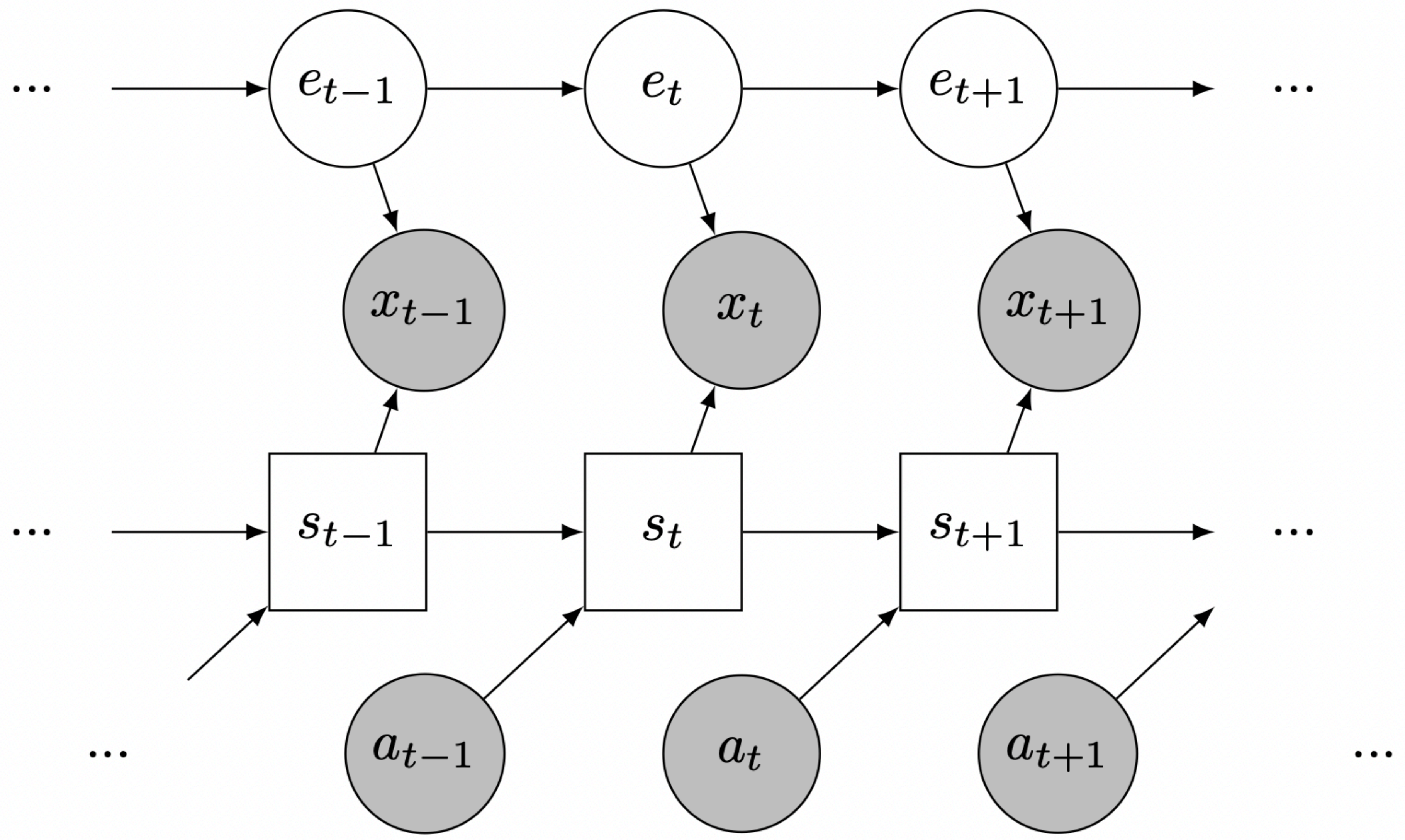
Learning a Fast Mixing Exogenous Block MDP using a Single Trajectory

Alexander Levine[1], Peter Stone[1,2], and Amy Zhang[1]
1: The University of Texas at Austin. 2: Sony AI. Correspondence to alevine0@cs.utexas.edu

Control-Endogenous Representation Learning

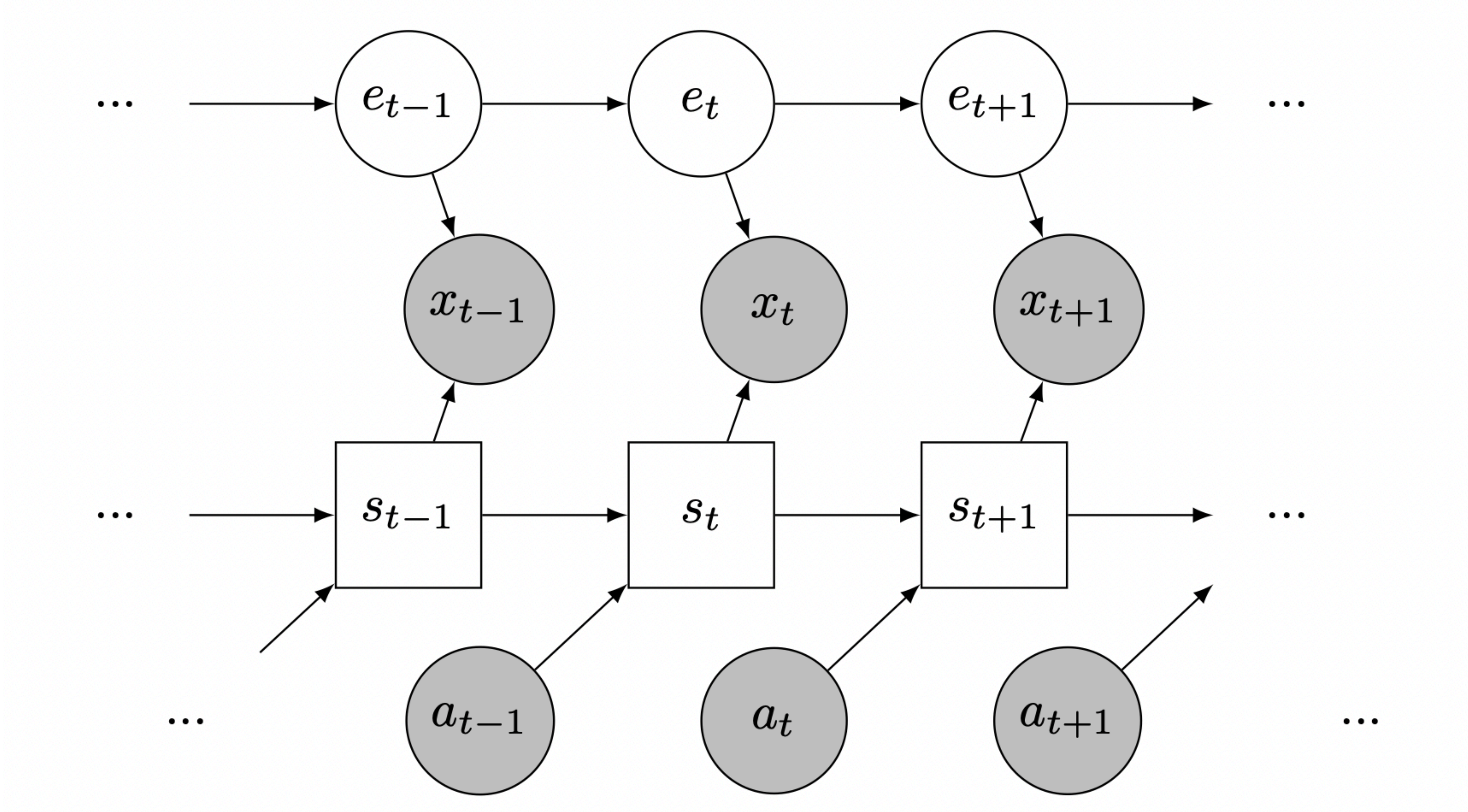
- Observation spaces in control problems can be high-dimensional, and may include factors irrelevant for control.
- These factors may be *time-correlated*
  - Example: leaves blowing/birds flying in the background in a robotic navigation environment.
- To learn to perform downstream tasks efficiently, we need representation learning algorithms that ***ignore control-irrelevant factors***.

Ex-BMDP Model (Efroni et al. 2022b)



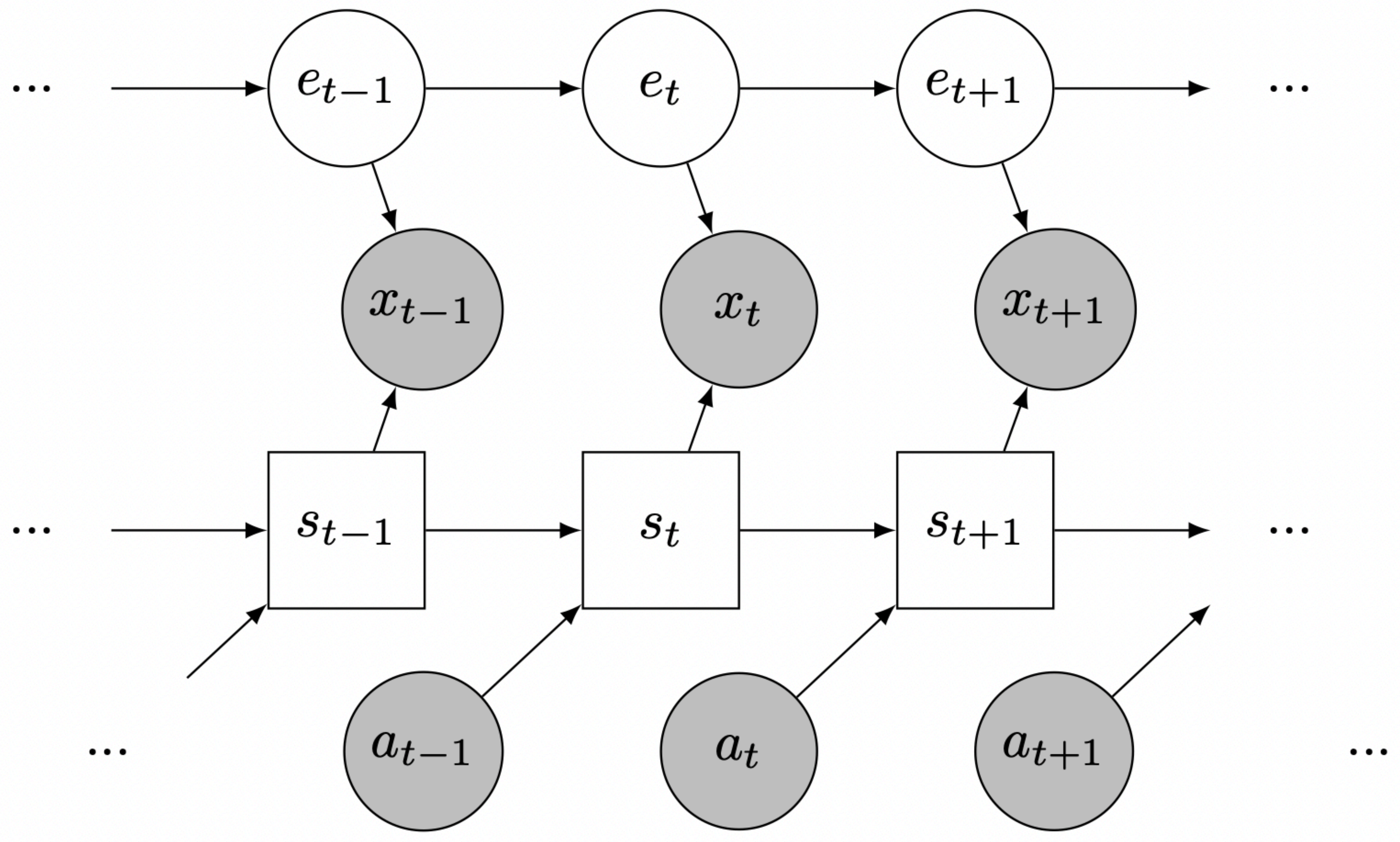
- State $x \in X$ can be factored into:
  - Endogenous state $s \in S$, discrete, evolves deterministically according to actions
  - Exogenous state $e \in \mathcal{E}$, stochastic, independent of actions (***noise***)
- Factorization is *not* known a priori, and s and e are *not* observed.

Ex-BMDP Model (Efroni et al. 2022b)



$$x_{t+1} \sim \mathcal{Q}(x|s_{t+1}, e_{t+1}),$$
$$s_{t+1} = T(s_t, a_t), \quad s_t = \phi(x_t),$$
$$e_{t+1} \sim \mathcal{T}_e(e|e_t)$$

Ex-BMDP Model (Efroni et al. 2022b)

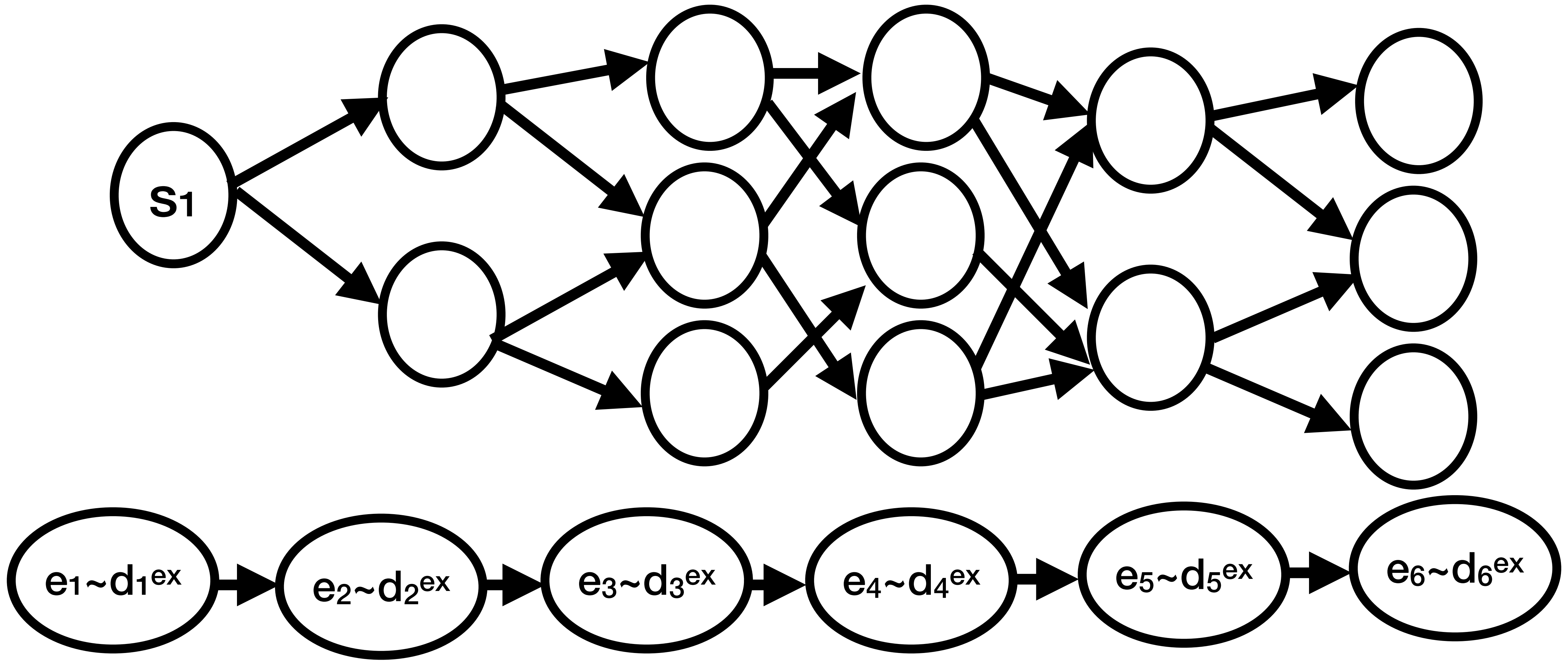


$$x_{t+1} \sim \mathcal{Q}(x|s_{t+1}, e_{t+1}),$$
$$s_{t+1} = T(s_t, a_t), \quad s_t = \phi(x_t),$$
$$e_{t+1} \sim \mathcal{T}_e(e|e_t)$$

Our goal: learn ϕ, with provable sample complexity with *no* direct dependence on |X|, |$\mathcal{E}$|

PPE (Efroni et al. 2022b)

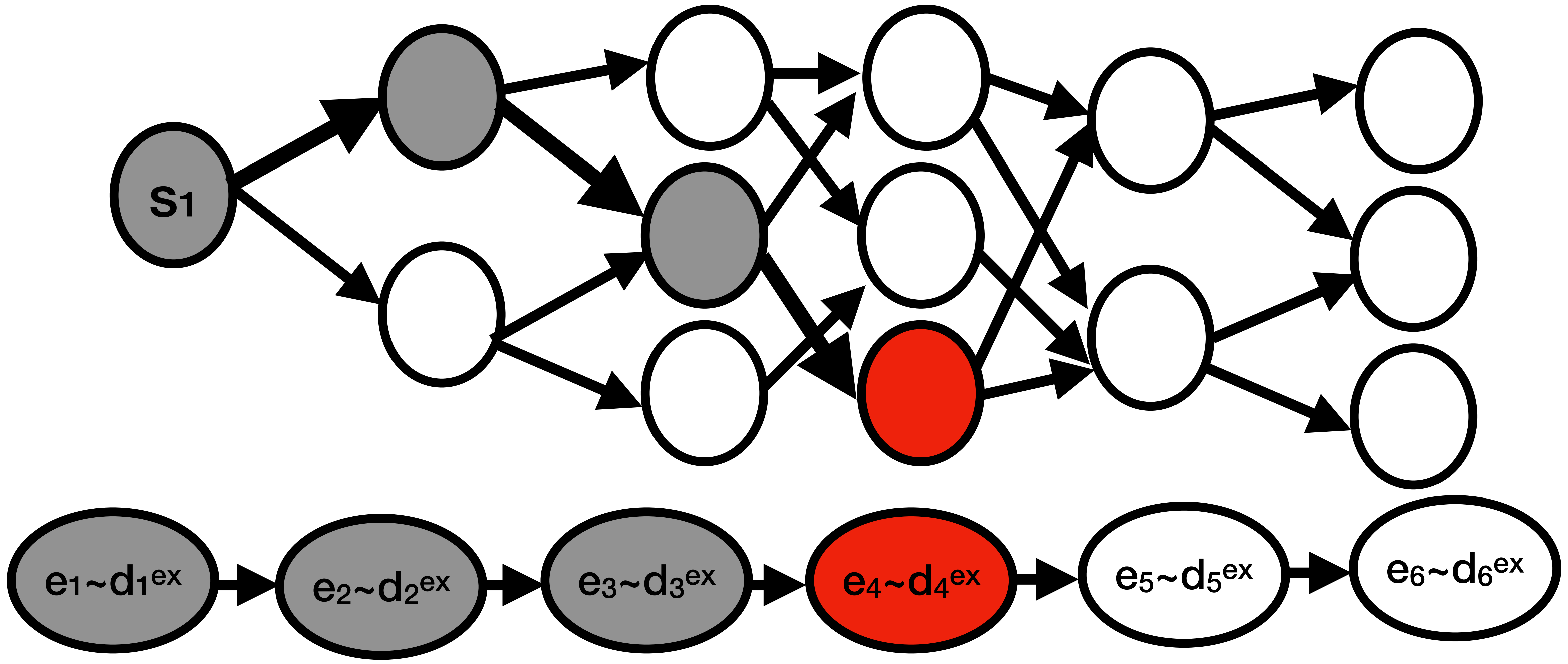
- Efroni et al. consider *episodic* case, with (near) deterministic start state s_1:
  - s_1 is (near) constant; s_t is (near) deterministic function of $a_1,\ldots,a_{t-1}$
  - $e_1 \sim d_1^{ex}$; action-independent dynamics implies $e_t \sim d_t^{ex}$



- IID samples of observations x corresponding to any s can by obtained by simply taking the same sequence of actions $a_1,\ldots,a_{t-1}$ repeatedly.

PPE (Efroni et al. 2022b)

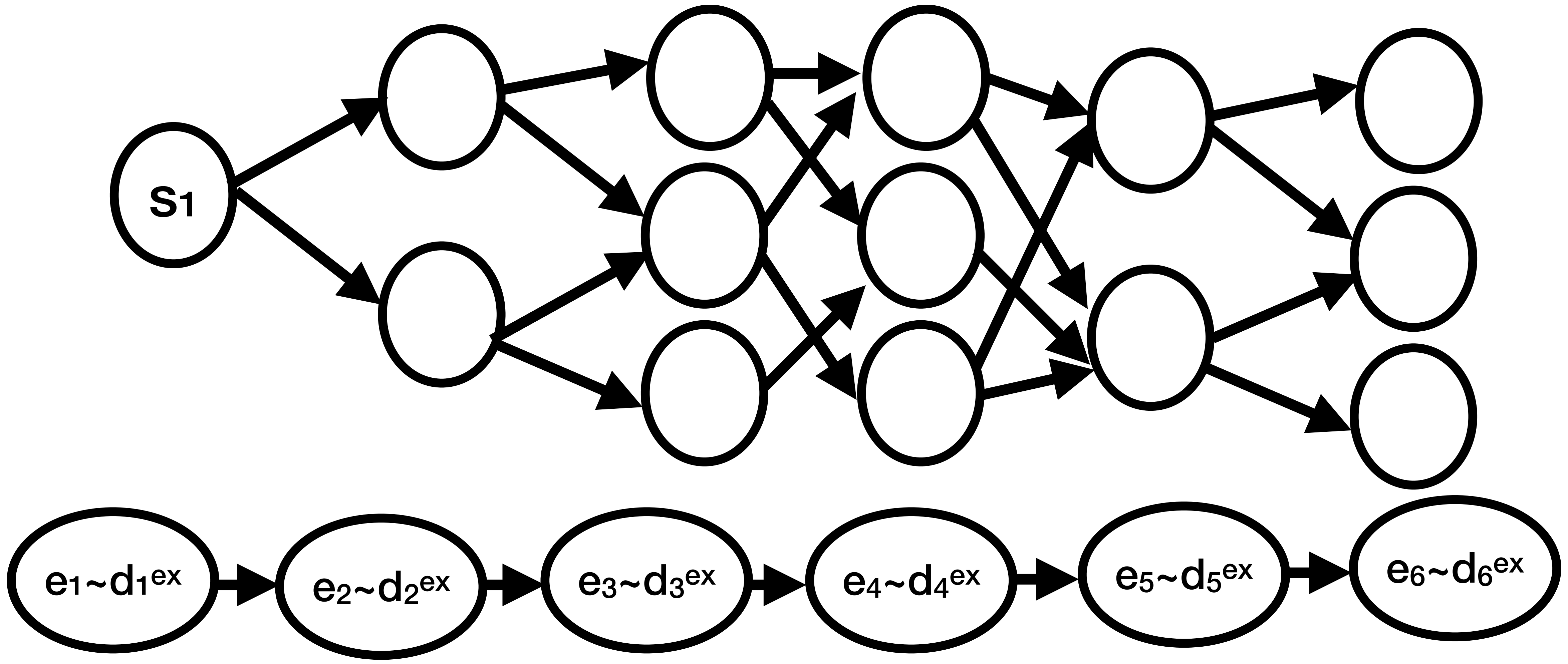
- Efroni et al. consider *episodic* case, with (near) deterministic start state s_1:
  - s_1 is (near) constant; s_t is (near) deterministic function of $a_1,\ldots,a_{t-1}$
  - $e_1 \sim d_1^{ex}$; action-independent dynamics implies $e_t \sim d_t^{ex}$



- IID samples of observations x corresponding to any s can by obtained by simply taking the same sequence of actions $a_1,\ldots,a_{t-1}$ repeatedly.

PPE (Efroni et al. 2022b)

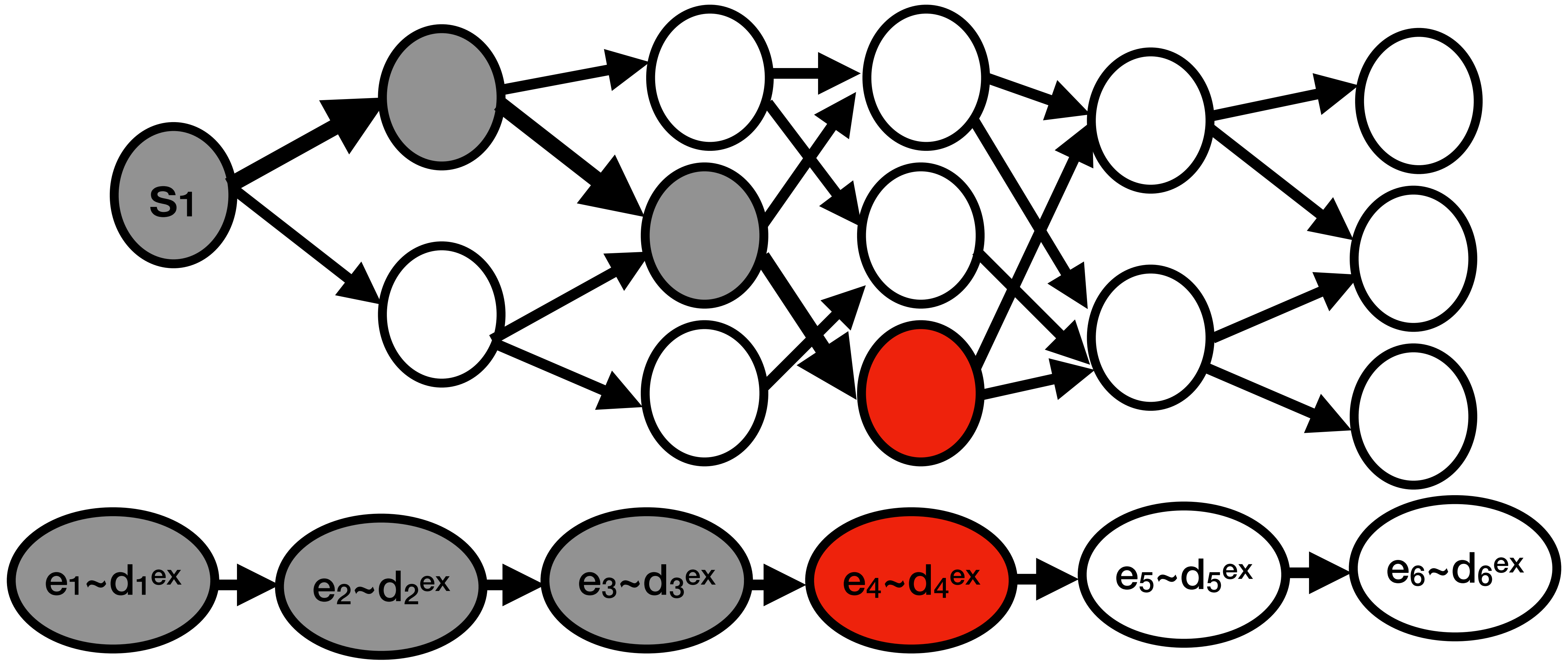
- Efroni et al. consider *episodic* case, with (near) deterministic start state s_1:
  - s_1 is (near) constant; s_t is (near) deterministic function of $a_1,\ldots,a_{t-1}$
  - $e_1 \sim d_1^{ex}$; action-independent dynamics implies $e_t \sim d_t^{ex}$



- IID samples of observations x corresponding to any s can by obtained by simply taking the same sequence of actions $a_1,\ldots,a_{t-1}$ repeatedly.

PPE (Efroni et al. 2022b)

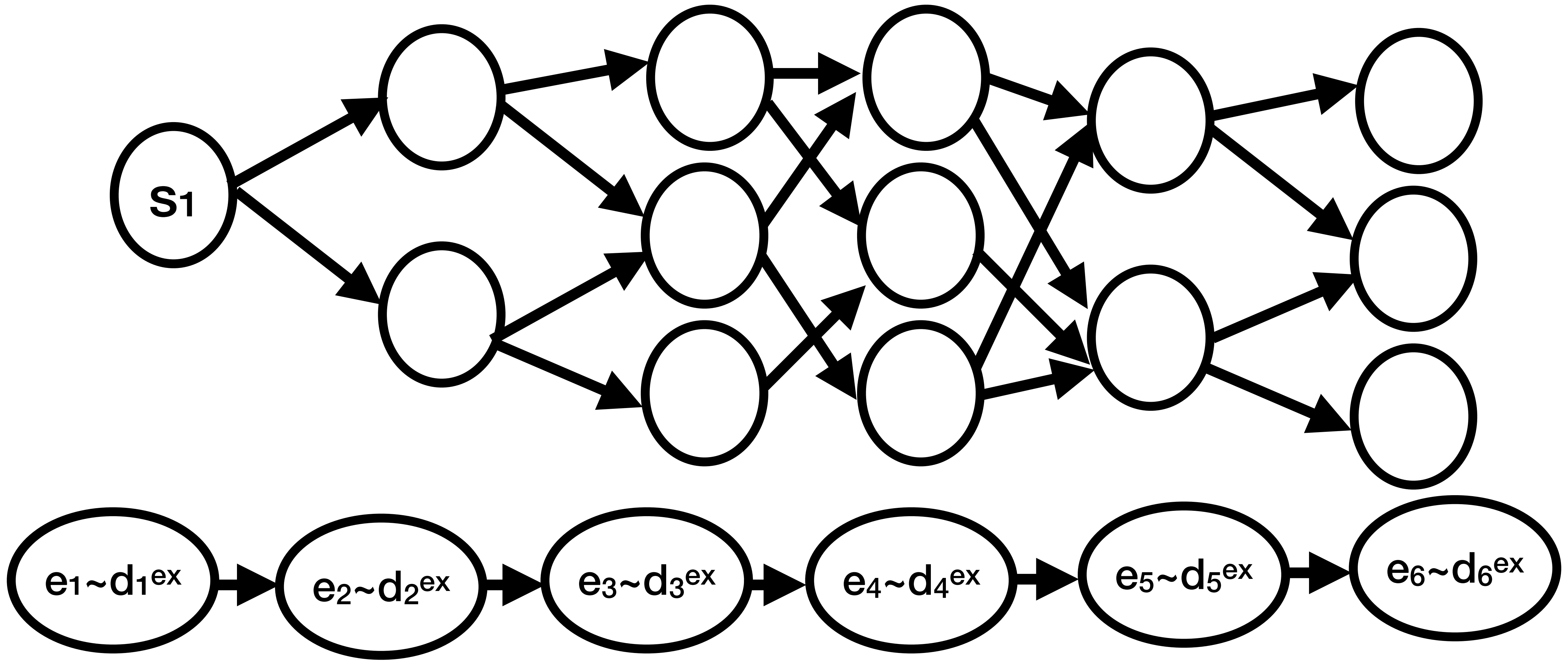
- Efroni et al. consider *episodic* case, with (near) deterministic start state s_1:
  - s_1 is (near) constant; s_t is (near) deterministic function of $a_1,\ldots,a_{t-1}$
  - $e_1 \sim d_1^{ex}$; action-independent dynamics implies $e_t \sim d_t^{ex}$



- IID samples of observations x corresponding to any s can by obtained by simply taking the same sequence of actions $a_1,\ldots,a_{t-1}$ repeatedly.

PPE (Efroni et al. 2022b)

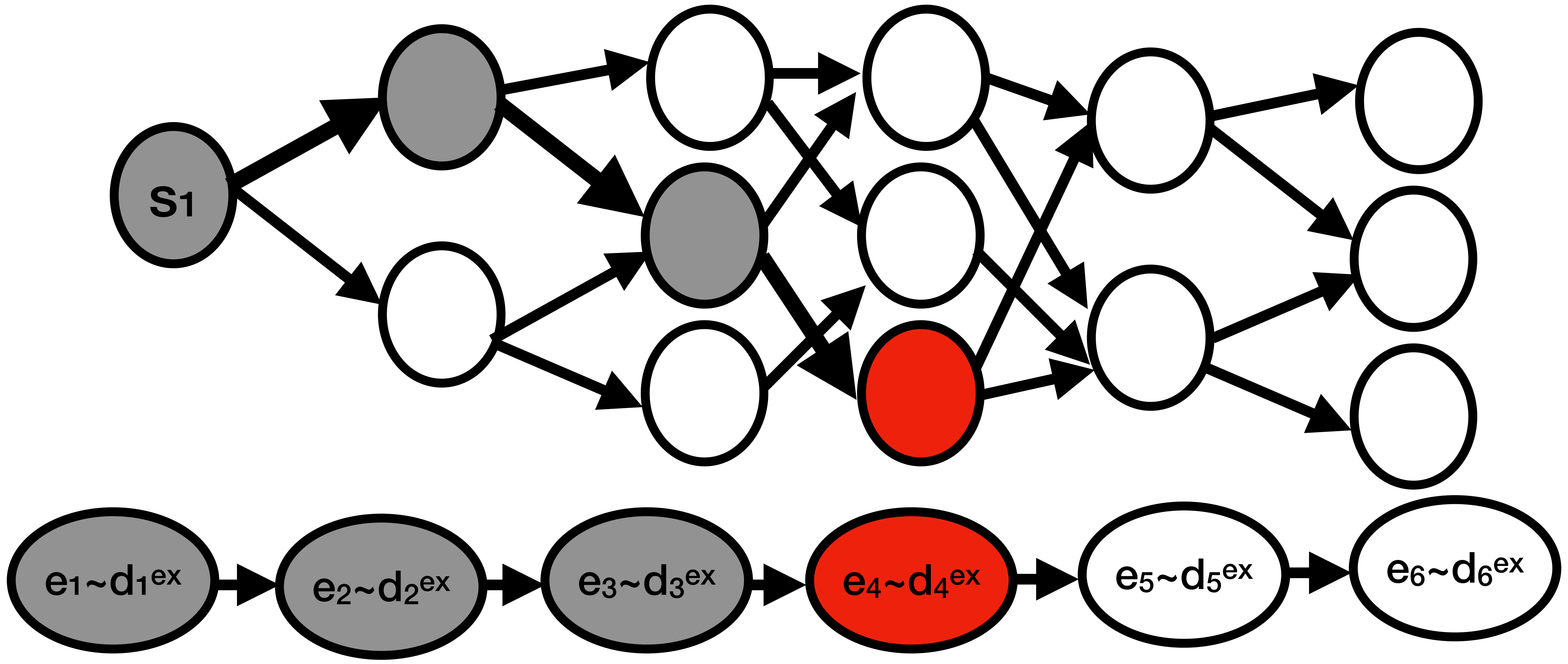
- Efroni et al. consider *episodic* case, with (near) deterministic start state s_1:
  - s_1 is (near) constant; s_t is (near) deterministic function of $a_1,\ldots,a_{t-1}$
  - $e_1 \sim d_1^{ex}$; action-independent dynamics implies $e_t \sim d_t^{ex}$



- IID samples of observations x corresponding to any s can by obtained by simply taking the same sequence of actions $a_1,\ldots,a_{t-1}$ repeatedly.

PPE (Efroni et al. 2022b)

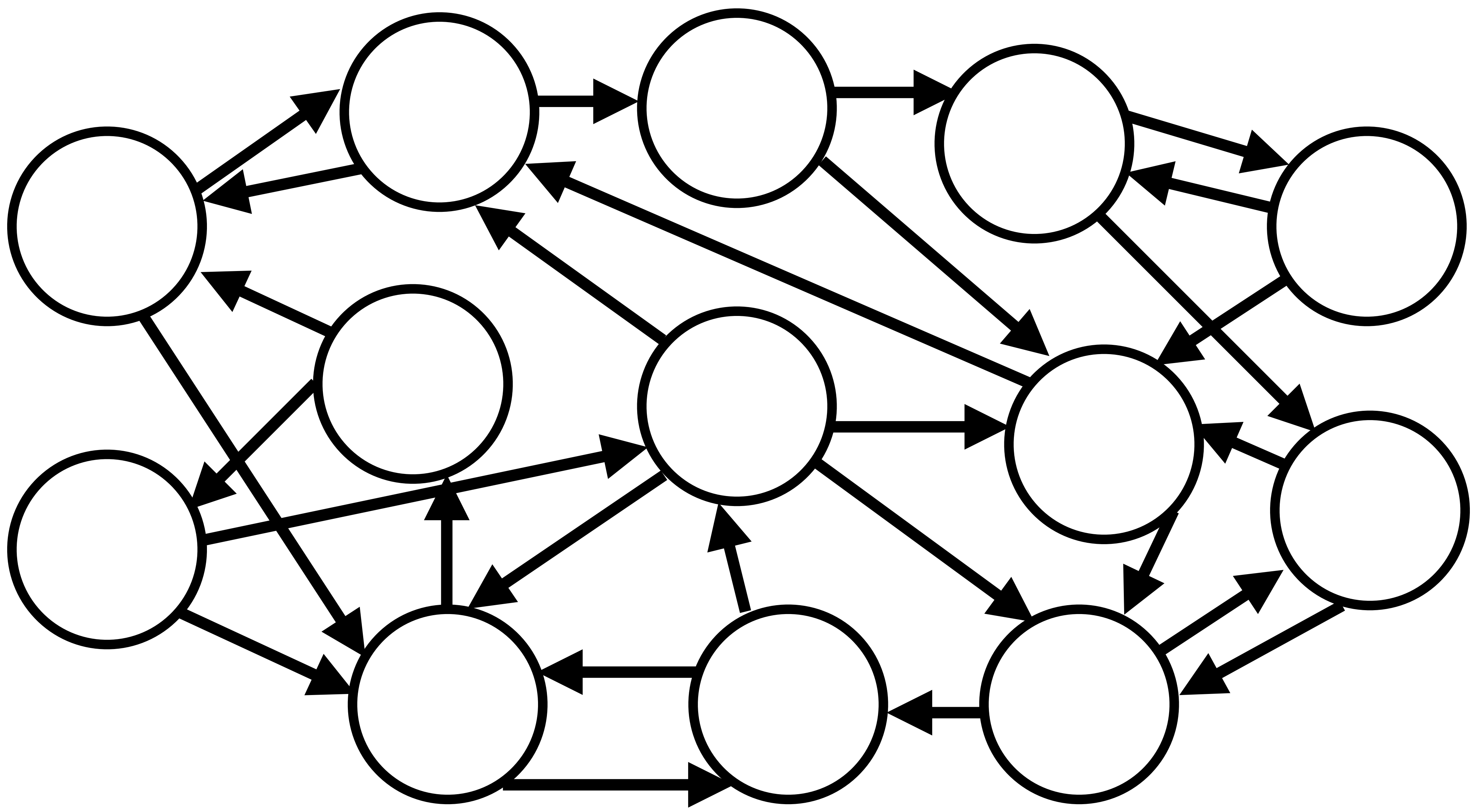
- Efroni et al. consider *episodic* case, with (near) deterministic start state s_1:
  - s_1 is (near) constant; s_t is (near) deterministic function of $a_1, \ldots, a_{t-1}$
  - $e_1 \sim d_1^{ex}$; action-independent dynamics implies $e_t \sim d_t^{ex}$



- IID samples of observations x corresponding to any s can by obtained by simply taking the same sequence of actions $a_1, \ldots, a_{t-1}$ repeatedly.

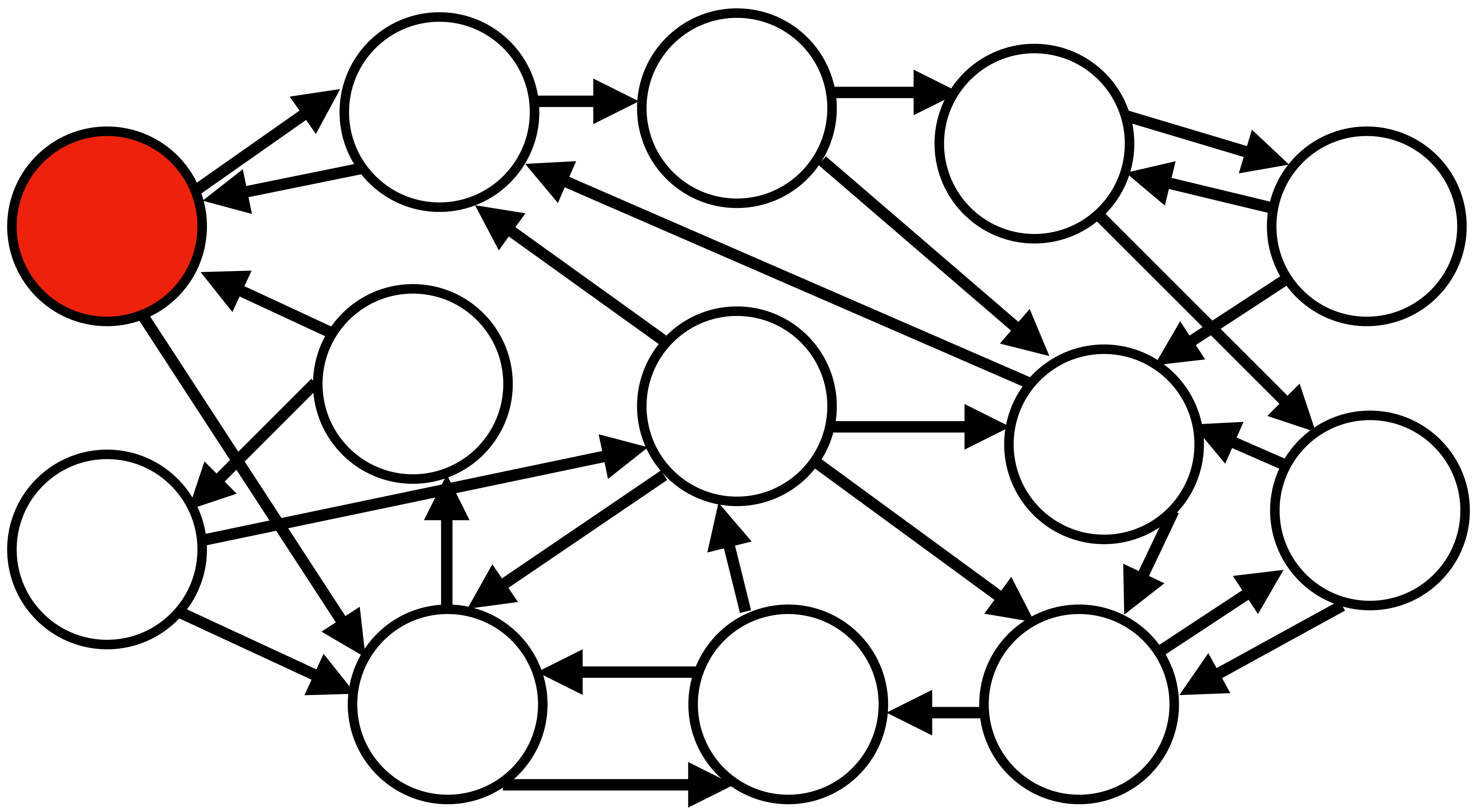
No-Reset Setting

- What if we can't reset to s_1?
  - ***Single-trajectory, infinite horizon***, no-reset setting
  - Not obvious how to get IID sample of any particular latent state
    - In fact, exogenous component is never IID at all



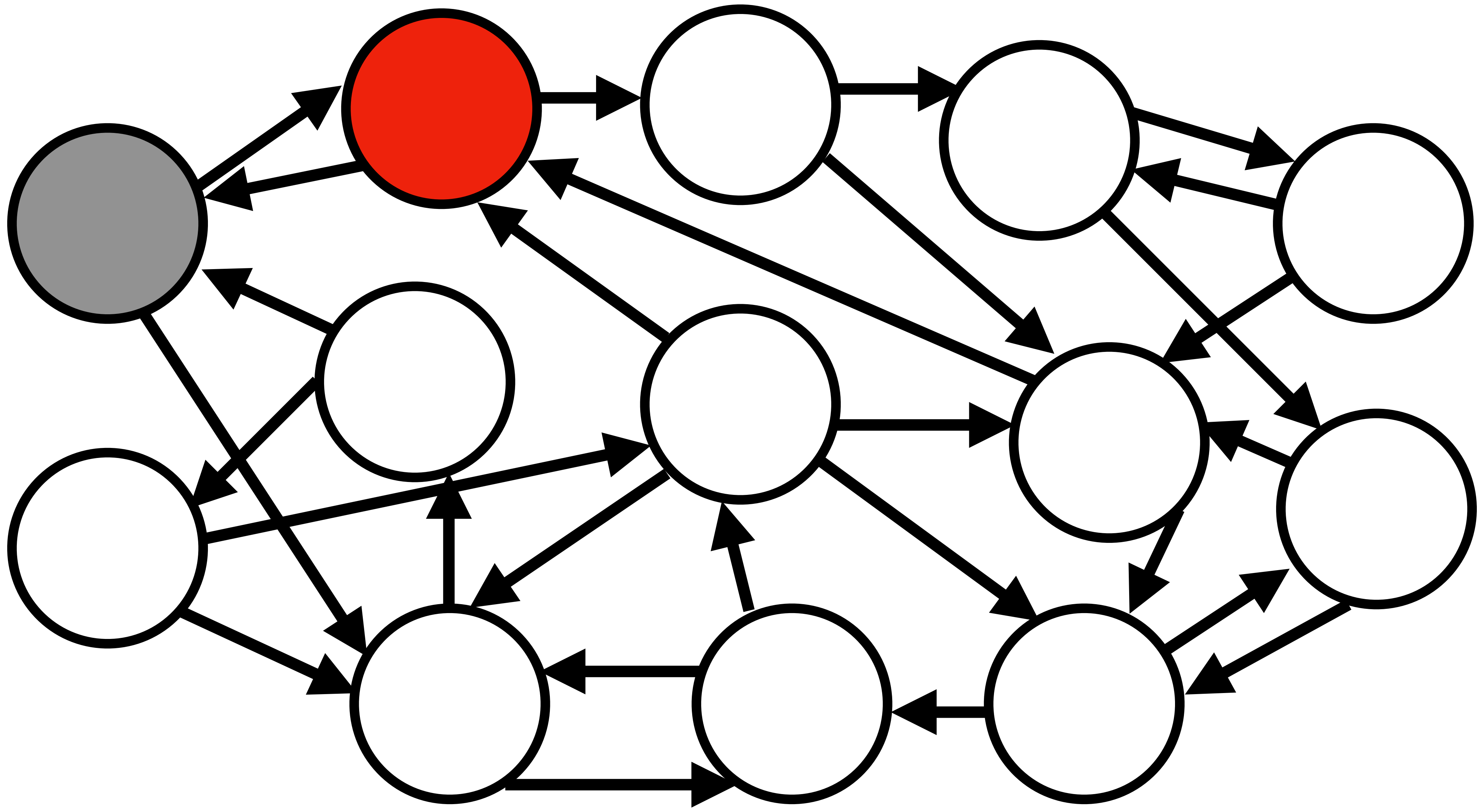
No-Reset Setting

- What if we can't reset to s_1?
  - ***Single-trajectory, infinite horizon***, no-reset setting
  - Not obvious how to get IID sample of any particular latent state
    - In fact, exogenous component is never IID at all



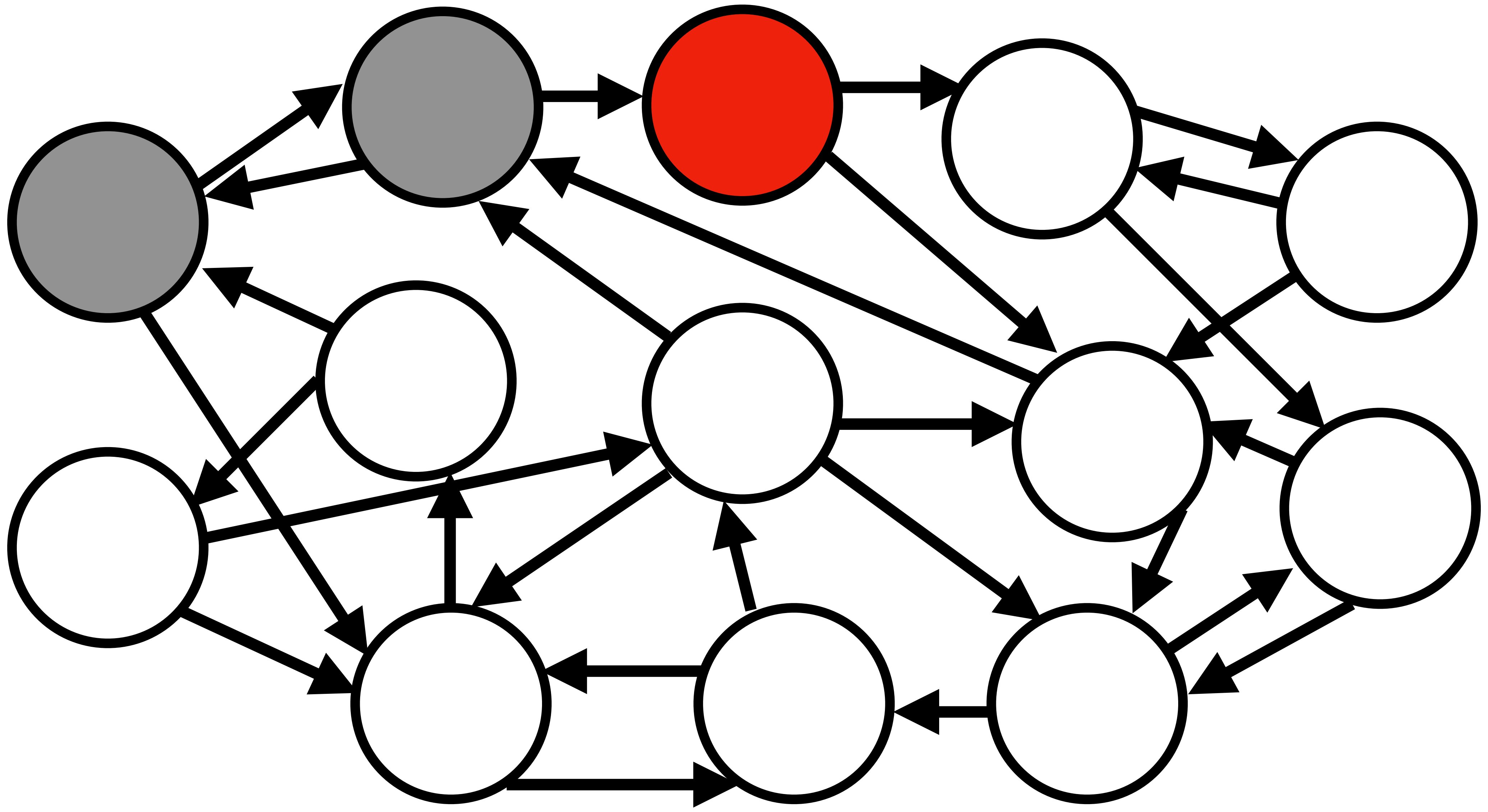
No-Reset Setting

- What if we can't reset to s_1?
  - ***Single-trajectory, infinite horizon***, no-reset setting
  - Not obvious how to get IID sample of any particular latent state
    - In fact, exogenous component is never IID at all



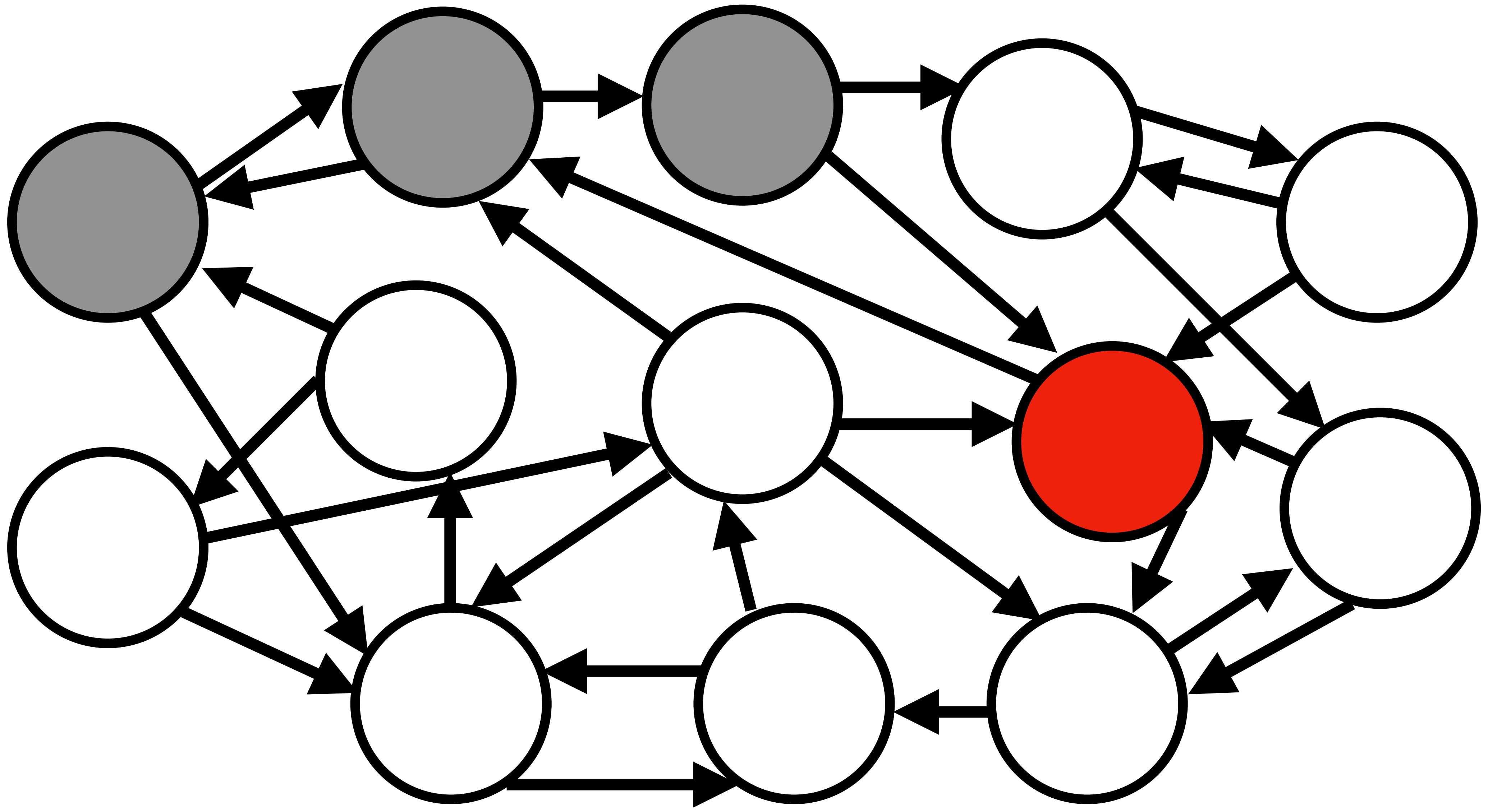
No-Reset Setting

- What if we can't reset to s_1?
  - ***Single-trajectory, infinite horizon***, no-reset setting
  - Not obvious how to get IID sample of any particular latent state
    - In fact, exogenous component is never IID at all



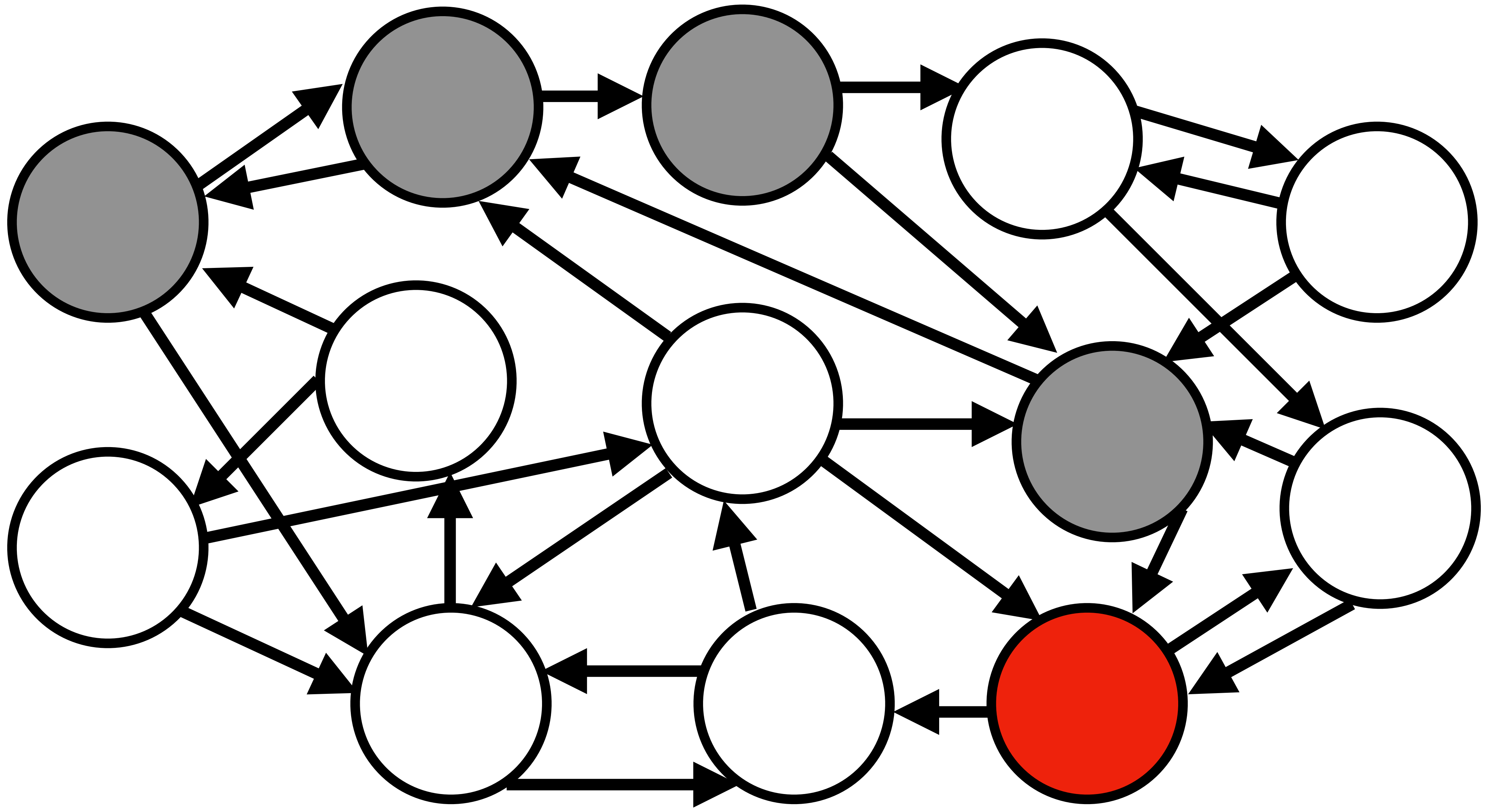
No-Reset Setting

- What if we can't reset to s_1?
  - ***Single-trajectory, infinite horizon***, no-reset setting
  - Not obvious how to get IID sample of any particular latent state
    - In fact, exogenous component is never IID at all



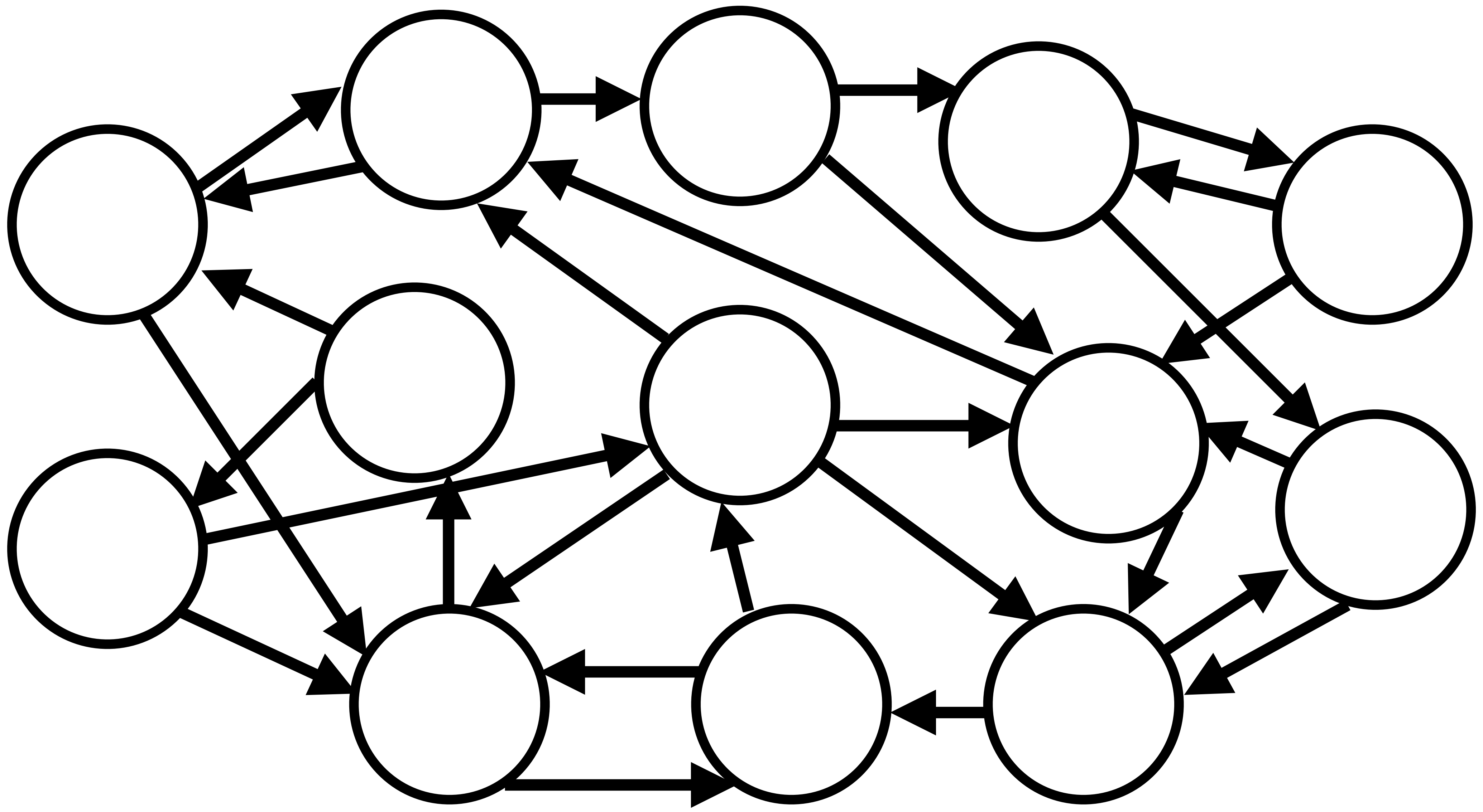
No-Reset Setting

- What if we can't reset to s_1?
  - ***Single-trajectory, infinite horizon***, no-reset setting
  - Not obvious how to get IID sample of any particular latent state
    - In fact, exogenous component is never IID at all



No-Reset Setting

- Prior works:
  - Lamb et al. 2023, Levine et al. 2024:
    - Present ***asymptotically*** correct methods
      - No sample-complexity guarantees given
      - ***The hard part***: how to explore efficiently, if you don't know what state you're currently in?
        - Lamb et al. gives an exploration method, but it's not proven to be sample-efficient, or even asymptotically correct



STEEL Algorithm

- We propose a provably sample-efficient algorithm in this setting
- Additional Assumptions:
  - All latent states s eventually reachable from each other (i.e., no "getting stuck") **— Necessary Assumption**
  - Known upper-bound N on |S|
  - Exogenous state e "mixes fast": **— Necessary Assumption**

$$\forall e \in \mathcal{E}, \ \| \Pr(e_{t+t_{\text{mix}}(\epsilon)} = e' | e_t = e) - \pi_{\mathcal{E}}(e') \|_{\text{TV}} \le \epsilon.$$

$$t_{\text{mix}} := t_{\text{mix}}(1/4)$$

There is a known upper bound $\hat{t}_{\text{mix}}$ on the *mixing time* t_{mix}

STEEL Algorithm

- Sample-Complexity:

$$\mathcal{O}^*\left(ND|\mathcal{S}|^2|\mathcal{A}| \cdot \log\frac{|\mathcal{F}|}{\delta} + |\mathcal{S}||\mathcal{A}|\hat{t}_{mix} \cdot \log\frac{N|\mathcal{F}|}{\delta} + \frac{|\mathcal{S}|^2 D}{\epsilon} \cdot \log\frac{|\mathcal{F}|}{\delta} + \frac{|\mathcal{S}|\hat{t}_{mix}}{\epsilon} \cdot \log\frac{|\mathcal{F}|}{\delta}\right),$$

where $\mathcal{O}^*(f(x)) := \mathcal{O}(f(x)\log(f(x)))$.

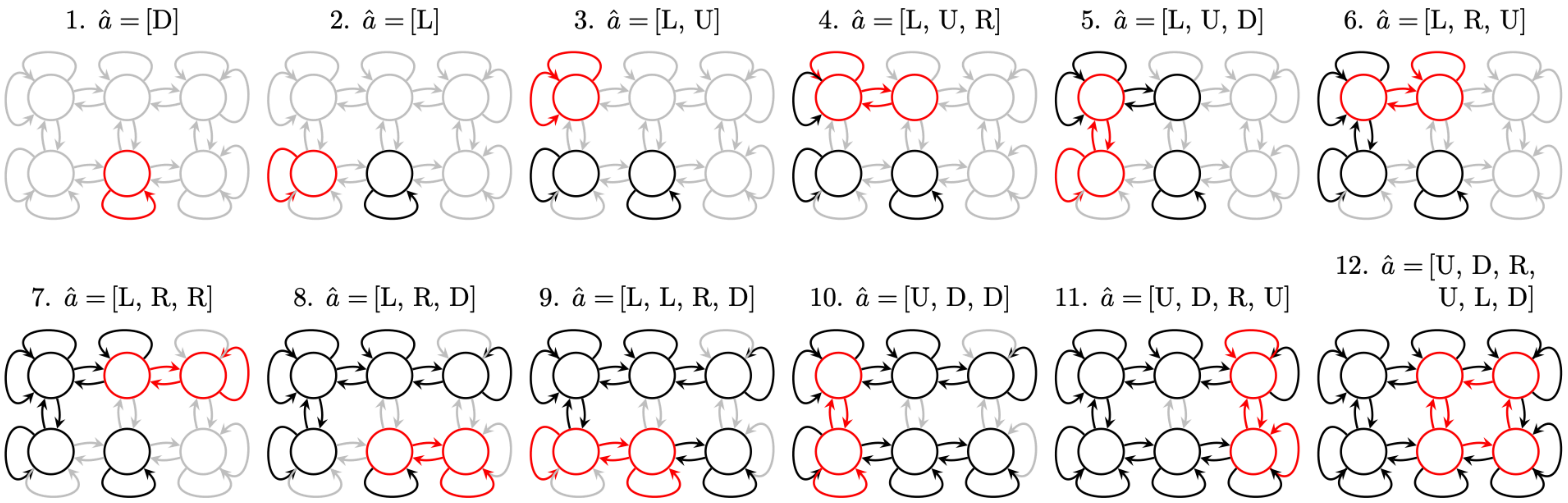
  - *F:* hypothesis class for binary one-versus-rest classification on latent states in S (ϕ is constructed from these classifiers).
  - D: diameter of latent state transition graph T.
  - δ: algorithm failure rate.
  - ε: maximum failure rate of encoder (on *any* latent state s, at stationary distribution of e)

STEEL Algorithm

- Basic idea:
  - Repeating any action sequence a = [a_1,..,a_n] is guaranteed to *eventually* enter a loop of latent states (of length at most n*N)
  - Once we're in a loop, we can "wait out" the exogenous state mixing time to get near-IID samples
  - If we find the period of the cycle, we can get near-IID datasets from all visited latent states

STEEL Algorithm

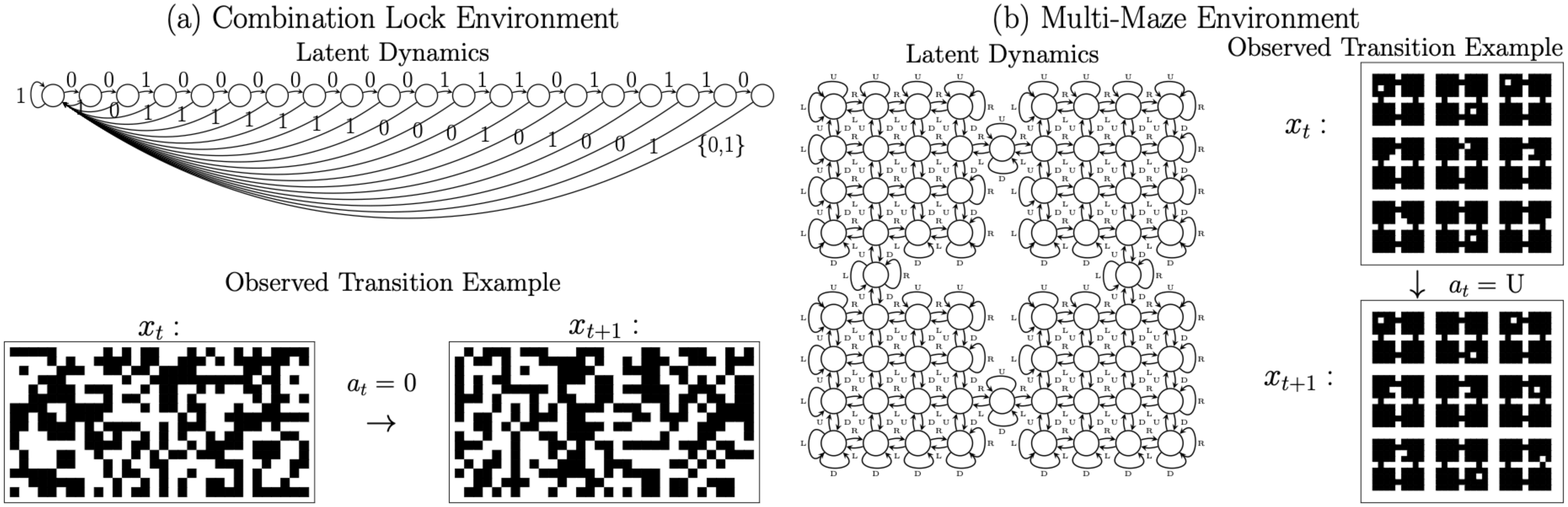
- Dynamics are constructed one cycle at a time



STEEL Algorithm

- Challenges:
  - How to determine period of each cycle?
  - How do we ensure that all states are covered by some cycle?
  - **See paper to learn!**

Results



	Combo. Lock ($K = 20$)	Combo. Lock ($K = 30$)	Combo. Lock ($K = 40$)	Multi-Maze
Fixed Env. Accuracy	20/20	20/20	20/20	20/20
Fixed Env. Steps	1886582±0	4286241±0	7914856±0	41003875±0
Variable Env. Accuracy	20/20	20/20	20/20	20/20
Variable Env. Steps	$2.00{\cdot}10^6$ $\pm 1.28{\cdot}10^5$	$4.78{\cdot}10^6$ $\pm 4.36{\cdot}10^5$	$9.59{\cdot}10^6$ $\pm 1.13{\cdot}10^6$	$4.13{\cdot}10^7$ $\pm 1.11{\cdot}10^6$

References

- Yonathan Efroni, Dipendra Misra, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Provably filtering exogenous distractors using multistep inverse dynamics. ICLR. 2022b.
- Alex Lamb, Riashat Islam, Yonathan Efroni, Aniket Rajiv Didolkar, Dipendra Misra, Dylan J Foster, Lekan P Molu, Rajan Chari, Akshay Krishnamurthy, and John Langford. Guaranteed discovery of control-endogenous latent states with multi-step inverse models. TMLR. 2022.
- Alexander Levine, Peter Stone, and Amy Zhang, Multistep inverse is not all you need. RLC 2024.