D. Algorithm tables

As introduced in Section 5, Trifle generally works in two phases: rejection sampling for action generation and beam search for action selection. The main algorithm is illustrated in Algorithm 1, where we take the current state s_t as well as the past trajectory $\tau_{<t}$ as input, utilize the specified value estimate f_v as a heuristic to guide beam search, and output the best trajectory. After that, we extract the current action a_t from the output trajectory to execute in the environment.

At the first step of the beam search, we will perform rejection sampling to obtain a candidate action set $\mathbf{a_t}$ (line 4 of Algorithm 1). The concrete rejection sampling procedure for s-Trifle is detailed in Algorithm 2. The major modification of m-Trifle compared to s-Trifle is the adoption of a multi-step value estimate instead of the single-step value estimate, which is also shown in Algorithm 3.

Algorithm 1 Trifle with Beam Search

1: **Input:** past trajectory $\tau_{<t}$, current state s_t, beam width N, beam horizon H, scaling ratio λ, sequence model $\mathcal{M}$,
value function f_v ▷ $f_v = \mathbb{E}[V_t]$ for s-Trifle and $\mathbb{E}[V_t^m]$ for m-Trifle
2: **Output:** The best trajectory
3: Let $\mathbf{x_t} \leftarrow \texttt{concat}(\tau_{<t}, s_t)\texttt{.reshape}(1, -1)\texttt{.repeat}(N, \texttt{dim} = 0)$ ▷ Batchify the input trajectory and prepare for the beam search
4: Perform rejection sampling to obtain $\mathbf{a_t}$ ▷ cf. Algorithm 2
5: Initialize $X_0 = \texttt{concat}(\mathbf{x_t}, \mathbf{a_t})$
6: **for** $t = 1, ..., H$ **do**
7: $X_{t-1} \leftarrow X_{t-1}\texttt{.repeat}(\lambda, \texttt{dim} = 0)$ ▷ Scale the number of trajectories from N to λN
8: $\mathcal{C}_t \leftarrow \{\texttt{concat}(\mathrm{x}_{t-1}, x) \mid \forall \mathrm{x}_{t-1} \in X_{t-1}, \text{sample } x \sim p_{\mathcal{M}}(\cdot \mid \mathrm{x}_{t-1})\}$ ▷ Candidate next-token prediction
9: $X_t \leftarrow \texttt{topk}_{X \in \mathcal{C}_t}(f_v(X), \texttt{k} = N)$ ▷ keep N most rewarding trajectories
10: **end for**
11: **return** $\texttt{argmax}_{X \in X_H}\ f_v(X)$

Algorithm 2 Rejection Sampling with Single-step Value Estimate

1: **Input:** past trajectory $\tau_{<t}$, current state s_t, dimension of action k, rejection rate $\delta > 0$
2: **Output:** The sampled action $a_t^{1:k}$
3: Let $x_t \leftarrow \texttt{concat}(\tau_{<t}, s_t)$
4: **for** $i = 1, ..., k$ **do**
5: Compute $p_{\mathrm{GPT}}(a_t^i \mid x_t, a_t^{<i})$ Note that $a_t^{<1} = \varnothing$.
6: Compute $p_{\mathrm{TPM}}(V_t \mid x_t, a_t^{<i}) = \sum_{a_t^{i:k}} p_{\mathrm{TPM}}(V_t \mid x_t, a_t^{1:k})$ ▷ The marginal can be efficiently computed by PC in linear time.
7: Compute $v_\delta = \texttt{max}_v\{v \in \texttt{val}(V_t) \mid p_{\mathrm{TPM}}(V_t \geq v \mid x_t, a_t^{<i}) \geq 1 - \delta\}$, for each $a_t^i \in \texttt{val}(A_t^i)$
8: Compute $\tilde{p}(a_t^i \mid x_t, a_t^{<i}; v_\delta) = \frac{1}{Z} \cdot p_{\mathrm{GPT}}(a_t^i \mid x_t, a_t^{<i}) \cdot p_{\mathrm{TPM}}(V_t \geq v_\delta \mid x_t, a_t^{\leq i})$ ▷ Apply Equation (2)
9: Sample $a_t^i \sim \tilde{p}(a_t^i \mid x_t, a_t^{<i}; v_\delta)$
10: **end for**
11: **return** $a_t^{1:k}$

Algorithm 3 Multi-step Value Estimate

1: **Input:** $\tau_{\leq t}$, sequence model $\mathcal{M}$, terminal timestep $t' > t$, discount γ
2: **Output:** The multi-step value estimate $\mathbb{E}\left[V_t^{\mathrm{m}}\right]$
3: Sample $r_t, s_{t+1}, a_{t+1}, r_{t+1}, ..., s_{t'}, a_{t'}, r_{t'}$ from $\mathcal{M}$
4: Compute $p_{\mathrm{TPM}}(\mathrm{RTG}_{t'} \mid \tau_{\leq t}, a_{t+1:t'}) = \sum_{s_{t+1:t'}} p_{\mathrm{TPM}}(\mathrm{RTG}_{t'} \mid \tau_{\leq t'})$ ▷ Marginalize over intermediate states $s_{t+1:t'}$
5: compute $\mathbb{E}\left[V_t^{\mathrm{m}}\right] = \sum_{h=t}^{t'} \gamma^{h-t} r_h + \gamma^{t'+1-t} \mathbb{E}_{\mathrm{RTG}_{t'} \sim p_{\mathrm{TPM}}(\cdot \mid s_t, a_{t:t'})}\left[\mathrm{RTG}_{t'}\right]$
6: **return** $\mathbb{E}\left[V_t^{\mathrm{m}}\right]$