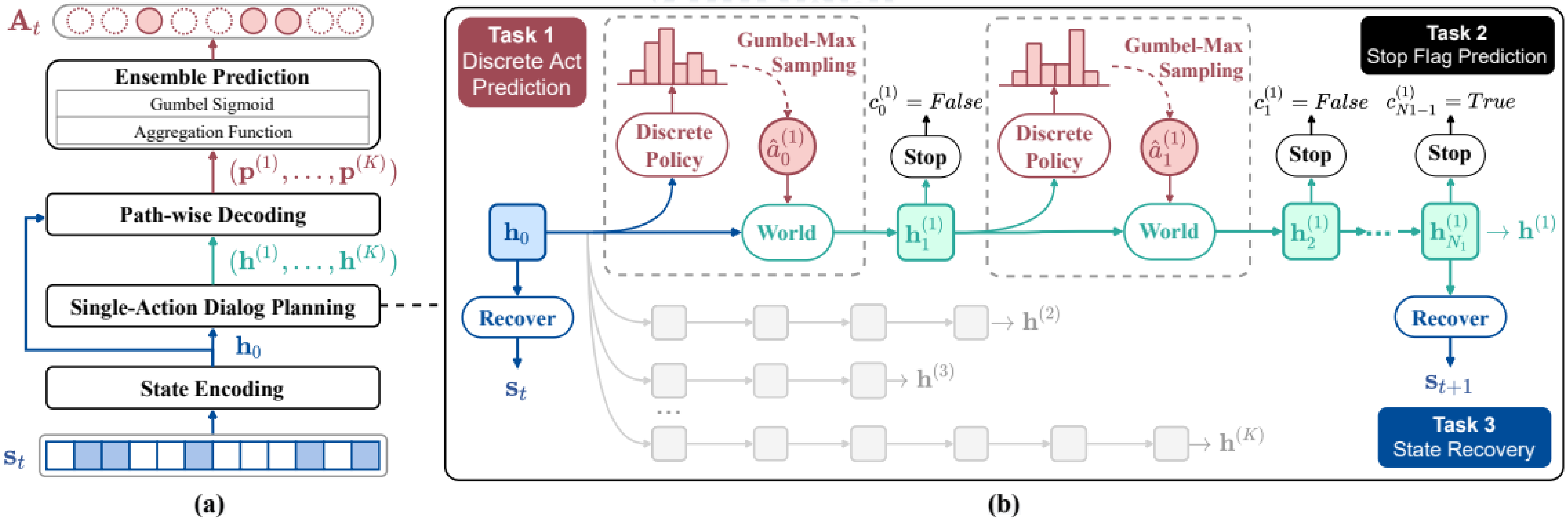
Task 2: Stop Flag Prediction (SFP)

$$p(\mathbf{c}|\mathbf{h}_0) = \prod_{n=0}^{N-1} \underbrace{p_\gamma(c_n|\mathbf{h}_{n+1}, \mathbf{h}_0)}_{\text{SFP}} \underbrace{p_{\phi,\theta}(\mathbf{h}_{n+1}|\mathbf{h}_n)}_{\text{1-step planning}}$$

$$\mathbf{c} = (c_0, \ldots, c_{N-1})$$

where γ parameterizes the stop prediction model, the joint probability of $p_{\phi,\theta}(\mathbf{h}_{n+1}|\mathbf{h}_n)$ is factorized as $p_\phi(\mathbf{h}_{n+1}|a_n, \mathbf{h}_n)p_\theta(a_n|\mathbf{h}_n)$ of state transition and discrete act prediction.

Approach



Training

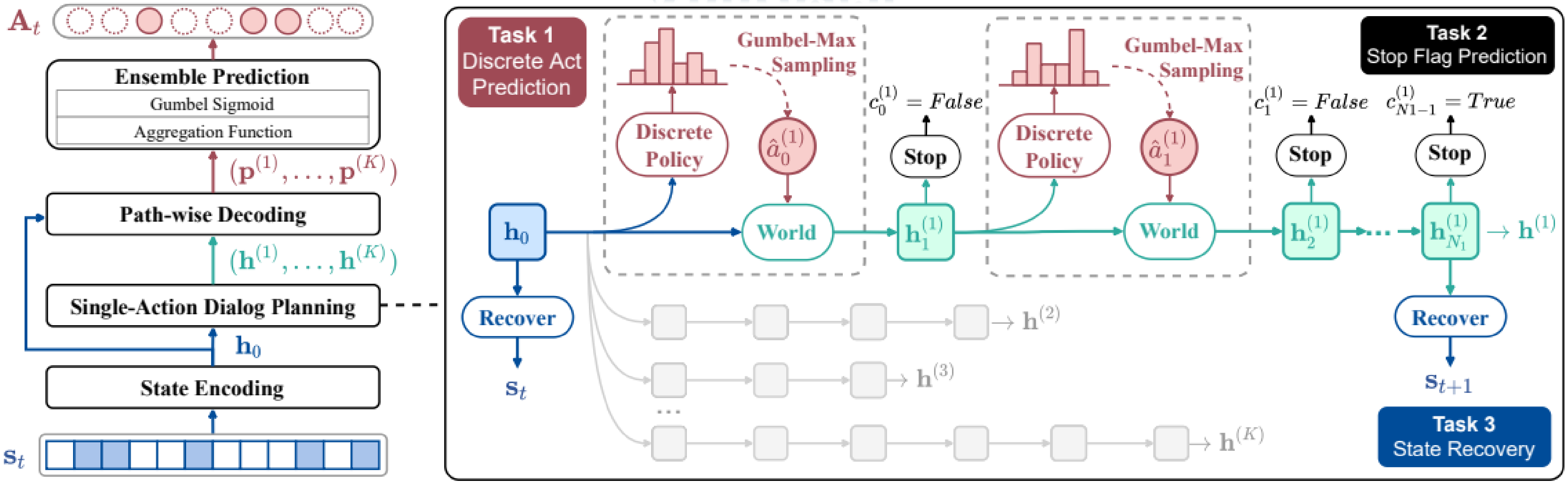
Task 3: State Recovery (SR)

$$p(\mathbf{s}_t) = \underbrace{p_\zeta(\mathbf{s}_t|\mathbf{h}_0)}_{\text{SR}} \underbrace{p_\eta(\mathbf{h}_0|\mathbf{s}_t)}_{\text{state encoding}}$$

$$p(\mathbf{s}_{t+1}|\mathbf{s}_t) = \underbrace{p_\zeta(\mathbf{s}_{t+1}|\mathbf{h}_N)}_{\text{SR}} \underbrace{p_\eta(\mathbf{h}_0|\mathbf{s}_t)}_{\text{state encoding}} \prod_{n=0}^{N-1} \underbrace{p_{\phi,\theta}(\mathbf{h}_{n+1}|\mathbf{h}_n)}_{\text{1-step planning}}$$

where η and ζ denotes trainable parameters for state encoder and the `Recover`, respectively. The joint probability $p_{\phi,\theta}(\mathbf{h}_{n+1}|\mathbf{h}_n)$ is the same as explained in Task 2.

Approach



(a) (b)

Training

Task 4: Multi-Action Prediction (MAP)

$$p(\mathbf{A}_t|\mathbf{s}_t) = \underbrace{p_\omega(\mathbf{A}_t|\mathbf{h}_0, \mathbf{h}_N)}_{\text{MAP}} \underbrace{p_\eta(\mathbf{h}_0|\mathbf{s}_t)}_{\text{state encoding}} \prod_{n=0}^{N-1} \underbrace{p_{\phi,\theta}(\mathbf{h}_{n+1}|\mathbf{h}_n)}_{\text{1-step planning}}$$

where ω denotes trainable parameters for the decoder. The rest is the same as explained in Task 3.

Experiments

	MultiWOZ				
Agent	**Turn**	**Match**	**Rec**	**F1**	**Success**
DiaMultiClass	11.46 ±0.56	0.68 ±3.9%	0.81 ±3.2%	0.81 ±2.1%	67.3 ±3.69
+ sample	9.23 ±0.2	0.82 ±1.1%	0.90 ±1.8%	0.77 ±1.2%	81.4 ±1.78
DiaSeq (beam)	9.06 ±0.67	0.81 ±0.4%	0.9 ±1.2%	0.86 ±0.9%	81.4 ±0.16
greedy	10.35 ±0.04	0.68 ±1.5%	0.80 ±0.5%	0.77 ±0.5%	67.7 ±1.02
+ sample	8.82 ±0.1	0.86 ±0.6%	0.93 ±0.4%	0.81 ±0.5%	86.9 ±0.49
DiaMultiDense	9.66 ±0.15	0.85 ±0.6%	0.94 ±0.4%	0.87 ±0.6%	86.3 ±0.64
- sample	12.75 ±0.77	0.61 ±6%	0.72 ±5.4%	0.80 ±2.3%	58.4 ±6.05
gCAS	11.69 ±0.53	0.56 ±1.4%	0.72 ±0.4%	0.76 ±1.4%	58.8 ±2.82
GP-MBCM[5]	**2.99**	0.44	-	0.19	28.9
ACER[5]	10.49	0.62	-	0.78	50.8
PPO [5]	15.56	0.60	0.72	0.77	57.4
ALDM[5]	12.47	0.69	-	0.81	61.2
GDPL	7.54 ±0.43	0.84 ±0.9%	0.89 ±2.2%	**0.88** ±1.2%	83.2 ±1.48
DiaAdv	8.90 ±0.18	0.87 ±0.9%	0.94 ±0.75%	0.85 ±0.58%	87.6 ±0.9
- sample	11.9 ±0.88	0.62 ±5.9%	0.73 ±4.6%	0.80 ±2.1%	61.7 ±5.59
PEDP	8.69 ±0.15	**0.88** ±1.3%	**0.97** ±0.4%	0.87 ±1.1%	**90.6** ±0.68
- planning	9.66 ±0.15	0.85 ±0.6%	0.94 ±0.4%	0.87 ±0.6%	86.3 ±0.64
- ensemble	9.25 ±0.43	0.88 ±1.97%	0.96 ±0.8%	0.85 ±2.5%	89.1 ±1.74
- sample	8.85 ±0.22	0.82 ±2.5%	0.93 ±1.4%	0.86 ±1.6%	83.4 ±1.01

Table 1: Interactive evaluation results. We simulate 1,000 dialogs per run and report the mean and standard deviation over 5 runs.

Experiments

	MultiWOZ			SGD (scaling)		
Agent	**F1%**	**Precision%**	**Recall%**	**F1%**	**Precision%**	**Recall%**
DiaMultiClass	39.41 ±1.08	54.59 ±1.71	34.32 ±1.32	58.09 ±0.63	81.29 ±1.13	46.29 ±0.57
+ sample	38.91 ±0.74	47.28 ±0.68	37.56 ±1.08	58.03 ±0.64	81.48 ±0.18	46.14 ±0.80
DiaSeq (beam)	44.64 ±2.08	51.91 ±0.99	43.66 ±2.27	63.13 ±0.18	86.04 ±0.5	50.83 ±0.30
greedy	48.34 ±0.45	54.71 ±0.21	48.84 ±0.84	63.21 ±0.35	86.31 ±0.7	50.85 ±0.40
+ sample	37.82 ±0.45	43.02 ±0.48	38.91 ±0.64	62.64 ±1.03	85.54 ±1.62	50.40 ±0.76
DiaMultiDense	35.92 ±0.54	51.93 ±0.33	30.10 ±0.69	57.85 ±0.68	80.64 ±0.43	46.21 ±0.89
- sample	34.35 ±0.62	52.14 ±0.19	27.74 ±0.74	56.69 ±0.62	79.54 ±0.88	45.19 ±0.75
gCAS	50.01 ±0.62	55.56 ±0.59	51.21 ±1.74	76.37 ±1.60	77.70 ±1.46	79.99 ±1.03
GDPL	31.89±0.96	50.14 ±0.79	24.99 ±1.14	-	-	-
+ sample	34.60 ±0.47	45.01 ±0.24	31.54 ±0.80	-	-	-
DiaAdv	40.97 ±0.95	53.44 ±0.50	36.84 ±1.30	-	-	-
- sample	41.71 ±0.47	56.46 ±0.45	36.28 ±1.48	-	-	-
PEDP	64.63 ±0.16	77.03 ±1.39	61.77 ±1.01	84.12 ±0.38	91.66 ±0.52	81.19 ±0.4
- planning	35.92 ±0.54	51.93 ±0.33	30.10 ±0.69	57.85 ±0.68	80.64 ±0.43	46.21 ±0.89
- ensemble	64.34 ±0.29	77.63 ±2.04	60.85 ±1.54	83.31 ±0.55	91.66 ±0.78	80.10 ±0.55
- sample	**66.95** ±0.45	**78.11** ±3.03	**65.02** ±1.22	**84.74** ±0.55	**92.07** ±0.97	**81.30** ±0.82

Table 2: Standard evaluation results. We report the mean and standard deviation over 5 runs.

Experiments

Dialog pair	**Win**	**Lose**	**Tie**	α
PEDP vs. DiaSeq	41.7	31.3	27.0	0.820
PEDP vs. DiaAdv	36.5	27.6	35.9	0.856
PEDP vs. GDPL	32.6	26.5	40.9	0.839

Table 3: Human evaluation results. We report the mean over 9 judges and Krippendorff's alpha (α) that measures the inter-rater reliability. Typically, results are considered reliable if $\alpha > 0.800$.

Thank you！