A NOTATION

Throughout the paper we use the following notation to indicate mutual information:

$$I(\mathbf{x};\mathbf{z}) := \mathrm{KL}(p(\mathbf{x},\mathbf{z})||p(\mathbf{x})p(\mathbf{z}))$$
$$= \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log\frac{p(\mathbf{x},\mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}\right]. \tag{10}$$

To improve readability we omit the subscript for the expectation. Unless otherwise specified, expectations are computed with respect to the ground-truth distribution $p(\mathbf{x},\mathbf{z})$.

Similarly, we leave the expectation for conditional KL-divergence implicit:

$$\mathrm{KL}(p(\mathbf{x}|\mathbf{z})||q(\mathbf{x}|\mathbf{z})) := \mathbb{E}_{p(\mathbf{x},\mathbf{z})}\left[\log\frac{p(\mathbf{x}|\mathbf{z})}{q(\mathbf{x}|\mathbf{z})}\right]. \tag{11}$$

We will use $\boldsymbol{\epsilon}$ to indicate a stochastic source external to the system, i.e. $I(\boldsymbol{\epsilon};\cdot)=0$ with $\cdot$ as a placeholder for any variable (or combination of variables) in the system (excluding $\boldsymbol{\epsilon}$ itself).

B PROOFS

We start by introducing the assumptions and properties that are used throughout the section. Then, we list proofs for the statements in the main text.

B.1 GENERAL ASSUMPTIONS

As a preliminary step for proving the statements in Section 2, we clarify our general assumptions

(A.1) $\mathbf{z}_t$ is a representation of $\mathbf{x}_t$.
With this statement, we signify that $\mathbf{z}_t$ can be expressed as a noisy function of $\mathbf{x}_t$: $\mathbf{z}_t = f(\mathbf{x}_t,\boldsymbol{\epsilon}_t)$. This implies that $\mathbf{z}_t$ is conditionally independent of any other variable of the system when $\mathbf{x}_t$ is observed: $I(\mathbf{z}_t;\cdot|\mathbf{x}_t,\cdot)=0$, in which $\cdot$ is a placeholder for other variables (or combinations thereof) in the system.

(A.2) $\mathbf{y}_t$ is a representation of $\mathbf{x}_t$.
Analogously to the previous assumption, we assume that the target of interest can be expressed as a noisy function of $\mathbf{x}_t$. Therefore we will assume the same corresponding conditional independence.

(A.3) $[\mathbf{x}_t]_{t=s}^{T}$ form a homogeneous Markov Chain.
This assumption can be expressed in terms of conditional independence between past events $[\mathbf{x}_t]_{t=s}^{m-1}$, and future $\mathbf{x}_{m+1}$ when the current event $\mathbf{x}_m$ is observed:

$$I([\mathbf{x}_t]_{t=s}^{m-1};\mathbf{x}_{m+1}|\mathbf{x}_m)=0,$$

for any $m\in[s+1,T-1]$.

B.2 PROPERTIES

For completeness, here we list properties of mutual information that are used to prove the statements in the following sections. Let $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$ be random variables with some joint distribution $p(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d})$.

(P.1) Non-negativity of (conditional) mutual information.
Mutual information (and conditional mutual information) is non-negative:

$$I(\mathbf{a};\mathbf{b}|\mathbf{c})\geq 0 \tag{12}$$

(P.2) Chain rule of mutual information.
Mutual information (and conditional mutual information) can be factorized as follows:

$$I(\mathbf{ab};\mathbf{c}|\mathbf{d}) = I(\mathbf{a};\mathbf{c}|\mathbf{d}) + I(\mathbf{b};\mathbf{c}|\mathbf{ad})$$
$$= I(\mathbf{b};\mathbf{c}|\mathbf{d}) + I(\mathbf{a};\mathbf{c}|\mathbf{bd}) \tag{13}$$

(P.3) Data processing inequality (DPI).
Mutual information (and conditional mutual information) between two random variables cannot increase by applying functions to either argument (on the left side of the conditioning). In this paper, we will use a slightly more general version of DPI in which we also consider noisy functions (with independent noise):

$$I(\mathbf{a};\mathbf{b}|\mathbf{c}) \geq I(f(\mathbf{a},\boldsymbol{\epsilon});\mathbf{b}|\mathbf{c}) \tag{14}$$

B.3 Autoinformation and Data Processing Inequality

Here we demonstrate that the autoinformation in the original space $AI(\mathbf{x}_t;\tau)$ is an upper bound for the autoinformation of the representation $AI(\mathbf{z}_t;\tau)$.

Statement. *The autoinformation for $\mathbf{x}_t$ upper-bounds the autoinformation for the corresponding representation $\mathbf{z}_t$*

$$AI(\mathbf{x}_t;\tau) \geq AI(\mathbf{z}_t;\tau) \tag{15}$$

Proof.

$$\begin{aligned}
AI(\mathbf{x}_t;\tau) &= I(\mathbf{x}_t;\mathbf{x}_{t+\tau}) \\
&\overset{(P.2)}{=} I(\mathbf{x}_t\mathbf{z}_t;\mathbf{x}_{t+\tau}) - I(\mathbf{z}_t;\mathbf{x}_{t+\tau}|\mathbf{x}_t) \\
&\overset{(A.1)}{=} I(\mathbf{x}_t\mathbf{z}_t;\mathbf{x}_{t+\tau}) \\
&\overset{(P.2)}{=} I(\mathbf{z}_t;\mathbf{x}_{t+\tau}) + I(\mathbf{x}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_t) \\
&\overset{(P.2)}{=} I(\mathbf{z}_t;\mathbf{x}_{t+\tau}\mathbf{z}_{t+\tau}) - I(\mathbf{z}_t;\mathbf{z}_{t+\tau}|\mathbf{x}_{t+\tau}) + I(\mathbf{x}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_t) \\
&\overset{(A.1)}{=} I(\mathbf{z}_t;\mathbf{x}_{t+\tau}\mathbf{z}_{t+\tau}) + I(\mathbf{x}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_t) \\
&\overset{(P.2)}{=} I(\mathbf{z}_t;\mathbf{z}_{t+\tau}) + I(\mathbf{z}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_{t+\tau}) + I(\mathbf{x}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_t) \\
&= AI(\mathbf{z}_t;\tau) + I(\mathbf{z}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_{t+\tau}) + I(\mathbf{x}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_t).
\end{aligned} \tag{16}$$

Using property (P.1), we infer $AI(\mathbf{x}_t;\tau) \geq AI(\mathbf{z}_t;\tau)$ □

Remark 4. *The autoinformation gap upper bounds the amount of information that $\mathbf{x}_t$ and $\mathbf{x}_{t+\tau}$ share whenever one of the two corresponding representations is observed:*

$$AIG(\mathbf{z}_t;\tau) \geq I(\mathbf{x}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_t), \tag{17}$$

and

$$AIG(\mathbf{z}_t;\tau) \geq I(\mathbf{x}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_{t+\tau}) \tag{18}$$

Proof. Re-arranging the terms in equation 16 we can also characterize the autoinformation gap as:

$$\begin{aligned}
AI(\mathbf{x}_t;\tau) - AI(\mathbf{z}_t;\tau) &= I(\mathbf{z}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_{t+\tau}) + I(\mathbf{x}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_t) &(19)\\
&= I(\mathbf{x}_t;\mathbf{z}_{t+\tau}|\mathbf{z}_t) + I(\mathbf{x}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_{t+\tau}). &(20)
\end{aligned}$$

The expression in the second line can be derived by symmetry. Statement 4 follows from (P.1). □

B.4 Autoinformation of sequences

Statement. *A sequence of representations preserves autoinformation at τ if and only if all the pairs of its elements at temporal distance τ preserve autoinformation*

$$AIG([\mathbf{z}_t]_{t=s}^{T};\tau) = 0 \iff AIG(\mathbf{z}_m;\tau) = 0 \quad \forall m \in [s, T-\tau] \tag{21}$$

Proof. We prove the two directions of the implication separately:

- $\Longrightarrow$
  Proof by contradiction. Assume

(T.1) $\exists m \in [s, T-\tau]$ for which the autoinformation in $\mathbf{x}_m$ is strictly larger than the autoinformation of the corresponding representation $\mathbf{z}_m$

$$AI(\mathbf{x}_m;\tau) > AI(\mathbf{z}_m;\tau)$$

.

The autoinformation gap between the two sequences can be written as:

$$\begin{aligned} AIG([\mathbf{z}_t]_{t=s}^{T};\tau) &= \mathbb{E}_t\left[AI(\mathbf{x}_t;\tau) - AI(\mathbf{z}_t;\tau)\right] \\ &\overset{B.3}{\geq} \frac{1}{T-s-\tau+1}\left(AI(\mathbf{x}_m;\tau) - AI(\mathbf{z}_m;\tau)\right) \\ &\overset{(T.1)}{>} 0. \end{aligned} \tag{22}$$

We derived that the autoinformation gap must be strictly positive, which results in a contradiction.

- $\Longleftarrow$
  If we assume that mutual information is the same for all the pairs, clearly their average is also the same.

□

B.5 Autoinformation and Sufficiency

Statement. *Whenever $\mathbf{z}_t$ preserves autoinformation, $\mathbf{z}_t$ is sufficient for any (noisy) function of $\mathbf{x}_{t+\tau}$:*

$$AIG(\mathbf{z}_t;\tau) \iff I(\mathbf{x}_t;\mathbf{y}_{t+\tau}) = I(\mathbf{z}_t;\mathbf{y}_{t+\tau}) \quad \forall \mathbf{y}_{t+\tau} := f(\mathbf{x}_{t+\tau},\epsilon).$$

Proof. We address the two directions of the implication in Lemma 1 separately:

- $\Longrightarrow$
  We start by assuming that $\mathbf{z}_t$ preserves information at τ: $AI(\mathbf{x}_t;\tau) - AI(\mathbf{z}_t;\tau) = 0$

$$\begin{aligned} I(\mathbf{x}_t;\mathbf{y}_{t+\tau}) &\overset{(P.2)}{=} I(\mathbf{x}_t\mathbf{z}_t;\mathbf{y}_{t+\tau}) - I(\mathbf{z}_t;\mathbf{y}_{t+\tau}|\mathbf{x}_t) \\ &\overset{(P.1)}{\leq} I(\mathbf{x}_t\mathbf{z}_t;\mathbf{y}_{t+\tau}) \\ &\overset{(P.2)}{=} I(\mathbf{z}_t;\mathbf{y}_{t+\tau}) + I(\mathbf{x}_t;\mathbf{y}_{t+\tau}|\mathbf{z}_t) \\ &\overset{(P.3)}{\leq} I(\mathbf{z}_t;\mathbf{y}_{t+\tau}) + I(\mathbf{x}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_t) \\ &\overset{4}{=} I(\mathbf{z}_t;\mathbf{y}_{t+\tau}). \end{aligned}$$

  Since we showed $I(\mathbf{x}_t;\mathbf{y}_{t+\tau}) \leq I(\mathbf{z}_t;\mathbf{y}_{t+\tau})$, and using DPI we have $I(\mathbf{x}_t;\mathbf{y}_{t+\tau}) \geq I(\mathbf{z}_t;\mathbf{y}_{t+\tau})$, we must conclude that $I(\mathbf{x}_t;\mathbf{y}_{t+\tau}) = I(\mathbf{z}_t;\mathbf{y}_{t+\tau})$

- $\Longleftarrow$ We prove the second direction of the double implication by contradiction.

(T.1) Let $\mathbf{y}_{t+\tau}$ be a noisy function of $\mathbf{x}_{t+\tau}$ for which $\mathbf{z}_t$ is not sufficient:

$$I(\mathbf{x}_t;\mathbf{y}_{t+\tau}) > I(\mathbf{z}_t;\mathbf{y}_{t+\tau})$$

then

$$\begin{aligned}
AI(\mathbf{z}_t;\tau) &= I(\mathbf{z}_t;\mathbf{z}_{t+\tau}) \\
&\overset{(P.2)}{=} I(\mathbf{z}_t;\mathbf{y}_{t+\tau}\mathbf{z}_{t+\tau}) - I(\mathbf{z}_t;\mathbf{y}_{t+\tau}|\mathbf{z}_{t+\tau}) \\
&\overset{(P.1)}{\leq} I(\mathbf{z}_t;\mathbf{y}_{t+\tau}\mathbf{z}_{t+\tau}) \\
&\overset{(P.2)}{=} I(\mathbf{z}_t;\mathbf{y}_{t+\tau}) + I(\mathbf{z}_t;\mathbf{z}_{t+\tau}|\mathbf{y}_{t+\tau}) \\
&\overset{(T.1)}{<} I(\mathbf{x}_t;\mathbf{y}_{t+\tau}) + I(\mathbf{z}_t;\mathbf{z}_{t+\tau}|\mathbf{y}_{t+\tau}) \\
&\overset{(P.3)}{\leq} I(\mathbf{x}_t;\mathbf{y}_{t+\tau}) + I(\mathbf{x}_t;\mathbf{x}_{t+\tau}|\mathbf{y}_{t+\tau}) \\
&\overset{(P.2)}{=} I(\mathbf{x}_t;\mathbf{x}_{t+\tau}\mathbf{y}_{t+\tau}) \\
&\overset{(P.2)}{=} I(\mathbf{x}_t;\mathbf{x}_{t+\tau}) + I(\mathbf{x}_t;\mathbf{y}_{t+\tau}|\mathbf{x}_{t+\tau}) \\
&\overset{(A.2)}{=} AI(\mathbf{x}_t;\tau).
\end{aligned} \tag{23}$$

We derived that the $AIG(\mathbf{z}_t;\tau) > 0$, which contradicts the premises, concluding the proof.

□

B.6 Markov Property

Statement. *Sequences of representations of a homogeneous Markov Chain that preserve information at some lag time τ also form a homogeneous Markov Chain at temporal resolution τ:*

$$AIG([\mathbf{z}_t]_{t=s}^{T};\tau) = 0 \implies [\mathbf{z}_{s'+k\tau}]_{k=0}^{K} \text{ is a homogeneous Markov Chain}, \tag{24}$$

with $s' \in [s, T-\tau]$, $K \leq \lfloor (T-s')/\tau \rfloor$.

Proof. In order to prove that $[\mathbf{z}_{s'+k\tau}]_{k=0}^{K}$ form a homogeneous Markov Chain, we first show that $[\mathbf{z}_{s'+k\tau}]_{k=0}^{K}$ satisfies the Markov property. This can be shown by upper-bounding the amount of information that the past $[\mathbf{z}_{s'+j\tau}]_{j=0}^{k-1}$ carries about the next representation $\mathbf{z}_{s'+(k+1)\tau}$ whenever the current representation $\mathbf{z}_{s'+k\tau}$ is observed:

$$\begin{aligned}
I([\mathbf{z}_{s'+j\tau}]_{j=0}^{k-1};\mathbf{z}_{s'+(k+1)\tau}|\mathbf{z}_{s'+k\tau}) &\overset{(P.3)}{\leq} I([\mathbf{x}_{s'+j\tau}]_{j=0}^{k-1};\mathbf{x}_{s'+(k+1)\tau}|\mathbf{z}_{s'+k\tau}) \\
&\overset{(P.2)}{=} I([\mathbf{x}_{s'+j\tau}]_{j=0}^{k-1}\mathbf{z}_{s'+k\tau};\mathbf{x}_{s'+(k+1)\tau}) - I(\mathbf{z}_{s'+k\tau};\mathbf{x}_{s'+(k+1)\tau}) \\
&\overset{(P.3)}{\leq} I([\mathbf{x}_{s'+j\tau}]_{j=0}^{k};\mathbf{x}_{s'+(k+1)\tau}) - I(\mathbf{z}_{s'+k\tau};\mathbf{x}_{s'+(k+1)\tau}) \\
&\overset{(P.2)}{=} I(\mathbf{x}_{s'+k\tau};\mathbf{x}_{s'+(k+1)\tau}) - I(\mathbf{z}_{s'+k\tau};\mathbf{x}_{s'+(k+1)\tau}) \\
&\quad + I([\mathbf{x}_{s'+j\tau}]_{j=0}^{k-1};\mathbf{x}_{s'+(k+1)\tau}|\mathbf{x}_{s'+k\tau}) \\
&\overset{(A.3)}{=} I(\mathbf{x}_{s'+k\tau};\mathbf{x}_{s'+(k+1)\tau}) - I(\mathbf{z}_{s'+k\tau};\mathbf{x}_{s'+(k+1)\tau}) \\
&\overset{(P.3)}{\leq} I(\mathbf{x}_{s'+k\tau};\mathbf{x}_{s'+(k+1)\tau}) - I(\mathbf{z}_{s'+k\tau};\mathbf{z}_{s'+(k+1)\tau}) \\
&= AI(\mathbf{x}_{s'+k\tau};\tau) - AI(\mathbf{z}_{s'+k\tau};\tau) \\
&= AIG(\mathbf{z}_{s'+k\tau};\tau) = 0,
\end{aligned} \tag{25}$$

for any $s' \in [s, T-\tau]$, $K \leq \lfloor (T-s')/\tau \rfloor$, and $k \in [1, K-1]$.
Using the results from B.4 and the premise that the autoinformation gap is zero, we can conclude that the conditional mutual information in the previous equation must be zero, and the Markov property holds. Furthermore since both $p(\mathbf{x}_t|\mathbf{x}_{t-\tau})$ and $p(\mathbf{z}_t|\mathbf{x}_t)$ are time-independent, we must conclude that $p(\mathbf{z}_t|\mathbf{z}_{t-\tau})$ must satisfy the same property. Therefore, we conclude that $[\mathbf{z}_{s'+k\tau}]_{k=0}^{K}$ forms a homogeneous Markov Chain. □

B.7 Bounds on the Autoinformation Gap

Statement. *For any $\tau' > \tau > 0$, the autoinformation gap for $\mathbf{z}_t$ at lag time τ' is upper-bounded by the sum of the autoinformation gap for $\mathbf{z}_t$ at lag time τ and the autoinformation gap for $\mathbf{z}_{t+\tau-\tau}$ at lag time τ:*

$$AIG(\mathbf{z}_t;\tau') \leq AIG(\mathbf{z}_t;\tau) + AIG(\mathbf{z}_{t+\tau'-\tau};\tau) \tag{26}$$

Proof. Let $\tau' > \tau > 0$. The autoinformation for $\mathbf{x}_t$ at τ' can be written as:

$$\begin{aligned}
AI(\mathbf{x}_t;\tau') &= I(\mathbf{x}_t;\mathbf{x}_{t+\tau'}) \\
&\overset{(P.2)}{=} I(\mathbf{x}_t\mathbf{z}_t;\mathbf{x}_{t+\tau'}) - I(\mathbf{z}_t;\mathbf{x}_{t+\tau'}|\mathbf{x}_t) \\
&\overset{(P.1)}{\leq} I(\mathbf{x}_t\mathbf{z}_t;\mathbf{x}_{t+\tau'}) \\
&\overset{(P.2)}{=} I(\mathbf{z}_t;\mathbf{x}_{t+\tau'}) + I(\mathbf{x}_t;\mathbf{x}_{t+\tau'}|\mathbf{z}_t) \\
&\overset{(P.3)}{\leq} I(\mathbf{z}_t;\mathbf{x}_{t+\tau'}) + I(\mathbf{x}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_t) \\
&\overset{4}{\leq} I(\mathbf{z}_t;\mathbf{x}_{t+\tau'}) + AIG(\mathbf{z}_t;\tau) \\
&\overset{(P.2)}{=} I(\mathbf{z}_t;\mathbf{x}_{t+\tau'}\mathbf{z}_{t+\tau'}) - I(\mathbf{z}_t;\mathbf{z}_{t+\tau'}|\mathbf{x}_{t+\tau'}) + AIG(\mathbf{z}_t;\tau) \\
&\overset{(P.1)}{\leq} I(\mathbf{z}_t;\mathbf{x}_{t+\tau'}\mathbf{z}_{t+\tau'}) + AIG(\mathbf{z}_t;\tau) \\
&\overset{(P.2)}{=} I(\mathbf{z}_t;\mathbf{z}_{t+\tau'}) + I(\mathbf{z}_t;\mathbf{x}_{t+\tau'}|\mathbf{z}_{t+\tau'}) + AIG(\mathbf{z}_t;\tau) \\
&\overset{(P.3)}{\leq} I(\mathbf{z}_t;\mathbf{z}_{t+\tau'}) + I(\mathbf{x}_t;\mathbf{x}_{t+\tau'}|\mathbf{z}_{t+\tau'}) + AIG(\mathbf{z}_t;\tau) \\
&\overset{(P.3)}{\leq} I(\mathbf{z}_t;\mathbf{z}_{t+\tau'}) + I(\mathbf{x}_{t+\tau'-\tau};\mathbf{x}_{t+\tau'}|\mathbf{z}_{t+\tau'}) + AIG(\mathbf{z}_t;\tau) \\
&\overset{4}{\leq} AI(\mathbf{z}_t;\tau') + AIG(\mathbf{z}_t;\tau) + AIG(\mathbf{z}_{t+\tau'-\tau};\tau).
\end{aligned} \tag{27}$$

Re-arranging the terms, we have:

$$AIG(\mathbf{z}_t;\tau') \leq AIG(\mathbf{z}_t;\tau) + AIG(\mathbf{z}_{t+\tau'-\tau};\tau). \tag{28}$$

Note that whenever $\mathbf{z}_t$ is sampled from the equilibrium, we have $AI(\mathbf{z}_t;\tau') \leq 2AI(\mathbf{z}_t;\tau)$. □

B.8 Slower Information preservation

Statement. *If a sequence of representation preserves autoinformation at lag time τ, then it preserves autoinformation for any $\tau' \geq \tau$:*

$$AIG([\mathbf{z}_t]_{t=s}^{T};\tau) = 0 \implies AIG([\mathbf{z}_t]_{t=s}^{T};\tau') = 0 \tag{29}$$

Proof. Using the result from B.7, we can express the autoinformation gap at τ' as:

$$\begin{aligned}
AIG([\mathbf{z}_t]_{t=s}^{T};\tau') &= \mathbb{E}_{t\sim U(s,T-\tau')}[AIG(\mathbf{z}_t;\tau')] \\
&\overset{B.7}{\leq} \mathbb{E}_{t\sim U(s,T-\tau')}[AIG(\mathbf{z}_t;\tau) + AIG(\mathbf{z}_{t+\tau'-\tau};\tau)] \\
&= AIG([\mathbf{z}_t]_{t=s}^{T-\tau'+\tau};\tau) + AIG([\mathbf{z}_t]_{t=s+\tau'-\tau}^{T};\tau).
\end{aligned} \tag{30}$$

Since both $[\mathbf{z}_t]_{t=s}^{T-\tau'+\tau}$ and $[\mathbf{z}_t]_{t=s+\tau'-\tau}^{T}$ are sub-sequences of $[\mathbf{z}_t]_{t=s}^{T}$, and $AIG([\mathbf{z}_t]_{t=s}^{T};\tau) = 0$, we can infer that the right side of equation 30 must be zero. Furthermore, since $AIG([\mathbf{z}_t]_{t=s}^{T};\tau') \geq 0$, we must conclude $AIG([\mathbf{z}_t]_{t=s}^{T};\tau') = 0$. □

B.9 Latent Simulation error and Autoinformation

Statement. *The average latent simulation error introduced by unfolding K steps using latent simulation starting from $\mathbf{x}_t$ with $t \sim U(s, s+\tau-1)$ is upper-bounded by K times the autoinformation gap for the sequence $[\mathbf{z}_t]_{t=s}^{T}$, with $T = s + (K+1)\tau - 1$:*

$$\mathbb{E}_t \left[\underbrace{\mathrm{KL}(p([\mathbf{y}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t) || p^{LS}([\mathbf{y}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t))}_{\textit{Latent Simulation error for } K \textit{ steps of } \tau \textit{ starting from t}} \right] \leq K \quad \underbrace{AIG([\mathbf{z}_t]_{t=s}^{T}; \tau)}_{\textit{Autoinformation gap for lag time } \tau} \quad . \tag{31}$$

Proof. We start with the following bound:

$$\mathrm{KL}(p([\mathbf{y}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t) || p^{LS}([\mathbf{y}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t)) \leq \mathrm{KL}(p([\mathbf{x}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t) || p^{LS}([\mathbf{x}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t)), \tag{32}$$

which holds because of assumption (A.2) and the data processing inequality. Secondly, we upper-bound the right-most term as a sum of autoinformation:

$$\begin{aligned}
\mathrm{KL}(p([\mathbf{x}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t) || p^{LS}([\mathbf{x}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t)) &\leq \mathrm{KL}(p(\mathbf{z}_t, [\mathbf{x}_{t+k\tau}, \mathbf{z}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t) || p^{LS}(\mathbf{z}_t, [\mathbf{x}_{t+k\tau}, \mathbf{z}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t)) \\
&= \mathbb{E}\left[\log \frac{p_\theta(\mathbf{z}_t|\mathbf{x}_t) \prod_{k=1}^{K} p(\mathbf{x}_{t+k\tau}|\mathbf{x}_{t+(k-1)\tau}) p_\theta(\mathbf{z}_{t+k\tau}|\mathbf{x}_{t+k\tau})}{p_\theta(\mathbf{z}_t|\mathbf{x}_t) \prod_{k=1}^{K} p(\mathbf{z}_{t+k\tau}|\mathbf{z}_{t+(k-1)\tau}) p(\mathbf{x}_{t+k\tau}|\mathbf{z}_{t+k\tau})} \right] \\
&= \sum_{k=1}^{K} \mathbb{E}\left[\log \frac{p(\mathbf{x}_{t+k\tau}|\mathbf{x}_{t+(k-1)\tau})}{p(\mathbf{z}_{t+k\tau}|\mathbf{z}_{t+(k-1)\tau})} \frac{p_\theta(\mathbf{z}_{t+k\tau}|\mathbf{x}_{t+k\tau})}{p(\mathbf{x}_{t+k\tau}|\mathbf{z}_{t+k\tau})} \right] \\
&= \sum_{k=1}^{K} \mathbb{E}\left[\log \frac{p(\mathbf{x}_{t+k\tau}|\mathbf{x}_{t+(k-1)\tau})}{p(\mathbf{z}_{t+k\tau}|\mathbf{z}_{t+(k-1)\tau})} \frac{p(\mathbf{z}_{t+k\tau})}{p(\mathbf{x}_{t+k\tau})} \right] \\
&= \sum_{k=1}^{K} \mathbb{E}\left[\log \frac{p(\mathbf{x}_{t+k\tau}|\mathbf{x}_{t+(k-1)\tau})}{p(\mathbf{x}_{t+k\tau})} \right] + \mathbb{E}\left[\log \frac{p(\mathbf{z}_{t+k\tau})}{p(\mathbf{z}_{t+k\tau}|\mathbf{z}_{t+(k-1)\tau})} \right] \\
&= \sum_{k=0}^{K-\tau} AI(\mathbf{x}_{t+k\tau}; \tau) - AI(\mathbf{z}_{t+k\tau}; \tau) \\
&= \sum_{k=0}^{K-\tau} AIG(\mathbf{z}_{t+k\tau}; \tau)
\end{aligned} \tag{33}$$

Lastly, we consider the average error when $t \sim U(s, s+\tau-1)$

$$\begin{aligned}
\mathbb{E}_t \left[\mathrm{KL}(p([\mathbf{y}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t) || p^{LS}([\mathbf{y}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t)) \right] &= \frac{1}{\tau} \sum_{t=s}^{s+\tau-1} \mathrm{KL}(p([\mathbf{x}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t) || p^{LS}([\mathbf{x}_{t+k\tau}]_{k=1}^{K} |\mathbf{x}_t)) \\
&= \frac{1}{\tau} \sum_{t=s}^{s+\tau-1} \sum_{k=0}^{K-\tau} AIG(\mathbf{z}_{t+k\tau}; \tau) \\
&= \frac{1}{\tau} \sum_{t=s}^{s+K\tau-1} AIG(\mathbf{z}_{t+k\tau}; \tau) \\
&= \frac{K\tau}{\tau} AIG([\mathbf{z}_t]_{t=s}^{s+(K+1)\tau-1}; \tau) \\
&= K\ AIG([\mathbf{z}_t]_{t=s}^{T}; \tau),
\end{aligned} \tag{34}$$

with $T := s + (K+1)\tau - 1$. This concludes the proof. □

B.10 Upper-bound for the Variational Latent Simulation error

Hereby, we outline the steps to obtain the expression reported in Equation 3:

$$\begin{aligned}
&\mathrm{KL}(p([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)||q^{LS}([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s))\\
&\quad=\mathbb{E}\left[\log\frac{p([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)}{p^{LS}([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)}\frac{p^{LS}([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)}{q^{LS}([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)}\right]\\
&\quad=\mathrm{KL}(p([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)||p^{LS}([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s))+\mathbb{E}\left[\log\frac{p^{LS}([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)}{q^{LS}([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)}\right].
\end{aligned}\tag{35}$$

Focusing on the second term:

$$\begin{aligned}
\mathbb{E}\left[\log\frac{p^{LS}([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)}{q^{LS}([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)}\right]&\le\mathbb{E}\left[\log\frac{p^{LS}([\mathbf{y}_{s+k\tau},\mathbf{z}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)}{q^{LS}([\mathbf{y}_{s+k\tau},\mathbf{z}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)}\right]\\
&=\mathbb{E}\left[\log\frac{\prod_{k=1}^{K}p(\mathbf{z}_{s+k\tau}|\mathbf{z}_{s+(k-1)\tau})p(\mathbf{y}_{s+k\tau}|\mathbf{z}_{s+k\tau})}{\prod_{k=1}^{K}q_\phi(\mathbf{z}_{s+k\tau}|\mathbf{z}_{s+(k-1)\tau})q_\psi(\mathbf{y}_{s+k\tau}|\mathbf{z}_{s+k\tau})}\right]\\
&=\sum_{k=1}^{K}\mathbb{E}\left[\log\frac{p(\mathbf{z}_{s+k\tau}|\mathbf{z}_{s+(k-1)\tau})}{q_\phi(\mathbf{z}_{s+k\tau}|\mathbf{z}_{s+(k-1)\tau})}\right]+\mathbb{E}\left[\log\frac{p(\mathbf{y}_{s+k\tau}|\mathbf{z}_{s+k\tau})}{q_\psi(\mathbf{y}_{s+k\tau}|\mathbf{z}_{s+k\tau})}\right]\\
&=\sum_{k=1}^{K}\mathrm{KL}(p(\mathbf{z}_{s+k\tau}|\mathbf{z}_{s+(k-1)\tau})||q_\phi(\mathbf{z}_{s+k\tau}|\mathbf{z}_{s+(k-1)\tau}))\\
&\qquad+\mathrm{KL}(p(\mathbf{y}_{s+k\tau}|\mathbf{z}_{s+k\tau})||q_\psi(\mathbf{y}_{s+k\tau}|\mathbf{z}_{s+k\tau})).
\end{aligned}\tag{36}$$

B.11 Superfluos Information decomposition

The total amount of information that a representation $\mathbf{z}_t$ contains about the original data $\mathbf{x}_t$ can be de-composed using the chain rule of mutual information as follows:

$$\begin{aligned}
I(\mathbf{x}_t;\mathbf{z}_t)&\overset{(P.2)}{=}I(\mathbf{x}_t\mathbf{z}_{t-\tau};\mathbf{z}_t)-I(\mathbf{z}_t;\mathbf{z}_{t-\tau}|\mathbf{x}_t)\\
&\overset{(A.1)}{=}I(\mathbf{x}_t\mathbf{z}_{t-\tau};\mathbf{z}_t)\\
&\overset{(P.2)}{=}\underbrace{I(\mathbf{z}_{t-\tau};\mathbf{z}_t)}_{\text{Autoinformation}}+\underbrace{I(\mathbf{x}_t;\mathbf{z}_t|\mathbf{z}_{t-\tau})}_{\text{Superfluous Information}}\quad.
\end{aligned}\tag{37}$$

We can further factorize superfluous information by considering the immediate past $\mathbf{x}_{t-1}$ as follows:

$$\begin{aligned}
\underbrace{I(\mathbf{x}_t;\mathbf{z}_t|\mathbf{z}_{t-\tau})}_{\text{Superfluous Information}}&\overset{(P.2)}{=}I(\mathbf{x}_t\mathbf{x}_{t-1};\mathbf{z}_t|\mathbf{z}_{t-\tau})-I(\mathbf{z}_{t-1};\mathbf{z}_t|\mathbf{x}_t)\\
&\overset{(A.1)}{=}I(\mathbf{x}_t\mathbf{x}_{t-1};\mathbf{z}_t|\mathbf{z}_{t-\tau})\\
&\overset{(P.2)}{=}I(\mathbf{x}_{t-1};\mathbf{z}_t|\mathbf{z}_{t-\tau})+I(\mathbf{x}_t;\mathbf{z}_t|\mathbf{x}_{t-1}\mathbf{z}_{t-\tau})\\
&\overset{(P.2)}{=}I(\mathbf{x}_{t-1};\mathbf{z}_t|\mathbf{z}_{t-\tau})+I(\mathbf{x}_t\mathbf{z}_{t-\tau};\mathbf{z}_t|\mathbf{x}_{t-1})-I(\mathbf{z}_{t-\tau};\mathbf{z}_t|\mathbf{x}_{t-1},\mathbf{x}_t)\\
&\overset{(A.1)}{=}I(\mathbf{x}_{t-1};\mathbf{z}_t|\mathbf{z}_{t-\tau})+I(\mathbf{x}_t\mathbf{z}_{t-\tau};\mathbf{z}_t|\mathbf{x}_{t-1})\\
&\overset{(P.2)}{=}I(\mathbf{x}_{t-1};\mathbf{z}_t|\mathbf{z}_{t-\tau})+I(\mathbf{x}_t;\mathbf{z}_t|\mathbf{x}_{t-1})+I(\mathbf{z}_t;\mathbf{z}_{t-\tau}|\mathbf{x}_{t-1})\\
&=\underbrace{I(\mathbf{x}_{t-1};\mathbf{z}_t|\mathbf{z}_{t-\tau})}_{\text{Dynamic Information faster than }\tau}+\underbrace{I(\mathbf{x}_t;\mathbf{z}_t|\mathbf{x}_{t-1})}_{\text{Time-independent information}}\quad,
\end{aligned}\tag{38}$$

in which the last step follows from:

$$0\overset{(P.1)}{\le}I(\mathbf{z}_t;\mathbf{z}_{t-\tau}|\mathbf{x}_{t-1})\overset{(A.1)}{\le}I(\mathbf{x}_t;\mathbf{x}_{t-\tau}|\mathbf{x}_{t-1})\overset{(A.3)}{=}0.\tag{39}$$

Note that $I(\mathbf{x}_{t-1};\mathbf{z}_t|\mathbf{z}_{t-\tau})$ refers to the information that $\mathbf{z}_t$ conveys about the immediate past $\mathbf{x}_{t-1}$ when the past representation $\mathbf{z}_{t-\tau}$ is observed. This quantity is positive whenever $\mathbf{z}_t$ contains information regarding processes that are faster than τ, i.e. are not predictable from the past representation $\mathbf{z}_{t-\tau}$ but can be inferred from $\mathbf{z}_t$. The second term $I(\mathbf{x}_t;\mathbf{z}_t|\mathbf{x}_{t-1})$ refers to the information that $\mathbf{z}_t$ contains about processes that appear time-independent at the highest available time-resolution ($\tau = 1$). This component includes both time-independent noise and other time-dependent processes that appear uncorrelated at the observed temporal resolution. These two last components are indistinguishable without having access to higher-resolution sequences.

C Computation and Approximations

C.1 A two-step minimization procedure

Consider the terms on the right side of expression 3. We use

$$\mathcal{L}^{LS}(\theta) := \mathrm{KL}(p([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)||p^{LS}([\mathbf{y}_{s+k\tau}]_{k=1}^{K}|\mathbf{x}_s)) \tag{40}$$

$$\mathcal{L}^{T}(\theta,\phi) := \sum_{k=1}^{K}\mathrm{KL}(p(\mathbf{z}_{s+k\tau}|\mathbf{z}_{s+(k-1)\tau})||q_\phi(\mathbf{z}_{s+k\tau}|\mathbf{z}_{s+(k-1)\tau})) \tag{41}$$

$$\mathcal{L}^{P}(\theta,\psi) := \sum_{k=1}^{K}\mathrm{KL}(p(\mathbf{y}_{s+k\tau}|\mathbf{z}_{s+k\tau})||q_\psi(\mathbf{y}_{s+k\tau}|\mathbf{z}_{s+k\tau})) \tag{42}$$

for notation brevity to underline the dependencies with the parameters θ, ϕ, ψ for the encoder, variational transition, and variational predictive distributions respectively. The joint optimization can then be written as:

$$\begin{aligned}\min_{\theta,\phi,\psi}\mathcal{L}^{LS}(\theta)+\mathcal{L}^{T}(\theta,\phi)+\mathcal{L}^{P}(\theta,\psi) &= \min_{\theta}\left[\mathcal{L}^{LS}(\theta)+\min_{\phi}\mathcal{L}^{T}(\theta,\phi)+\min_{\psi}\mathcal{L}^{P}(\theta,\psi)\right]\\ &\leq \mathcal{L}^{LS}(\hat\theta)+\min_{\phi}\mathcal{L}^{T}(\hat\theta,\phi)+\min_{\psi}\mathcal{L}^{P}(\hat\theta,\psi).\end{aligned} \tag{43}$$

With $\hat\theta := \arg\min_\theta \mathcal{L}^{LS}(\theta)$.

The upper bound in equation 43 is still tight for flexible variational transition and prediction distribution. For a fixed $\hat\theta$, the variational transition and predictive gaps depend uniquely on the variational parameters ϕ and ψ which can be optimized by minimizing the negative log-likelihood:

$$\arg\min_{\phi}\mathcal{L}^{T}(\hat\theta,\phi) = \arg\min_{\phi}\sum_{k=1}^{K}\mathbb{E}[-\log q_\phi(\mathbf{z}_{s+k\tau}|\mathbf{z}_{s+(k-1)\tau})] \tag{44}$$

$$\arg\min_{\psi}\mathcal{L}^{P}(\hat\theta,\psi) = \arg\min_{\psi}\sum_{k=1}^{K}\mathbb{E}[-\log q_\psi(\mathbf{y}_{s+k\tau}|\mathbf{z}_{s+k\tau})]. \tag{45}$$

C.2 Contrastive Learning on Markov Processes

Consider the expression reported in equation 6:

$$\begin{aligned}\mathcal{L}_{\text{InfoNCE}}^{\text{T-InfoMax}}([\boldsymbol{x}_t]_{t=s}^{T},\tau;\theta,\xi) &:= -\mathbb{E}\left[\log\frac{e^{F_\xi(\boldsymbol{z}_t,\boldsymbol{z}_{t-\tau})}}{\mathbb{E}_{\boldsymbol{z}'_t\sim p(\mathbf{z}_t)}\left[e^{F_\xi(\boldsymbol{z}'_t,\boldsymbol{z}_{t-\tau})}\right]}\right] &(46)\\ &\approx -\frac{1}{B}\sum_{i=1}^{B}\log\frac{e^{F_\xi(\boldsymbol{z}_{t_i},\boldsymbol{z}_{t_i-\tau})}}{\frac{1}{B}\sum_{j=1}^{B}e^{F_\xi(\boldsymbol{z}_{t_j},\boldsymbol{z}_{t_i-\tau})}}. &(47)\end{aligned}$$

Focusing on the denominator in equation 46, we note that estimating the partition function would require sampling $\boldsymbol{z}'_t$ from $p(\mathbf{z}_t)$. If the dataset consists of multiple trajectories $\left[\boldsymbol{x}_t^{(i)}\right]_{t=s_i}^{T_i} \overset{N}{\sim} p([\mathbf{x}_t]_{t=s}^{T})$, then this would require considering the representation of $\boldsymbol{x}_t^{(i)}$ from multiple trajectories at the given time t. Since we are considering time-independent homogeneous processes, even when the dataset consists of a single trajectory $[\boldsymbol{x}_t]_{t=s}^T$, we can approximate samples from $p(\mathbf{x}_t)$ by considering any $\boldsymbol{x}_{t'}$ in the same sequence, with $t' \sim U(s, T)$. This approximation is accurate whenever $p(\mathbf{x}_t)$ approaches the equilibrium distribution and the trajectory $[\boldsymbol{x}_t]_{t=s}^T$ is long enough to obtain de-correlated samples. In case multiple trajectories are available at training time, this approach would benefit from creating mini-batches of inputs $\boldsymbol{x}_t^{(i)}$ (and corresponding representations $\boldsymbol{z}_t^{(i)}$) that are sampled from distinct trajectories:

$$\mathcal{L}_{\text{InfoNCE}}^{\text{T-InfoMax}}(\left\{\left[\boldsymbol{x}_t^{(i)}\right]_{t=s_i}^{T_i}\right\}_{i=1}^N, \tau; \theta, \xi) \approx -\frac{1}{B}\sum_{i=1}^B \log \frac{e^{F_\xi(\boldsymbol{z}_{t_i}^{(i)}, \boldsymbol{z}_{t_i-\tau}^{(i)})}}{\frac{1}{B}\sum_{j=1}^B e^{F_\xi(\boldsymbol{z}_{t_j}^{(j)}, \boldsymbol{z}_{t_i-\tau}^{(i)})}}. \tag{48}$$

C.3 Superfluous information upper-bound

Computing superfluous information would require access to the true transition distribution $p(\mathbf{z}_t|\mathbf{z}_{t-\tau})$. Using standard variational inference, we can define a variational upper-bound based on the variational transition distribution instead:

$$\begin{aligned}
\underbrace{I(\mathbf{x}_t; \mathbf{z}_t|\mathbf{z}_{t-\tau})}_{\text{Superfluous information}} &= \mathbb{E}\left[\log \frac{p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{z}_{t-\tau})}{p(\mathbf{z}_t|\mathbf{z}_{t-\tau})}\right] \\
&= \mathbb{E}\left[\log \frac{p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{z}_{t-\tau})}{q_\phi(\mathbf{z}_t|\mathbf{z}_{t-\tau})}\frac{q_\phi(\mathbf{z}_t|\mathbf{z}_{t-\tau})}{p(\mathbf{z}_t|\mathbf{z}_{t-\tau})}\right] \\
&= KL(p_\theta(\mathbf{z}_t|\mathbf{x}_t)||q_\phi(\mathbf{z}_t|\mathbf{z}_{t-\tau})) - \underbrace{KL(p(\mathbf{z}_t|\mathbf{z}_{t-\tau})||q_\phi(\mathbf{z}_t|\mathbf{z}_{t-\tau}))}_{\text{Variational transition gap}} \\
&\leq KL(p_\theta(\mathbf{z}_t|\mathbf{x}_t)||q_\phi(\mathbf{z}_t|\mathbf{z}_{t-\tau})).
\end{aligned} \tag{49}$$

The Expected value of KL-divergence between encoding and transition distribution can be estimated using sampled representations:

$$KL(p_\theta(\mathbf{z}_t|\mathbf{x}_t)||q_\phi(\mathbf{z}_t|\mathbf{z}_{t-\tau})) \approx \log \frac{p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_t)}{q_\phi(\boldsymbol{z}_t|\boldsymbol{z}_{t-\tau})}, \tag{50}$$

with $\boldsymbol{z}_t$ and $\mathbf{z}_{t-\tau}$ as representations sampled from $p_\theta(\mathbf{z}_t|\boldsymbol{x}_t)$ and $p_\theta(\mathbf{z}_{t-\tau}|\boldsymbol{x}_{t-\tau})$ respectively, and $\boldsymbol{x}_t, \boldsymbol{x}_{t-\tau}$ as samples from the process at temporal distance τ. Notably, this procedure is similar to the one used to enforce a bottleneck in Fischer (2020).

D Comparison with the literature

D.1 Linear Correlation Maximization and Mutual Information

A conventional and successful approach to mutual information maximization is the maximization of linear autocorrelation (Andrew et al., 2013; Noé & Nüske, 2013). This can be expressed as:

$$\arg\max_\theta Tr(\text{Cov}[\mathbf{z}_{t-\tau}, \mathbf{z}_t]) \;\text{ subject to } \text{Cov}[\mathbf{z}_{t-\tau}, \mathbf{z}_{t-\tau}] = \text{Cov}[\mathbf{z}_t, \mathbf{z}_t] = \mathbf{I} \tag{51}$$

Here the maximization of the covariance trace is equivalent to the maximization of the sum of its D eigenvalues λ_i, where D denotes the dimensionality of the representation.

A variety of surrogates maximize the sum of squared eigenvalues (Mardt et al., 2018; Wu & Noé, 2020) or the squared Euclidean distance in the representation space (Lyu et al., 2022; Wiskott & Sejnowski, 2002). This objective can also be equivalently interpreted as maximizing mutual information for jointly Normal random variables (Borga, 2001) with linear encoders.

Assume that the representations $\mathbf{z}_{t-\tau}$ and $\mathbf{z}_t$ are jointly Normal distributed:

$$[\mathbf{z}_{t-\tau}, \mathbf{z}_t] \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{S}) \;\text{ with } \boldsymbol{S} = \begin{bmatrix} \boldsymbol{S}_{t-\tau,t-\tau} & \boldsymbol{S}_{t-\tau,t} \\ \boldsymbol{S}_{t,t-\tau} & \boldsymbol{S}_{t,t} \end{bmatrix} \tag{52}$$

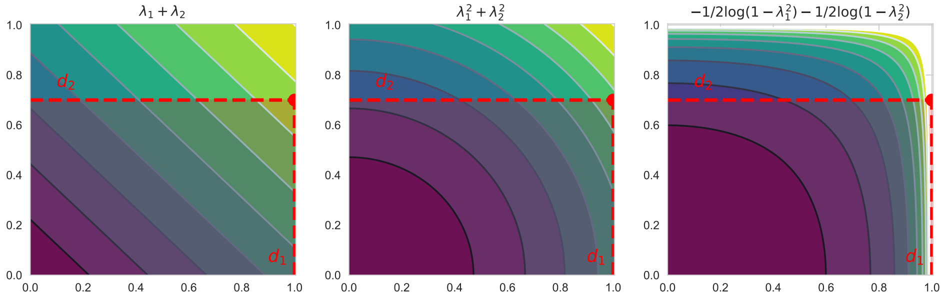


Figure 6: Visualization of several objectives as a function of the eigenvalues λ_1 and λ_2 of $\boldsymbol{S}_{t,t-\tau}\boldsymbol{S}_{t-\tau,t}$. The vertical lines for d_1 and d_2 correspond to the eigenvalues of $\boldsymbol{\Sigma}_{t,t-\tau}\boldsymbol{\Sigma}_{t-\tau,t}$ determined by the original covariance $\boldsymbol{\Sigma}_{t-\tau,t}$. Note that whenever $\mathbf{z}_t$ is a linear projection of $\mathbf{x}_t$, λ_1 and λ_2 are constrained to be in the shaded region determined by d_1 and d_2. As a result, all objectives are optimal for $\lambda_1 = d_1$ and $\lambda_2 = d_2$, which corresponds to a projection onto the principal components.

In this instance, autoinformation can be directly computed as follows:

$$\begin{aligned}
AI_{\mathcal{N}}(\mathbf{z}_{t-\tau}, \tau) &= \frac{1}{2}\log\frac{\det \boldsymbol{S}_{t-\tau,t-\tau}\det \boldsymbol{S}_{t,t}}{\det \boldsymbol{S}} \\
&= \frac{1}{2}\log\frac{\det \boldsymbol{S}_{t,t}}{\det\left(\boldsymbol{S}_{t,t} - \boldsymbol{S}_{t,t-\tau}\boldsymbol{S}_{t-\tau,t-\tau}^{-1}\boldsymbol{S}_{t-\tau,t}\right)} \\
&= -\frac{1}{2}\log\det\left(\mathbf{I} - \boldsymbol{A}\right) \\
&= -\frac{1}{2}\log\det\left(\boldsymbol{U}(\mathbf{I} - \boldsymbol{\Lambda})\boldsymbol{U}^T\right) \\
&= -\frac{1}{2}\log\det\left(\mathbf{I} - \boldsymbol{\Lambda})\right) \\
&= -\frac{1}{2}\sum_{i=1}^{D}\log\left(1 - \lambda_i\right) \qquad (53)
\end{aligned}$$

In which $\boldsymbol{A} := \boldsymbol{S}_{t,t}^{-1/2}\boldsymbol{S}_{t,t-\tau}\boldsymbol{S}_{t-\tau,t-\tau}^{-1}\boldsymbol{S}_{t-\tau,t}\boldsymbol{S}_{t,t}^{-1/2}$, and $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ refers to its eigendecomposition, and λ_i the corresponding eigenvalues. Under the assumption that $\boldsymbol{S}_{t-\tau,t-\tau}$ and $\boldsymbol{S}_{t,t}$ are restricted to be identity matrices, the expression for $\boldsymbol{A}$ simplifies to $\boldsymbol{A} = \boldsymbol{S}_{t,t-\tau}\boldsymbol{S}_{t-\tau,t}$.

As illustrated in Figure 6, for any linear encoder in the form $\mathbf{z}_t = \boldsymbol{W}\mathbf{x}_t$, maximizing the sum of the eigenvalues of $\boldsymbol{A}$, the sum of their squared values, or the expression in equation 53 is equivalent. This is true because under the constraint $\boldsymbol{S}_{t,t} = \boldsymbol{S}_{t-\tau,t-\tau} = \mathbf{I}$, the eigenvalues of $\boldsymbol{S}_{t,t-\tau}\boldsymbol{S}_{t-\tau,t}$ are upper-bounded by the eigenvalues of $\boldsymbol{\Sigma}_{t,t-\tau}\boldsymbol{\Sigma}_{t-\tau,t}$, with $\boldsymbol{\Sigma}_{t-\tau,t} := \text{Cov}[\mathbf{x}_{t-\tau}, \mathbf{x}_t]$.

Note that although the correlation matrix does capture linear relation between $\mathbf{z}_{t-\tau}$ and $\mathbf{z}_t$, it does not consider higher-order interaction between the representations. This is a limiting factor especially for low-dimensional representations because of the expressive power of linear transformations. This phenomenon can be clearly observed by comparing the autoinformation plots in Figure 4b (2D representations) and Figure 13 (16/32 dimensional representations). The autoinformation extracted by representations that use linear correlation maximization (TICA and VAMPNet) strongly depends on the number of dimensions of the representation $\mathbf{z}_t$. The effect on methods that rely on non-linear contrastive mutual information maximization methods is much more moderate, making them more flexible and suitable for 2D visualizations.

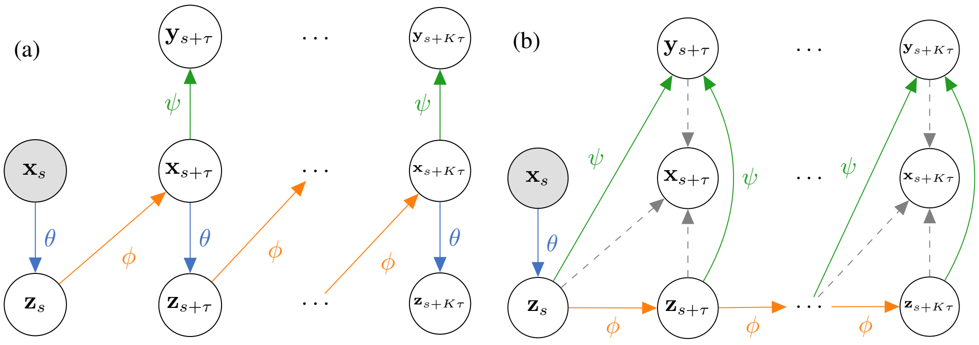


Figure 7: Viable inference schemes for maximal future state information: $AI(\mathbf{x}_t;\tau) - I(\mathbf{z}_t;\mathbf{x}_{t+\tau}) = 0$. The difference with the Latent Simulation inference scheme lies in the lack of the conditional independence $I(\mathbf{z}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_{t+\tau}) = 0$. Note that modeling $q_\phi(\mathbf{x}_t|\mathbf{z}_{t-\tau})$ is generally more difficult than modeling latent transitions $q_\phi(\mathbf{z}_t|\mathbf{z}_{t-\tau})$.

D.2 Maximizing information with respect to future states

Several models in the literature consider the reconstruction of future states as the training objective (Wehmeyer & Noé, 2018; Wang et al., 2019b). This objective can be interpreted as the maximization of the mutual information between the current representation $\mathbf{z}_t$ and the future state $\mathbf{x}_{t+\tau}$ Poole et al. (2019):

$$\begin{aligned}\max_\theta I(\mathbf{z}_t;\mathbf{x}_{t+\tau}) &= \max_\theta H(\mathbf{x}_{t+\tau}) - H(\mathbf{x}_{t+\tau}|\mathbf{z}_t)\\ &\geq H(\mathbf{x}_{t+\tau}) - \min_{\theta,\phi} \mathbb{E}_{p(\mathbf{x}_t)p_\theta(\mathbf{z}_t|\mathbf{z}_t)}\left[-\log q_\phi(\mathbf{x}_{t+\tau}|\mathbf{z}_t)\right], \end{aligned} \tag{54}$$

in which $q_\phi(\mathbf{x}_{t+\tau}|\mathbf{z}_t)$ refers to the decoder that predicts the future states given the current representation.

We note that autoinformation in $\mathbf{z}_t$ is always smaller or equal to $I(\mathbf{z}_t;\mathbf{x}_{t+\tau})$, which we will refer to as *future predictive information*:

$$AI(\mathbf{x}_t;\tau) = I(\mathbf{x}_t;\mathbf{x}_{t+\tau}) \geq I(\mathbf{z}_t;\mathbf{x}_{t+\tau}) \geq I(\mathbf{z}_t;\mathbf{z}_{t+\tau}) = AI(\mathbf{z}_t;\tau). \tag{55}$$

Preserving autoinformation is a stronger condition than having maximal future predictive information:

$$AIG(\mathbf{z}_t;\tau) = 0 \implies AI(\mathbf{x}_t;\tau) - I(\mathbf{z}_t;\mathbf{x}_{t+\tau}) = 0. \tag{56}$$

This is because the additional condition $I(\mathbf{z}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_{t+\tau}) = 0$ is required:

$$\begin{aligned}AIG(\mathbf{z}_t;\tau) &= AI(\mathbf{x}_t;\tau) - I(\mathbf{z}_t;\mathbf{z}_{t+\tau})\\ &= \underbrace{AI(\mathbf{x}_t;\tau) - I(\mathbf{z}_t;\mathbf{x}_{t+\tau})}_{\text{Missing future predictive information}} + I(\mathbf{z}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_{t+\tau}).\end{aligned} \tag{57}$$

This additional condition is not directly required to prove the results of Lemma 1, Lemma 2, and Lemma 3, which can be extended to the condition $AI(\mathbf{x}_t;\tau) - I(\mathbf{z}_t;\mathbf{x}_{t+\tau}) = 0$. However, the lack of the conditional independence $I(\mathbf{z}_t;\mathbf{x}_{t+\tau}|\mathbf{z}_{t+\tau}) = 0$ would result in a difference inference scheme, in which instead of approximating transitions directly in the latent space, each step would require modeling transitions from $\mathbf{z}_t$ to $\mathbf{x}_{t+\tau}$, as shown in Figure 7a. Alternatively, one could model latent transitions $q_\phi(\mathbf{z}_t|\mathbf{z}_{t-\tau})$, but the predictive target distribution would depend on both the current and future representation, as shown in Figure 7b. Both inference schemes and the maximization of $I(\mathbf{z}_t;\mathbf{x}_{t+\tau})$ are more computationally expensive than the proposed T-InfoMax training procedure and Latent Simulation inference.

D.3 State Predictive Information Bottleneck and target sufficiency

Wang & Tiwary (2021) introduce a State Predictive Information Bottleneck (SPIB) objective aiming to create a representation $\mathbf{z}_t$ that is sufficient for the next target $\mathbf{y}_{t+\tau}$ while compressing information:

$$\mathcal{L}^{SPIB}(\theta;\beta,\tau) = -\mathbb{E}_t[I(\mathbf{z}_t;\mathbf{y}_{t+\tau}) - \beta I(\mathbf{x}_t;\mathbf{z}_t)]. \tag{58}$$

Although this objective seems natural for training effective representations, we can show that sufficiency for a given target $\mathbf{y}_{t+\tau}$ is a necessary but not sufficient condition for autoinformation preservation. As a result, a representation that is optimal according to the SPIB objective may introduce inference error even when the true latent transition $p(\mathbf{z}_t|\mathbf{z}_{t-\tau})$ and latent future predictive $p(\mathbf{y}_{t+\tau}|\mathbf{z}_t)$ distributions are available at inference time, as shown in the following example.

Consider a dynamic system in which each state is described by a particle position, velocity, and acceleration governed by a simple time-discrete update:

$$\mathbf{x}_t = \begin{bmatrix}\mathbf{r}_t\\ \mathbf{v}_t\\ \mathbf{a}_t\end{bmatrix} = \begin{bmatrix}\mathbf{r}_{t-\tau} + \tau\mathbf{v}_{t-\tau}\\ \mathbf{v}_{t-\tau} + \tau\mathbf{a}_{t-\tau}\\ \boldsymbol{\eta}_t\end{bmatrix} = D_\tau(\mathbf{x}_{t-\tau}, \boldsymbol{\eta}_t), \tag{59}$$

in which the acceleration at each time step is sampled from a time-independent Normal distribution $\boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ and D_τ refers to the function used to unroll the true system dynamics at the time scale τ. Clearly, the system is an instance of a homogenous Markov process.

We are interested in predicting the particle position $\mathbf{y}_t = \mathbf{r}_t$. Clearly, since the next position depends solely on the current position and the current velocity, we have that a representation that contains only velocity and position information is sufficient for the next target prediction:

$$I(\mathbf{y}_{t+\tau}; \mathbf{x}_t) = I(\mathbf{y}_{t+\tau}; \mathbf{z}_t^{SPIB}) \quad \text{with } \mathbf{z}_t^{SPIB} = \begin{bmatrix}\mathbf{r}_t\\ \mathbf{v}_t\end{bmatrix} \tag{60}$$

On the other hand, a representation that maximizes autoinformation (and is optimal according to Equation 8) must also contain information regarding the acceleration since current acceleration is predictive for the future velocity:

$$I(\mathbf{x}_t; \mathbf{x}_{t+\tau}) = I(\mathbf{z}_t^{T-IB}; \mathbf{z}_{t+\tau}^{T-IB}) > I(\mathbf{z}_t^{SPIB}; \mathbf{z}_{t+\tau}^{SPIB}) \quad \text{with } \mathbf{z}_t^{T-IB} = \begin{bmatrix}\mathbf{r}_t\\ \mathbf{v}_t\\ \mathbf{a}_t\end{bmatrix}. \tag{61}$$

Note that a representation that is optimal according to SPIB would instead explicitly discard acceleration because of the compression regularization:

$$I(\mathbf{x}_t; \mathbf{z}_t^{T-IB}) > I(\mathbf{x}_t; \mathbf{z}_t^{SPIB}). \tag{62}$$

Since both representations are sufficient for $\mathbf{y}_{t+\tau}$, they yield the same predictive distribution for the next target:

$$p(\mathbf{y}_{t+\tau}|\mathbf{z}_t^{SPIB}) = p(\mathbf{y}_{t+\tau}|\mathbf{z}_t^{T-IB}) = p(\mathbf{y}_{t+\tau}|\mathbf{x}_t). \tag{63}$$

However, if we look at the predictive distribution at times larger than τ, we observe some discrepancies. In particular, we can show that:

$$p(\mathbf{y}_{t+2\tau}|\mathbf{z}_t^{SPIB}) \neq p(\mathbf{y}_{t+2\tau}|\mathbf{z}_t^{T-IB}) = p(\mathbf{y}_{t+2\tau}|\mathbf{x}_t), \tag{64}$$

In which the first inequality follows from the fact that $\mathbf{z}_t^{SPIB}$ does not contain knowledge about the acceleration, while the second inequality follows from Lemma 1+Lemma 3. Therefore we showed that latent simulation performed on representations that are optimal according to the SPIB objective (and not according to T-IB) introduces inference error for time scales larger than τ. The intuition is that sufficiency for the next target $\mathbf{y}_{t+\tau}$ does not guarantee a transfer of the Markov property from the original space $\mathbf{x}_t$ to the representation $\mathbf{z}_t$. That requirement is satisfied only whenever the representation $\mathbf{z}_t$ preserves autoinformation, as shown in Lemma 1+ Lemma 2.

E Experimental Details

We include additional details regarding the training data, architectures, and optimization procedure to ensure the reproducibility of the reported results.

E.1 Data

E.1.1 Prinz 2D

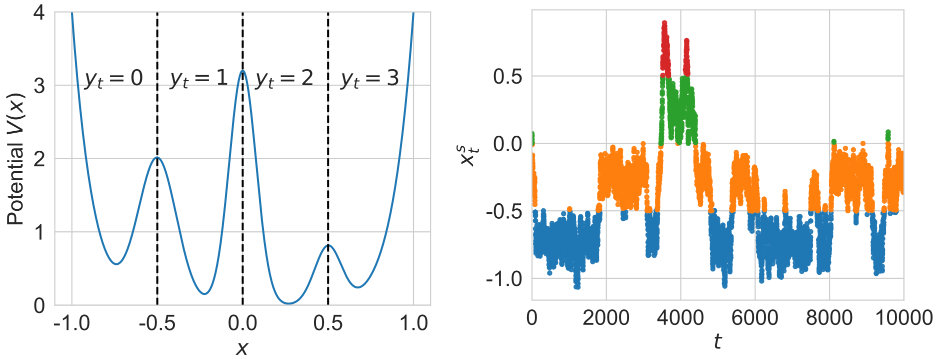


Figure 8: Left: visualization of the 1D Prinz potential, and the corresponding regions used to define the discrete targets $\boldsymbol{y}$. Right: Visualization of the 1D slow component x_t^s colored by y_t^s.

The Prinz 2D trajectories consist of sequences of 100K data points generated by diffusing a point particle into a potential $V(x) := 4\left(x^8 + 0.8e^{-80x^2} + 0.2e^{-80(x-0.5)^2} + 0.5e^{-40(x+0.5)^2}\right)$ with an Euler-Maruyama integrator following the update:

$$x_{t+1} = x_t - h\nabla V(x_t) + \sqrt{h}\,\eta_t, \tag{65}$$

in which $h = 10^{-4}$ refers to the integrator step and η_t is standard Normal uncorrelated noise. We generate $\left[x_t^f\right]_{t=s}^T$ by performing 160 integration steps in-between consecutive timesteps, while $[x_t^s]_{t=s}^T$ is generated by considering 5 integration steps. The Deep Time package (Hoffmann et al., 2021) is used to produce the slow and fast trajectories, and the corresponding potential $V(x)$ is visualized in Figure 8. The fast and slow independent components are then mixed as follows:

$$\boldsymbol{x}_t = \begin{bmatrix} \tanh(x_t^s + x_t^f) \\ \tanh(x_t^s - x_t^f) \end{bmatrix}, \tag{66}$$

to produce the trajectories visualized in Figure 3a.

E.1.2 Molecular Data

Trajectories We analyze trajectories obtained by simulating *Alanine Dipeptide*, *Chignolin*, and *Villin* (Lindorff-Larsen et al., 2011). For Alanine Dipeptide, the three splits correspond to separate simulations of 250K/100K/100K frames respectively. In contrast, for Chignolin and Villin simulation, a single trajectory is split into 3 temporally disjoint parts: 334.743/100K/100K frames for Chignolin, and 427.907/100K/100K frames for Villin. Each observation $\mathbf{x}_t$ consists of the set of the Euclidean coordinates of all the atoms and a one-hot corresponding to the atomic number for the Alanine Dipeptide trajectories. The input data for the mini-proteins, on the other hand, consists of a coarse-grained representation indicating the 3D location of the amino acids in the protein chain (10 for Chignolin and 35 for Villin), along with a one-hot encoding for the amino acid type.

Targets Targets for the Alanine Dipeptide molecules are generated by clustering torsion angles ϕ and ψ into 6 regions, corresponding to the known meta-stable states. For the Chignolin and Villin molecules, we generate targets $\mathbf{y}_t$ by clustering the 32D invariant TICA projections obtained by

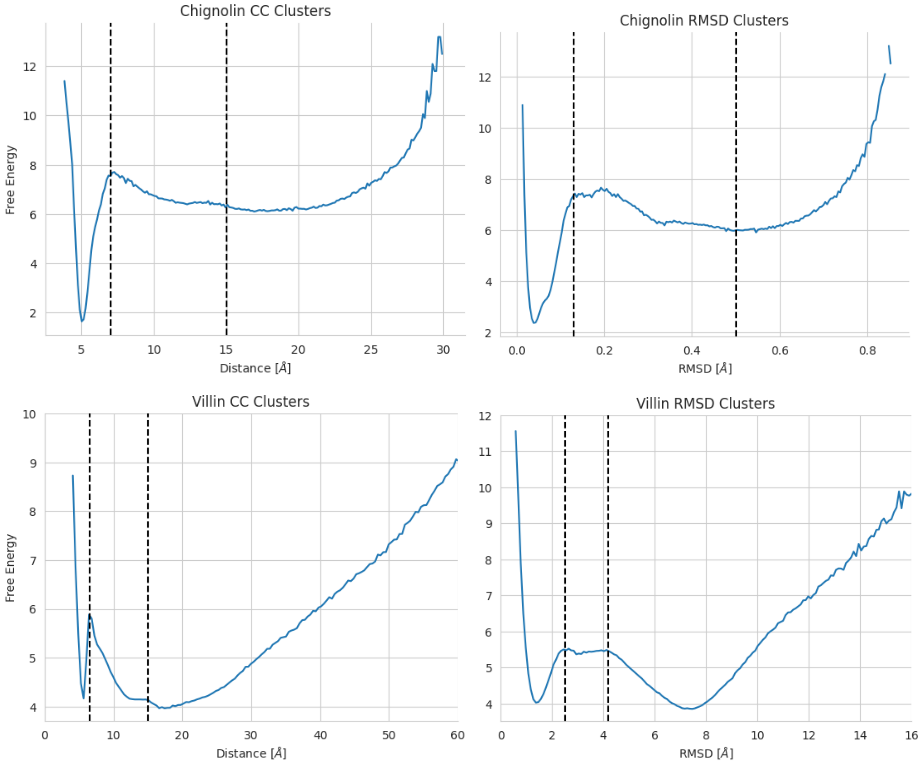


Figure 9: Visualization of the 1D free energy induced by the distribution of the distances between first and last C-alpha atoms in the chain (CC) and Root Mean Squared Distance (RMSD) for the molecules of Chignoling and Villin. Vertical dashed lines are used to denote the margin between the different discretized regions. The thresholds are set to $[7, 15]$, $[1.3, 5]$ Angstroms for Chignolin CC and RMDS respectively, while the values of $[6.5, 15]$ and $[2.5, 4.2]$ are used for Villin.

following the same procedure described in Köhler et al. (2023) using KMeans with 5 centroids, as depicted in Figure 4a. We produce additional sets of targets by considering the distance between the first and last C-alpha carbon atoms in the amino acid sequence (CC, 3 clusters), and the Root Mean Squared Distance (RMSD, 3 clusters) from the stable folded configuration. The thresholds used to create the clusters are visualized together with the corresponding free energy in Figure 9.

Lag time We decide on a training lag time τ for each molecule that is long enough to capture relevant meta-stable state transitions, see Figure 4b and Figure 13. We focus on a time scale on which most of the dynamic information is still present while modeling transitions that are orders of magnitudes larger compared to the original simulations. We used a train lag time of 16 ps on Alanine Dipeptide simulations, while 3200 ps was used for the Chignolin and Villin simulations. The same value of τ is used both to train the encoder (step 1) and the transition model (step 2).

E.2 Architectures and Optimization

The models used for the experiments reported in this paper are described in detail in the following sections. The experiments reported in this paper required a total of 25 days of computation on $A100$ GPUs. This estimation includes model development, hyper-parameter tuning, and evaluation.

E.2.1 Encoder training

We train each encoder for a maximum of 50 epochs with mini-batches of size 512 using the AdamW (Loshchilov & Hutter, 2019) optimizer. To prevent overfitting, we use early stopping based on the validation loss. Following previous work (Chen et al., 2020), the models are trained with an initial learning rate of 10^{-6}, which is gradually increased up to 5×10^{-4} over the course of 5 epochs with a linear schedule. The learning rate is then decreased to the initial value using a cosine schedule over the following 45 epochs.

For the Prinz 2D experiments, encoders consist of MLPs with two hidden units of size 64 and a 2D output. In the molecular settings, each encoder architecture consists of a TorchMD Equivariant Transformer (Thölke & Fabritiis, 2022) followed by global mean pooling and a linear layer to produce a rotation, translation, reflection, and permutation invariant representation for each molecule. We use a total of 3 layers of 32 hidden units with 8 heads each for the Alanine Dipeptide experiment. The more challenging Chignolin and Villin molecules use encoders consisting of 5 layers with 64 hidden units and 8 projection heads instead. For the evaluation of the quality of unfolded trajectories, we use a 16-dimensional representation for Alanine Dipeptide. A total of 32 dimensions are used for Chignolin and Villin.

TICA Temporal Independent Component Analysis consists of a linear projection of the input data onto the principal temporal components. As a result, $p_\theta(\mathbf{z}_t|\mathbf{x}_t)$ consists of a simple linear projection instead of a neural network that has been optimized using the Deeptime python library (Hoffmann et al., 2021). For the Prinz2D experiments, we apply TICA directly to the original sequence $[\mathbf{x}_t]_{t=s}^T$ to project each data point $\mathbf{x}_t$ onto the principal temporal component z_t. For the Alanine Dipeptide Experiments, the TICA representations correspond directly to the torsion angles determined by the carbon skeleton (2 angles), which are commonly used in literature to describe the configuration of this small molecule (Vymětal & Vondrášek, 2010; Mardt et al., 2018). The representations for Chignolin and Villin are produced following the same procedure described in detail in Köhler et al. (2023), in which torsion angles and inter-atomic distances are projected onto the principal temporal components.

VAMPNet We train the encoder $p_\theta(\mathbf{z}_t|\mathbf{x}_t)$ using VAMP-2 score (Mardt et al., 2018; Wu & Noé, 2020) using the implementation from the Deeptime python library (Hoffmann et al., 2021). The VAMPNet model was originally designed for estimating dominant spectral components of molecular simulations. However, Sidky et al. (2020) has shown the effectiveness of VAMPNet for Latent Simulation inference.

T-InfoMax As a representative of non-linear mutual information maximization methods, we consider the popular InfoNCE method (van den Oord et al., 2018; Chen et al., 2020). Following the literature (van den Oord et al., 2018; Poole et al., 2019), we model the log-ratio between joint and product distribution with a separable architecture:

$$F_\xi(\mathbf{z}_{t-\tau};\mathbf{z}_t) = g_{\xi_1}(\mathbf{z}_{t-\tau})^T g_{\xi_2}(\mathbf{z}_t), \tag{67}$$

in which g_{ξ_1} and g_{ξ_2} are neural networks mapping the latent representations into a 128-dimensional normalized vector. The two architectures have distinct weights with one hidden layer of 256 units and group normalization (Wu & He, 2018) before the ReLU non-linearity.

T-IB Analogously to the T-InfoMax counterpart, the Time-lagged Information Bottleneck objective makes use of InfoNCE for time-lagged information maximization, with an additional regularization term modulated by the hyper-parameter β as shown in equation 9. Following (Federici et al., 2020) we first train the encoder with an initial value of $\beta = 10^{-6}$ for 5 epochs. This is to prevent the representation from collapsing into a constant at the beginning of training. Secondly, the regularization strength is gradually increased up to the final desired value over the course of 30 epochs. We empirically observed that the T-IB models benefit from the use of a stochastic encoder $p_\theta(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{z}_t|\mu_\theta(\mathbf{x}_t), \sigma_\theta(\mathbf{x}_t)\mathbf{I})$. The parameter vectors $\mu_\theta(\mathbf{x}_t)$ and $\sigma_\theta(\mathbf{x}_t)$ are obtained using two linear projection heads on top of the encoder features, as a result, the size of the stochastic encoders is comparable to the corresponding deterministic counterpart.

We initialize the architectures with a value of $\sigma_\theta(\mathbf{x}_t) \approx 10^{-4}$ to reduce the amount of Gaussian additive noise in the initial part of the training. Empirical results showed that the additional stochasticity produces smooth transitions between different levels of regularization strength. This is in contrast with the sharp regime changes observed with deterministic encoders (Figure 11, **Top**). We believe that this is due to the fact that the addition of Gaussian noise allows the encoder to destroy superfluous information locally when necessary.

E.2.2 Transition and Prediction training

The variational transition and prediction models ($q_\phi(\mathbf{z}_t|\mathbf{z}_{t-\tau})$ and $q_\psi(\mathbf{z}_t|\mathbf{y}_t)$, respectively) are jointly trained on the embedding produced by the encoder trained in the previous step. The training procedure uses mini-batches of size 512 with AdamW and a fixed learning rate of 5×10^{-4} over a total of 50 epochs. Contrary to the previous step, we did not observe any overfitting with only marginal improvements in the training and validation scores by the end of the training procedure.

Transition The transition model consists of conditional Flow++ layers (Ho et al., 2019) due to their flexibility, sampling speed, and ability to model correlated distributions. The transitions for Prinz2D and Alanine Dipeptide representations consist of 3 flow layers. Each layer is composed of a conditional mixture of logistics CDF coupling transformation consisting of a neural network with two hidden layers of 64 hidden units, which maps the representations $\mathbf{z}_{t-\tau}$ into the parameters of a mixture of 16 logistics distributions. An architecture of 5 layers is used to learn the more challenging transition distributions for Chignolin and Villin. To prevent numerical overflows while unfolding long simulations, we clip samples to be in the interval $[-10^6, 10^6]$.

Prediction Each feature predictor used in this work consists of a simple 1-hidden layer MLP with 128 hidden units mapping the representation $\mathbf{z}_t$ into the logits for the variational predictive distribution $q_\psi(\mathrm{y}_t|\mathbf{z}_t)$.

E.3 Evaluation

We focus our evaluation on two main aspects. First, we analyze the amount of autoinformation that several models extract from the molecular data to better understand which temporal characteristics of the molecular process are successfully captured. The second aspect involves the evaluation of the fidelity of trajectories unfolded using the Variational Latent Simulation process.

E.3.1 Mutual Information

Autoinformation We estimate autoinformation for evaluation purposes using SMILE (Song & Ermon, 2020) on the trained representations $\mathbf{z}_t$ with a clipping interval of $[-5, 5]$. The ratio estimation architecture consists of an initial projection head $g : \mathbb{Z} \to \mathbb{R}^{128}$ with one hidden layer of 256 units and output $\mathbf{h}_t := g(\mathbf{z}_t)$ with a dimension of 128. Pairs of the 128-dimensional feature vectors $\mathbf{h}_t$ at different temporal resolutions are then concatenated and fed into a second MLP $r_\tau : \mathbb{R}^{128} \times \mathbb{R}^{128} \to \mathbb{R}$ with 64 hidden units and 1 output, which corresponds to the estimated log-ratio value. Each pair of $\mathbf{h}_t, \mathbf{h}_{t+\tau}$ is fed into a distinct architecture r_τ for each τ. This setup allows us to estimate autoinformation at several time-lags at once to produce the plots visualized in Figure 4b, Figure 13 and Figure 14a. Each dot in the figure corresponds to the expected output of one ratio estimation model $r_\tau(g(\mathbf{z}_t), g(\mathbf{z}_{t+\tau}))$ on the entirety of the training set. The ratio estimation models are fit for at most 20 epochs using early stopping based on the validation loss. Note that samples from the marginal distribution used to estimate the value of the partition function are sampled by sampling $\mathbf{x}_{t'}$ with uniform probability using the same strategy described in Appendix C.2. Estimation is performed using the Torch-Mist package(Federici et al., 2023).

Target Information Following Poole et al. (2019); McAllester & Stratos (2020); Song & Ermon (2020), we estimate the amount of target information in the representations as a difference of cross-entropies:

$$I(\mathbf{z}_t; \mathbf{y}_t) = H(\mathbf{y}_t) - H(\mathbf{y}_t|\mathbf{z}_t) \leq H(\mathbf{y}_t) - \mathbb{E}[-\log q_\psi(\mathbf{y}_t|\mathbf{z}_t)], \tag{68}$$

in which the marginal entropy $H(\mathbf{y}_t)$ for the discrete targets $\mathbf{y}_t$ is estimated by counting the frequency of each class, while the expected cross entropy $\mathbb{E}[-\log q_\psi(\mathbf{y}_t|\mathbf{z}_t)]$ is evaluated using the trained predictor $q_\psi(\mathbf{y}_t|\mathbf{z}_t)$ on the entirety of the test trajectory and computing the corresponding expected log-likelihood. Note that with $I(\mathbf{z}_t; \mathbf{y}_t)$ we implicitly refer to the expected amount of target information over an entire trajectory rather than the amount of information estimated specifically at the time-step t.

E.3.2 Unfolding trajectories

Accurately estimating a measure of divergence between joint distributions when only samples are accessible is generally a challenging task due to the number of samples required for a reliable estimation. For this reason, instead of considering continuous multi-dimensional targets $\mathbf{y}_t$, we focus our attention on discrete targets. The targets in our experiments are designed to capture properties of interest of the trajectories

Our evaluation procedure can be described in 3 steps:

1. First we encode the initial (unobserved) test state $\boldsymbol{x}_s$ into the latent configuration $\boldsymbol{z}_s$ using $p_\theta(\mathbf{z}_t|\mathbf{x}_t)$. Starting from $\boldsymbol{z}_s$, we sample a total of 256 trajectories $\left[\tilde{\boldsymbol{z}}^{(i)}_{s+k\tau}\right]_{k=1}^{K}$ by sampling from the variational transition model $q_\phi(\mathbf{z}_t|\mathbf{z}_{t-\tau})$ sequentially for a total temporal duration which is comparable to the time-span covered by the test trajectories $T - s \approx K\tau$. Using the prediction model $q_\psi(\mathrm{y}_t|\mathbf{z}_t)$ we then sample a target $\tilde{y}^{(i)}_t$ for each sampled $\tilde{\boldsymbol{z}}^{(i)}_t$, obtaining 256 sequences of targets $\left[\tilde{y}^{(i)}_{+k\tau}\right]_{k=1}^{K}$.
2. We count the number of transitions from each discrete state $\tilde{y}^{(i)}_t$ to the following $\tilde{y}^{(i)}_{t+k\tau}$ for various numbers of steps k, effectively creating a series of transition count matrix $\tilde{\boldsymbol{C}}^{(i)}_{k\tau}$ and $\boldsymbol{C}_{k\tau}$ respectively for $\left[\tilde{y}^{(i)}_{+k\tau}\right]_{k=1}^{K}$ and $[y_t]_{t=s}^{T}$. The 256 count matrices for the unfolded trajectories are then averaged to produce $\tilde{\boldsymbol{C}}_{k\tau} = 1/256\sum_{i=1}^{256}\tilde{\boldsymbol{C}}^{(i)}_{k\tau}$. We normalize each row of $\tilde{\boldsymbol{C}}_{k\tau}$ and $\boldsymbol{C}_{k\tau}$ to estimate the transition probability matrices $\tilde{\boldsymbol{T}}_{k\tau}$ and $\boldsymbol{T}_{k\tau}$. Analogously, we count the number of times that each state is visited to determine the normalized counts $\boldsymbol{m}$ and $\tilde{\boldsymbol{m}}$ using the ground truth and unfolded trajectories respectively.
3. We compute the Jensen-Shannon divergence between each row of $\boldsymbol{T}_{k\tau}$ and $\tilde{\boldsymbol{T}}_{k\tau}$, then we average the values obtained for each row into a single number, representing the average Jensen-Shannon divergence. With this last step, we obtain one value of transition Jensen-Shannon divergence (TJS) for each chosen number of unfolding steps k (see Figure 15). The values for each row are averaged using the same weighting instead of the relative state probability to accentuate errors when transitioning from rare states. Analogously we compute the value of marginal JS (MJS) by computing the divergence between the probability distribution induced by $\boldsymbol{m}$ and $\tilde{\boldsymbol{m}}$.

F Additional Results

In this section, we report additional ablation studies and the performance of the models considered in this analysis for different sets of targets.

F.1 T-IB regularization strength and train lag time

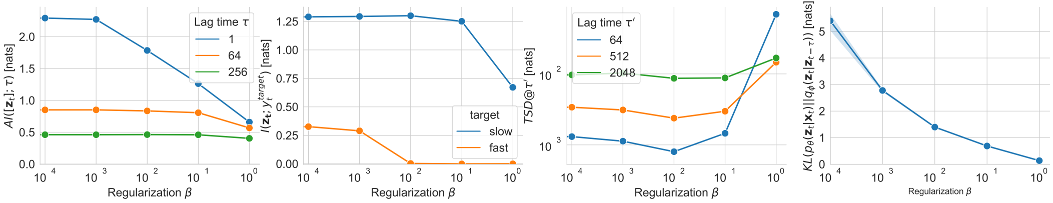


Figure 10: Visualization of the effect of the regularization strength β on Autoinformation, information regarding slow and fast modes, transition JS, and amount of superfluous information on the Prinz 2D dataset. All representations are trained using $\tau = 64$. Representations trained with $\beta < 0.01$ tend to contain information regarding the fast mode and higher autoinformation at small temporal scales, while strong regularization $\beta > 0.1$ results in representations that contain too little information. Note that the best performance in terms of transition JS divergence is achieved by the representation that contains the least information regarding y^f_t and most about y^s_t, which corresponds to the most compressed sufficient representation.

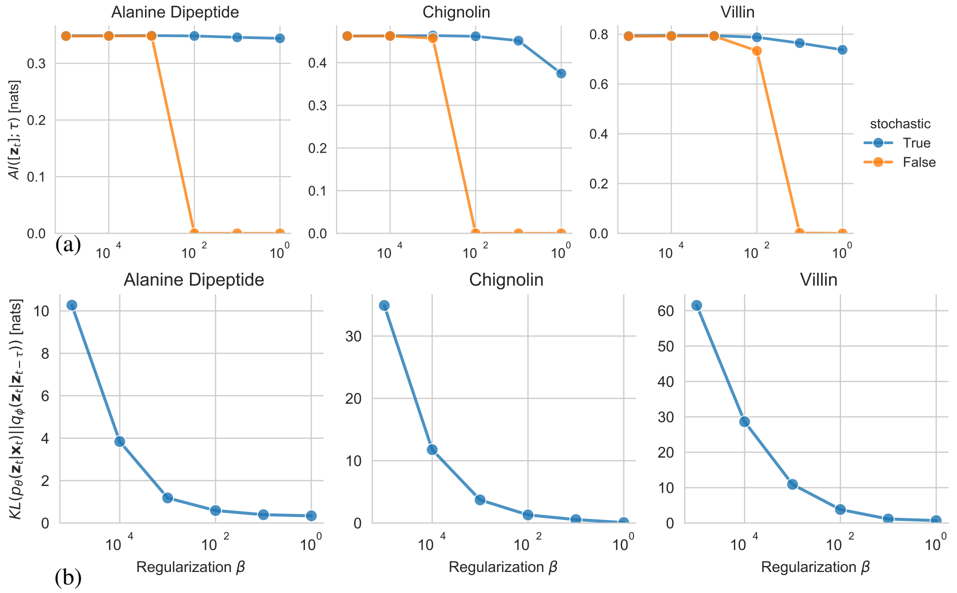


Figure 11: Visualization of the effect of the regularization strength on the representations produced with T-IB on molecular simulations with fixed train lag time τ. 11a: estimated autoinformation (y-axis) for the three molecules as a function of the training regularization strength β (x-axis). Stochastic encoders (in blue) show a much smoother interpolation. 11b: the amount of superfluous information (y-axis, Equation 49) as a function of the regularization strength.

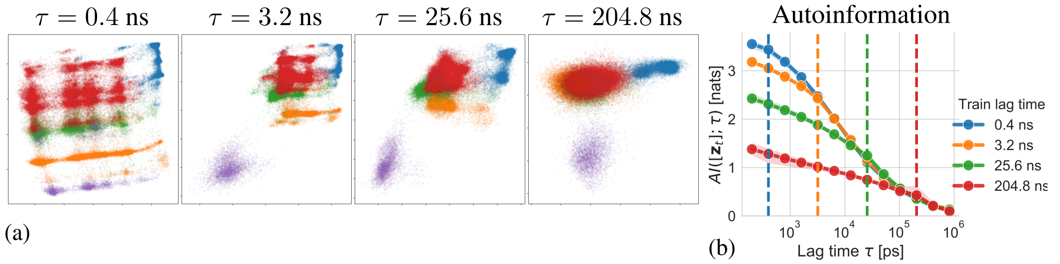


Figure 12: Visualization of the effect of the train lag time τ on 2D T-IB representations of Villin trained with $\beta = 0.01$. 12a: representation of the test trajectories for models trained with several lag times, colored by the clustered TICA embedding, as reported in Figure 4a. 12b: corresponding autoinformation plot in which the dashed vertical lines correspond to the respective training lag times. Note that, as motivated in Section 2.2, representation trained with a higher temporal resolution also captures slower processes at the cost of introducing more information into the representation. This can be clearly seen by observing the number of distinct clusters emerging in the visualized representations.

Figure 10 reports the effect of the regularization strength for T-IB representations of the Prinz 2D data. Consistently with the hypothesis, the best-performing model is the one that produces minimal sufficient representations at the training time scale $\tau = 64$. This corresponds to a regularization strength of $\beta = 0.01$.

Figure 11 compares the effects of several regularization strengths, demonstrating the differences between deterministic and stochastic encoders. Deterministic encoders (in yellow) tend to sharply transition from a fully informative representation (on the left) to a constant uninformative representation (on the right). A secondary advantage of using a stochastic encoder is the possibility to compute an upper bound of superfluous information thanks to the expression for the density $p_\theta(\mathbf{z}_t|\mathbf{x}_t)$. This is generally not possible for a deterministic encoder for which $KL(p_\theta(\mathbf{z}_t|\mathbf{x}_t)||q_\phi(\mathbf{z}_t|\mathbf{z}_{t-\tau}))$ can be evaluated only up to a constant. Regularization strength β for T-IB is selected based on validation performance: $\beta = 0.01$ for Alanine Dipeptide and Villin; $\beta = 0.001$ for Chignolin.

In our experiments on molecular data, we observed that even small values of β can have a substantial impact on reducing the amount of superfluous information contained in the representations, with only a moderate impact on autoinformation. We believe the possible reduction of autoinformation for larger β is due to the fact that processes faster than τ cannot always be fully disentangled. This includes processes that contain lots of information at smaller time scales, but are only marginally informative for events that are τ time-steps apart. Reducing information regarding faster processes can drop the amount of superfluous information in the representation but still decrease autoinformation whenever the faster process can not be temporally disentangled. Nevertheless, regularization strength in the order of 10^{-3} reduces the amount of superfluous information by a substantial factor ($10\times$) with little to no effect on the amount of extracted autoinformation at τ.

Figure 12b shows the effect of the train lag time selection on T-IB models trained with $\beta = 0.01$ and a 2-dimensional representation space for the Villin trajectory. Smaller train lag time corresponds to higher information content and more complex representations, while larger train time scales are associated with simpler representations that are not suitable for unfolding simulation at higher temporal resolution. Note that the larger the training lag-time the longer the training trajectories need to be.

F.2 Autoinformation for larger representations

Plots in Figure 4b, Figure 14a, and Figure 13 confirm that with an appropriate regularization strength, T-IB model preserves the maximum amount of autoinformation at the training timescale while decreasing autoinformation for smaller lag times (left of the dashed lines).

Note that the autoinformation plot all the models considered in this analysis matches for large time scales. This suggests that all the corresponding representations preserve autoinformation at large lag times while still differing in the amount of superfluous information at faster scales and the representation structure. The perfect overlap is also justified by Lemma 3 which guarantees that representations that preserve autoinformation at some lag time τ must also preserve information at larger lag times.

Encoders trained with the VAMPNet objective on complex systems tend to preserve autoinformation only for slower processes. We further observe that VAMPNet models tend to become less numerically stable for increasing representation size, while methods based on non-linear autoinformation maximization are less affected by this hyperparameter choice.

F.3 Evaluating statistics for multiple targets and time-steps

Figure 15, Figure 14b, and Table 1 report the values of average Jensen-Shannon divergence for transition distribution for different targets $\mathbf{y}_t$. We observe that the T-IB model consistently outperforms the other models for transition matrices computed based on different objectives and several lag times.

One of the main challenges for the evaluation of statistics of slow processes (large transition times in Figure 15) lies in the limited amount of test time frames. We observed that, for large time intervals, the estimation of the ground-truth transition distribution from rare states may be too noisy to produce accurate measures of Jensen-Shannon divergence. As a result, the values reported for large transition times (x-axis) become dependent on the specific test trajectory used for evaluation. Nevertheless, we believe that the relative comparison between the performance of different models may still represent their ability to match the original statistics. More accurate quantitative analysis in this regime would require access to much longer molecular simulations.

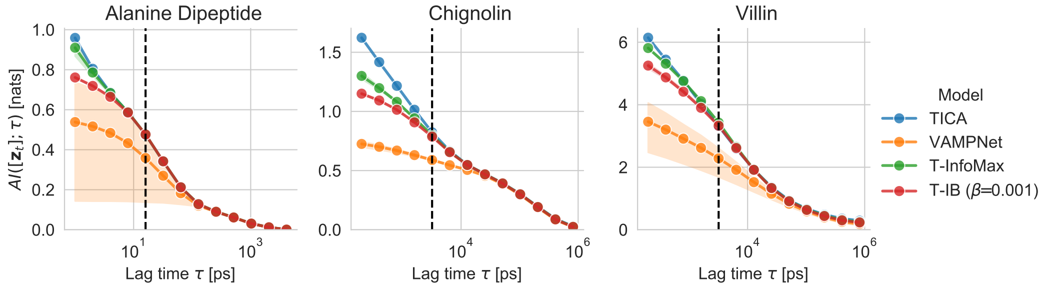
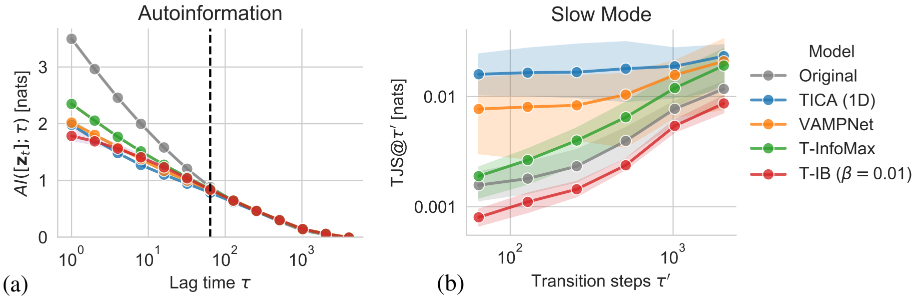


Figure 13: Autoinformation plot for high dimensional representations (16 for Alanine Dipeptide, 32 for Chignolin and Villin). Shaded regions indicate the standard deviation measured across 3 seeds and the dashed vertical line indicates the lag time at which the representations are trained. Representations trained with the VAMPNet objective are generally less consistent (higher variance) across different seeds. T-IB produces sufficient representations (maximal autoinformation at the training time scale) while minimizing the autoinformation for smaller scales.



(a) (b)

Figure 14: Measurements of autoinformation and transition JS estimated for several time scales. 14a: values of autoinformation estimated at several lag times τ for representations trained with $\tau = 64$. 14b: values of transition JS estimated at several time scales τ' from unfolded trajectories. T-IB contains the least autoinformation at small time scales while preserving information at the train lag time or larger. At the same time, T-IB results in the smaller TJS at all the considered time scales. The measure of standard deviation is obtained by considering 3 seeds for each model.

	Chignolin				Villin			
	CC Cluster		RMSD Cluster		CC Cluster		RMSD Cluster	
	MJS	TJS@51.2 ns	MJS	TJS@51.2 ns	MJS	TJS@51.2 ns	MJS	TJS@51.2 ns
TICA	7.5 ± 7.6	5.2 ± 3.4	7.4 ± 7.8	6.4 ± 3.7	1.7 ± 0.5	6.1 ± 2.4	7.3 ± 6.1	5.3 ± 3.7
VAMPNet	30 ± 20	103 ± 68	31.9 ± 21.8	117 ± 102	63 ± 88	57 ± 47	7 ± 41	40 ± 7
T-InfoMax	5.0 ± 1.7	3.5 ± 0.8	4.8 ± 1.6	3.3 ± 0.7	2.1 ± 0.5	5.3 ± 1.9	5.8 ± 2.3	8.5 ± 2.4
T-IB	3.3 ± 2.3	1.1 ± 0.2	2.9 ± 2.2	4.1 ± 1.1	0.8 ± 0.3	4.4 ± 1.1	1.7 ± 1.1	4.1 ± 1.6

Table 1: Values of marginal (MJS) and transition (TJS) Jensen-Shannon divergence for trajectories unfolded on latent spaces obtained with different models for the prediction of the CC and RMSD cluster targets described in Appendix E.1.2. The regularized T-IB model consistently outperforms the corresponding unregularized counterpart (T-InfoMax) at the considered time scale.

G Simulation Time

G.1 Molecular Dynamics Simulation

According to the data reported in Shaw et al. (2021), the 64-node Anton 3 supercomputer can simulate up to 250 microseconds per day for a system consisting of $\sim 10^5$ atoms, which is similar to the total atoms in the Villin and Chignolin simulations. On the other hand, a single A100 GPU can simulate only up to 1.5 microseconds each day. The estimate is based on the data reported in Table III in

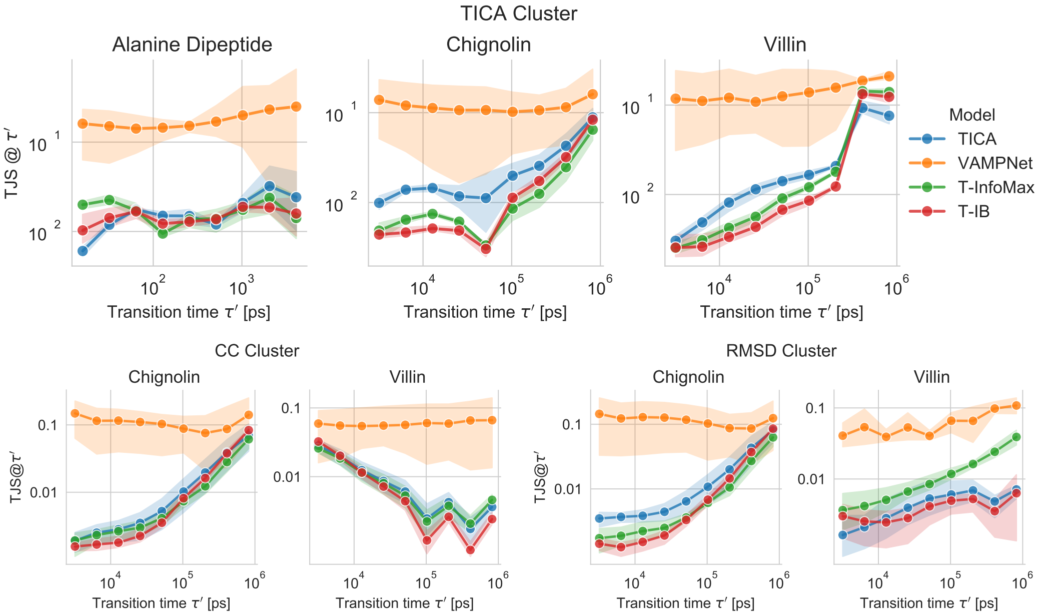


Figure 15: Measure of the average Jensen-Shannon divergence (y-axis) for the unfolded transition matrix for several discrete targets $\mathbf{y}_t$ as a function of the number of unfolding steps (x-axis).

Shaw et al. (2021) and the simulation condition described in the supplementary material provided by Lindorff-Larsen et al. (2011). Therefore, simulating a time jump of $\tau = 3.2$ nanoseconds would require approximately $200s$ on an A100 GPU and about 1 second on Anton 3.

G.2 Latent Simulation

Unfolding one transition step using the Flow++ transition model used in our experiments requires approximately 100 milliseconds on a single A100 GPU. As a result, for the reported Chignolin and Villin experiments, our estimated acceleration is about a factor $\times 1000$ compared to molecular simulations on the same hardware and $\times 10$ for the highly specialized Anton 3 supercomputer. The total simulation time to produce a new molecular simulation of the same length as the training one ($T \sim 100$ microseconds) is approximately 3 months on a single A100 GPU, 1 hour with Latent Simulation on the same GPU, and 10 hours on 64-nodes Anton 3.

Note that total time require to unfold a latent simulation T^{LS} decreases as we increase the lag time τ:

$$T^{LS} = T\, t^{LS}/\tau,$$

in which the cost t^{LS} is determined by the size of the latent space, the transition model, the prediction model, and the hardware. As shown in Table 2, for our experiments, the prediction time is negligible when compared to the cost of unfolding latent transitions. This is because the prediction model consists of a simple MLP. Whenever the target of interest $\mathbf{y}_t$ is also high-dimensional, the prediction cost may increase significantly. However, it is reasonable to assume both t_P^{LS} and t_T^{LS} to require in the order of 100 milliseconds for most tasks of interest.

The Latent Simulation process is also highly parallelizable. As shown in Table 2, it is possible to simultaneously unfold more than 10^5 trajectories on a single A100 GPU with little to no overhead.

It is important to note that learning encoder, transition, and prediction models for Latent Simulation still require several ground truth trajectories of length $T >= \tau$, and the unfolded Latent Simulations are approximations of the molecular dynamics. This is because we do not directly represent the water molecules around the proteins nor the single atoms composing the amino acids.

	10 Trajectories	100 Trajectories	1000 Trajectories	10000 Trajectories
Transition	124 ± 3	128 ± 1	130 ± 1	166 ± 1
Prediction	0.690 ± 0.001	0.700 ± 0.001	0.732 ± 0.002	0.739 ± 0.003

Table 2: Estimations for the time (in milliseconds) required to unfold one step of parallel Latent Simulation of Chignoling and Villin on a single A100 GPU. The estimates refer to the time required to produce samples from the conditional distribution $q_\phi(\mathbf{z}_{t+\tau}|\mathbf{z}_t)$ and $q_\psi(\mathbf{y}_t|\mathbf{z}_t)$ for given $\mathbf{z}_t$. Note that the cost of conditioning and sampling the transition model dominates the one of making predictions. Details on the architectures for the transition prediction models are described in Appendix E.2.

	TICA $[s]$	VAMPnet $[10^3 s]$	T-InfoMax $[10^3 s]$	T-IB $[10^3 s]$
Alanine Dipeptide	1.04 ± 0.01	2.39 ± 0.02	2.63 ± 0.08	2.78 ± 0.02
Chignolin	42.2 ± 0.2	2.8 ± 0.3	3.0 ± 0.3	3.5 ± 0.3
Villin	60 ± 1	12.5 ± 0.3	12.8 ± 0.2	13.3 ± 0.4

Table 3: Training time required to train the encoder architectures on the Alanine Dipeptide, Chignolin, and Villin data. The measurements are reported in seconds for the TICA experiments, and 10^3 seconds for the other models relying on TorchMD encoders.

Data	Training Time $[10^3 s]$
Alanine Dipeptide	1.7 ± 0.1
Chignolin	4.5 ± 0.3
Villin	4.5 ± 0.5

Table 4: Estimated training time required to fit the transition and predictive model for a fixed representation. The estimates also include the time required to unroll and evaluate latent simulation for validation purposes.

G.3 Training time

We report the training time corresponding to all the models reported in our experimental section by differentiating the time required to train the encoder (step 1) from the training of transition and prediction model (step 2) described in Section 2.1.

Table 3 reports the total time required to train the encoder architectures with the TICA, VAMPnet, T-InfoMax and T-IB objectives. The training time for TICA is substantially shorter since it relies on linear mapping instead of a flexible TorchMD architecture. The variance of the time estimates is computed over three runs per experiment.

The training time for the second step is equivalent for all models since the same transition and prediction architecture are fit to each representation using maximum likelihood. Train time is not influenced by the encoder (linear or Deep NN) since we encode and store the entire dataset to disk at the end of step 1. As a result, the total cost depends solely on the dataset size and size of the latent representation, as reported in Table 4.

The total training time (step 1 + step 2) for the T-IB model on Villin amounts to approximately 5 hours. Unfolding a latent simulation of the same length of the training trajectory requires another hour, bringing the total to 6 hours. Even by including the training time, Latent Simulation is $100\times$ to $1000\times$ faster than running molecular dynamics on comparable hardware.