Appendix A Action-Conditioned Temporal Abstraction State Transition

We implement action-conditioned state transition

$$p\left(z_t | a_t, z_{<t}, m_{<t}\right) = \begin{cases} \delta(z_t = z_{t-1}) & \text{if } m_{t-1} = 0 \text{ (COPY)}, \\ \tilde{p}(z_t | c_t) & \text{otherwise (UPDATE)} \end{cases}$$

where the action input a_t is only affecting the UPDATE operation and we feed it into the deterministic path as the following:

$$c_t = \begin{cases} c_{t-1} & \text{if } m_{t-1} = 0 \text{ (COPY)}, \\ f_z\left(z_{t-1} \| a_t, c_{t-1}\right) & \text{otherwise (UPDATE).} \end{cases}$$

Appendix B Goal-Oriented Navigation

Algorithm 1 Goal-oriented navigation with imagination (single episode)

Input: environment `env`, environment model $\mathcal{E}$, maximum length of imagined trajectories l_{img}
Output: reward r

Initialize reward $r \leftarrow 100$
Initialize environment $x \leftarrow$ `env.reset`$()$
Initialize context $X_{\text{ctx}} \leftarrow [x]$
Sample goal position and extract goal map feature g
while $r > 0$ **do**
    Sample action a from imagination-based planner given $\mathcal{E}, X_{\text{ctx}}, g, l_{\text{img}}$ (Algorithm 2)
    Do action $x \leftarrow$ `env.step`(a)
    Update context $X_{\text{ctx}} \leftarrow X_{\text{ctx}} + [x]$
    Update reward $r \leftarrow r - 1$
    if current position is at the goal position **then**
        break
return reward r

Algorithm 2 Imagination-based planner

Input: environment model $\mathcal{E}$, previously observed sequence (context) X_{ctx}, maximum length of imagined trajectories l_{img}, goal map feature g
Output: action a_{max}

Initialize model $\mathcal{E}$ with $c_0 = f_{\text{ctx}}\left(X_{\text{ctx}}\right)$
Initialize $d_{\text{max}} \leftarrow -\infty$
Initialize $a_{\text{max}} \leftarrow$ None
Set a list $\mathcal{A}$ of all possible action sequences based on l_{img}
for each action seuqence A in $\mathcal{A}$ **do**
    Get a sequence of states $\mathcal{S} \in \mathbb{R}^{l_{\text{img}} \times d}$ by doing imagination with model $\mathcal{E}$ conditioned on A
    Compute cosine-similarity $D \in \mathbb{R}^{l_{\text{img}}}$ between all states $\mathcal{S}$ and goal map feature g
    if $\max(D) > d_{\text{max}}$ **then**
        Update $d_{\text{max}} \leftarrow \max(D)$
        Update $a_{\text{max}} \leftarrow A[0]$
return action a_{max}

Appendix C Implementation Details

For bouncing balls, we define the reconstruction loss (data likelihood) by using binary cross-entropy. The images from 3D maze are pre-processed by reducing the bit depth to 5 bits (Kingma & Dhariwal, 2018) and therefore the reconstruction loss is computed by using Gaussian distribution. We used the AMSGrad (Reddi et al., 2018), a variant of Adam, optimizer with learning rate $5e-4$ and all mini-batchs are with 64 sequences with length $T = 20$. Both CNN-based encoder and decoder are composed of 4 convolution layers with ELU activations (Clevert et al., 2016). A GRU (Cho et al., 2014) is used for all RNNs with 128 hidden units. The state representations of temporal abstraction and observation abstraction are sampled from 8-dimensional diagonal Gaussian distributions.

Appendix D Evidence Lower Bound (ELBO)

We derive the ELBO without considering recurrent deterministic paths.

Log-likelihood $\log p\left(X\right)$

$$\begin{aligned}
\log p\left(X\right) &= \log \sum_{M} \int_{Z,S} p\left(X, Z, S, M\right) \\
&\geq \sum_{M} \int_{Z,S} q\left(Z, S, M|X\right) \log \frac{p\left(X, Z, S, M\right)}{q\left(Z, S, M|X\right)} \\
&= \sum_{M} \int_{Z,S} q\left(Z, S, M|X\right) \log \frac{p\left(X|Z, S\right) p\left(Z, S, M\right)}{q\left(Z, S, M|X\right)} \\
&= \sum_{M} \int_{Z,S} q\left(Z, S, M|X\right) \left[\log p\left(X|Z, S\right) + \log \frac{p\left(Z, S, M\right)}{q\left(Z, S, M|X\right)}\right] \\
&= \underbrace{\mathbb{E}_{q(Z,S,M|X)}\left[\log p\left(X|Z, S\right)\right]}_{\text{reconstruction}} - \underbrace{\mathrm{KL}\left[q\left(Z, S, M|X\right) || p\left(Z, S, M\right)\right]}_{\text{KL divergence}}
\end{aligned}$$

Decomposing $p\left(X|Z, S\right)$ **and** $p\left(Z, S, M\right)$

$$\begin{aligned}
p\left(X|Z, S\right) &= \prod_{t} \underbrace{p\left(x_t|s_t\right)}_{\text{decoder}} \\
p\left(Z, S, M\right) &= \prod_{t} \underbrace{p\left(z_t|z_{t-1}, m_{t-1}\right)}_{\text{temporal abstract transition}} \underbrace{p\left(s_t|s_{t-1}, z_t, m_{t-1}\right)}_{\text{observation abstract transition}} \underbrace{p\left(m_t|s_t\right)}_{\text{boundary prior}}
\end{aligned}$$

Decomposing $q\left(Z, S, M|X\right)$

$$\begin{aligned}
q\left(Z, S, M|X\right) &= q\left(M|X\right) q\left(Z, S|M, X\right) \\
&= q\left(M|X\right) q\left(Z|M, X\right) q\left(S|Z, M, X\right) \\
&= q\left(M|X\right) \prod_{t} q\left(z_t|M, X\right) q\left(s_t|z_t, M, X\right) \\
q(M|X) &= \prod_{t} q(m_t|X) = \prod_{t} \text{Bern}\left(m_t|\sigma(\varphi(X))\right)
\end{aligned}$$

Reconstruction Term in ELBO

$$\begin{aligned}
&\sum_{M} \int_{Z,S} q\left(Z, S, M|X\right) \log p\left(X|Z, S\right) \\
&= \underbrace{\sum_{M} q\left(M|X\right)}_{\text{sample } M} \int_{Z,S} q\left(Z, S|M, X\right) \log p\left(X|Z, S\right) \\
&\approx \int_{Z,S} q\left(Z, S|M, X\right) \sum_{t} \log p\left(x_t|s_t\right) \\
&= \int_{Z,S} \sum_{t} q\left(z_t, s_t|M, X\right) \log p\left(x_t|s_t\right) \\
&= \int_{Z,S} \sum_{t} \underbrace{q\left(z_t|M, X\right) q\left(s_t|z_t, M, X\right)}_{\text{sampling } z_t \text{ and } s_t} \log p\left(x_t|z_t, s_t\right) \\
&\approx \sum_{t} \log p\left(x_t|s_t\right)
\end{aligned}$$

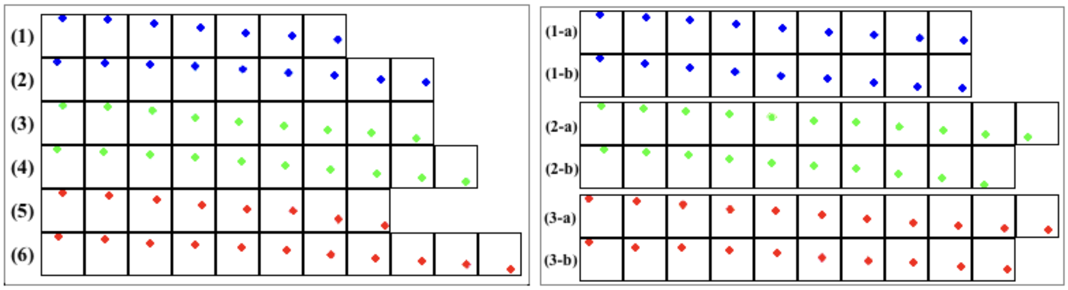
KL Term in ELBO

$$\begin{aligned}
&\log q\left(Z,S,M|X\right)-\log p\left(Z,S,M\right)\\
&=\log q\left(M|X\right)+\log q\left(Z,S|M,X\right)-\log p\left(Z,S,M\right)\\
&=\log q\left(M|X\right)+\log q\left(Z|M,X\right)+\log q\left(S|Z,M,X\right)-\log p\left(Z,S,M\right)\\
&=\log q\left(M|X\right)+\sum_t \log\frac{q\left(z_t|M,X\right)q\left(s_t|z_t,M,X\right)}{p\left(z_t|z_{t-1},m_{t-1}\right)p\left(s_t|s_{t-1},z_t,m_{t-1}\right)p\left(m_t|s_t\right)}\\
&=\log q\left(M|X\right)+\sum_t \log\frac{q\left(z_t|M,X\right)}{p\left(z_t|z_{t-1},m_{t-1}\right)}+\log\frac{q\left(s_t|z_t,M,X\right)}{p\left(s_t|s_{t-1},z_t,m_{t-1}\right)}+\log\frac{1}{p\left(m_t|s_t\right)}
\end{aligned}$$

$$\begin{aligned}
&\sum_M\int_{Z,S} q\left(Z,S,M|X\right)\left[\log q\left(Z,S,M|X\right)-\log p\left(Z,S,M\right)\right]\\
&=\sum_M q\left(M|X\right)\int_{Z,S} q\left(Z,S|M,X\right)\left[\log q\left(Z,S,M|X\right)-\log p\left(Z,S,M\right)\right]\\
&=\sum_M q\left(M|X\right)\int_{Z,S} q\left(Z,S|M,X\right)\left[\log q\left(M|X\right)+\sum_t\log\frac{q\left(z_t|M,X\right)q\left(s_t|z_t,M,X\right)}{p\left(z_t|z_{t-1},m_{t-1}\right)p\left(s_t|s_{t-1},z_t,m_{t-1}\right)p\left(m_t|s_t\right)}\right]\\
&=\sum_M q\left(M|X\right)\left[\log q\left(M|X\right)+\sum_t\int_{Z,S} q\left(Z,S|M,X\right)\log\frac{q\left(z_t|M,X\right)q\left(s_t|z_t,M,X\right)}{p\left(z_t|z_{t-1},m_{t-1}\right)p\left(s_t|s_{t-1},z_t,m_{t-1}\right)p\left(m_t|s_t\right)}\right]\\
&=\sum_M q\left(M|X\right)\left[\log q\left(M|X\right)+\sum_t\int_{z_t,s_t} q\left(z_t,s_t|M,X\right)\log\frac{q\left(z_t|M,X\right)q\left(s_t|z_t,M,X\right)}{p\left(z_t|z_{t-1},m_{t-1}\right)p\left(s_t|s_{t-1},z_t,m_{t-1}\right)p\left(m_t|s_t\right)}\right]\\
&=\sum_M q\left(M|X\right)\left[\log q\left(M|X\right)+\sum_t \mathrm{KL}\left(q'(z_t)||p'(z_t)\right)+\int_{z_t,s_t} q\left(z_t,s_t|M,X\right)\log\frac{q\left(s_t|z_t,M,X\right)}{p\left(s_t|s_{t-1},z_t,m_{t-1}\right)p\left(m_t|s_t\right)}\right]\\
&=\sum_M q\left(M|X\right)\left[\log q\left(M|X\right)+\sum_t \mathrm{KL}\left(q'(z_t)||p'(z_t)\right)+\int_{z_t}\underbrace{q'\left(z_t\right)}_{\text{sample } z_t}\left[\mathrm{KL}\left(q'(s_t)||p'(s_t)\right)-\log p\left(m_t|s_t\right)\right]\right]\\
&\approx\sum_M q\left(M|X\right)\left[\log q\left(M|X\right)+\sum_t \mathrm{KL}\left(q'(z_t)||p'(z_t)\right)+\mathrm{KL}\left(q'(s_t)||p'(s_t)\right)-\log p\left(m_t|s_t\right)\right]\\
&=\sum_M q\left(M|X\right)\left[\log\frac{\prod_t q\left(m_t|X\right)}{\prod_t p\left(m_t|s_t\right)}+\sum_t \mathrm{KL}\left(q'(z_t)||p'(z_t)\right)+\mathrm{KL}\left(q'(s_t)||p'(s_t)\right)\right]\\
&=\sum_{t'}\mathrm{KL}\left(q'\left(m_{t'}\right)||p'\left(m_{t'}\right)\right)+\sum_M\underbrace{q\left(M|X\right)}_{\text{sample } M}\left[\sum_t \mathrm{KL}\left(q'\left(z_t\right)||p'\left(z_t\right)\right)+\mathrm{KL}\left(q'\left(s_t\right)||p'\left(s_t\right)\right)\right]\\
&\approx\sum_t\underbrace{\mathrm{KL}\left(q'\left(m_t\right)||p'\left(m_t\right)\right)}_{\text{sequence decomposer}}+\underbrace{\mathrm{KL}\left(q'\left(z_t\right)||p'\left(z_t\right)\right)}_{\text{temporal abstraction}}+\underbrace{\mathrm{KL}\left(q'\left(s_t\right)||p'\left(s_t\right)\right)}_{\text{observation abstraction}}
\end{aligned}$$

where $p'(m_t)=p(m_t|s_t), p'(z_t)=p\left(z_t|z_{t-1},m_{t-1}\right), p'(s_t)=p\left(s_t|s_{t-1},z_t,m_{t-1}\right)$
$q'(m_t)=q(m_t|X), q'(z_t)=q\left(z_t,s_t|M,X\right), q'(s_t)=q\left(s_t|z_t,M,X\right)$

Appendix E Generated Samples



(a) Subsequences with different z (and different s)

(b) Subsequences with same z (and different s)

Figure 1: Generated subsequences with different content and temporal structure

References

Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D14-1179`.

Clevert, D., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL `http://arxiv.org/abs/1511.07289`.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 10215–10224. Curran Associates, Inc., 2018. URL `http://papers.nips.cc/paper/8224-glow-generative-flow-with-invertible-1x1-convolutions.pdf`.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=ryQu7f-RZ`.