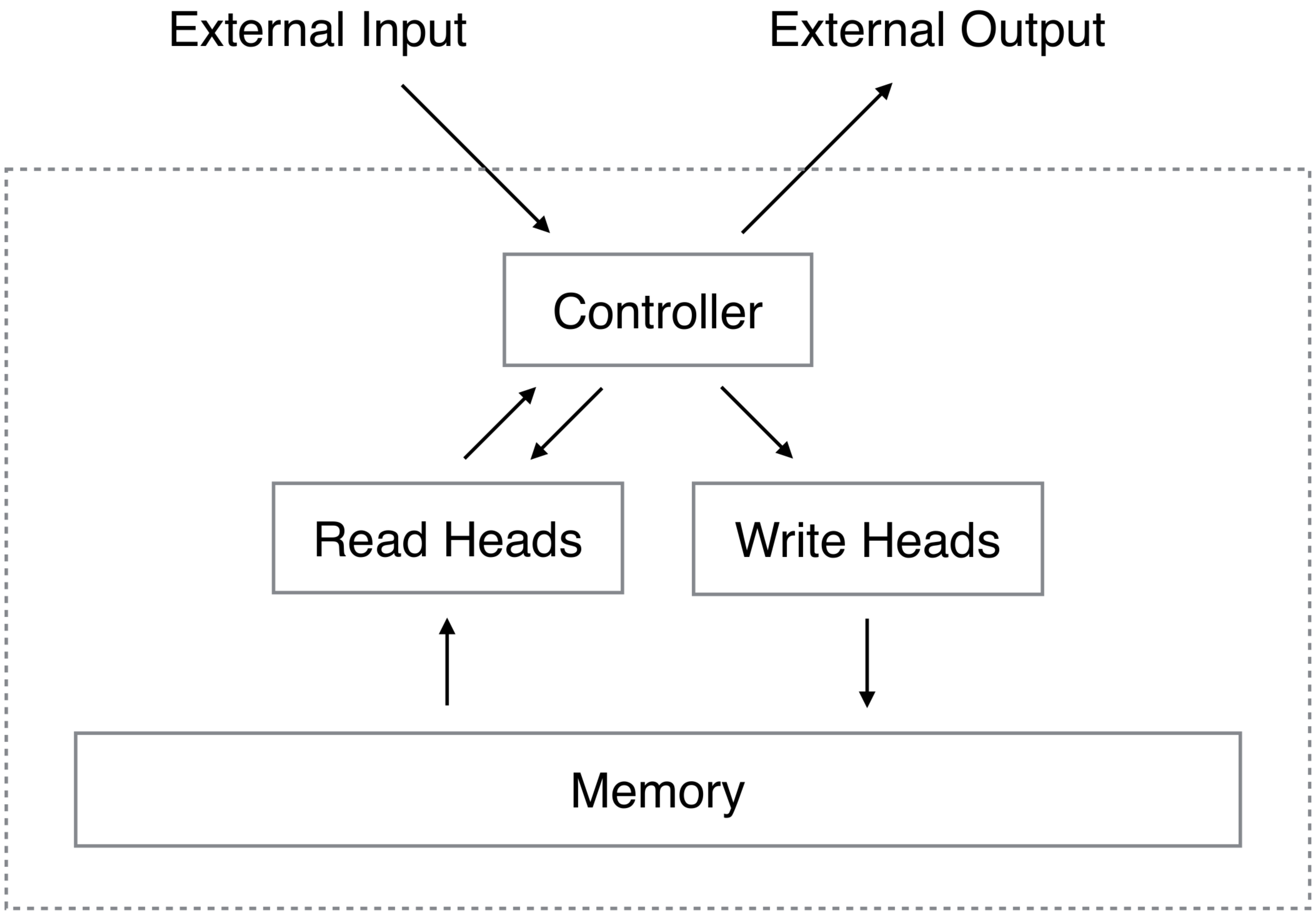
# The Kanerva Machine

April 19, 2019

Daniil Polykovskiy
CMC MSU; Insilico Medicine

# Neural Computers



- Controller (RNN or fully-connected) has access to external memory

- On each iteration, controller writes some data into the memory and then reads from it

# Memory Model

- Memory is a matrix $M \in \mathbb{R}^{K \times C}$

- Models learn useful read/write patterns

| |
|---|
| M[1] |
| M[2] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[K] |

# Read Weights

- Controller produces a *key* $k$
- Generate wights $v_i = K[k, M[i]]$
- Where $K[u, v] = \dfrac{u^T v}{\|u\| \cdot \|v\|}$
- This recalls associative arrays

| |
|---|
| M[1] |
| M[2] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[K] |

# Reading From The Memory

- Controller produces a temperature $\beta$

- Normalize the distribution with softmax:

$$w_i = \frac{e^{\beta v_i}}{\sum_{j=1}^{K} e^{\beta v_j}}$$

- Return a weighted sum over the memory elements

| | |
|---|---|
| M[1] | 0 |
| **M[2]** | **0.3** |
| M[…] | 0 |
| M[…] | 0 |
| M[…] | 0 |
| M[…] | 0 |
| M[…] | 0 |
| M[…] | 0 |
| M[…] | 0 |
| **M[15]** | **0.5** |
| M[…] | 0 |
| M[…] | 0 |
| **M[K]** | **0.2** |

$$r = 0.3 \cdot M[2] + 0.5 \cdot M[15] + 0.2 \cdot M[K]$$

# Writing To The Memory

- Controller produces erase $e_t$ and addition $a_t$ vectors and weights $\widetilde{w}_t$

- Update the memory with

$$M_t[i] = M_{t-1}[i] \cdot (1 - \widetilde{w}_t[i]e_t) + \widetilde{w}_t[i]a_t$$

Erase

Add

| |
|---|
| M[1] |
| M[2] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[…] |
| M[K] |

# Key Idea 1: Memory model

# Key Idea 1: Memory model

- **Slot-based memory (read/write to one row):**
  - Can collapse to read/write operations with few rows
  - Or stores each object in its own row

# Key Idea 1: Memory model

- **Slot-based memory (read/write to one row):**
    - Can collapse to read/write operations with few rows
    - Or stores each object in its own row
- **Distributed memory (read/write to multiple rows):**
    - Overlapping representations
    - Some rows may encode class-specific representations, others will store object-specific variations

# Key Idea 2: Memory as inference

- Memory is a latent variable
- Writing is inference: $p(M \mid X)$
- Iterative writing: $p(M \mid x_{<T}, x_T) \propto \underset{\substack{\uparrow \\ \textbf{Already computed}}}{p(M \mid X_{<T})}\,p(x_T \mid M)$

# Sparse Distributed Memory

Model works only with binary (-1, 1) vector data, contains: A—table of addresses (fixed), M—memory

$$w_k = \begin{cases} 1, & h(x, A_k) \leqslant \tau \\ 0, & \textbf{otherwise} \end{cases}$$

- **Reading**:

$$\widehat{x_i} = \begin{cases} 1, & \sum_{k=1}^{K} w_k M_{k,i} > 0 \\ -1, & \textbf{otherwise} \end{cases}$$

- **Writing:**

$$M_k \leftarrow M_{k-1} + w_k x$$

# Sparse Distributed Memory

Model works only with binary (-1, 1) vector data, contains: A—table of addresses (fixed), M—memory

$$w_k = \begin{cases} 1, & h(x, A_k) \leqslant \tau \\ 0, & \textbf{otherwise} \end{cases}$$

- **Reading**:

$$\widehat{x_i} = \begin{cases} 1, & \sum_{k=1}^{K} w_k M_{k,i} > 0 \\ -1, & \textbf{otherwise} \end{cases}$$

- **Writing:**

$$M_k \leftarrow M_{k-1} + w_k x$$

**Application**:
Denoising with iterative queries

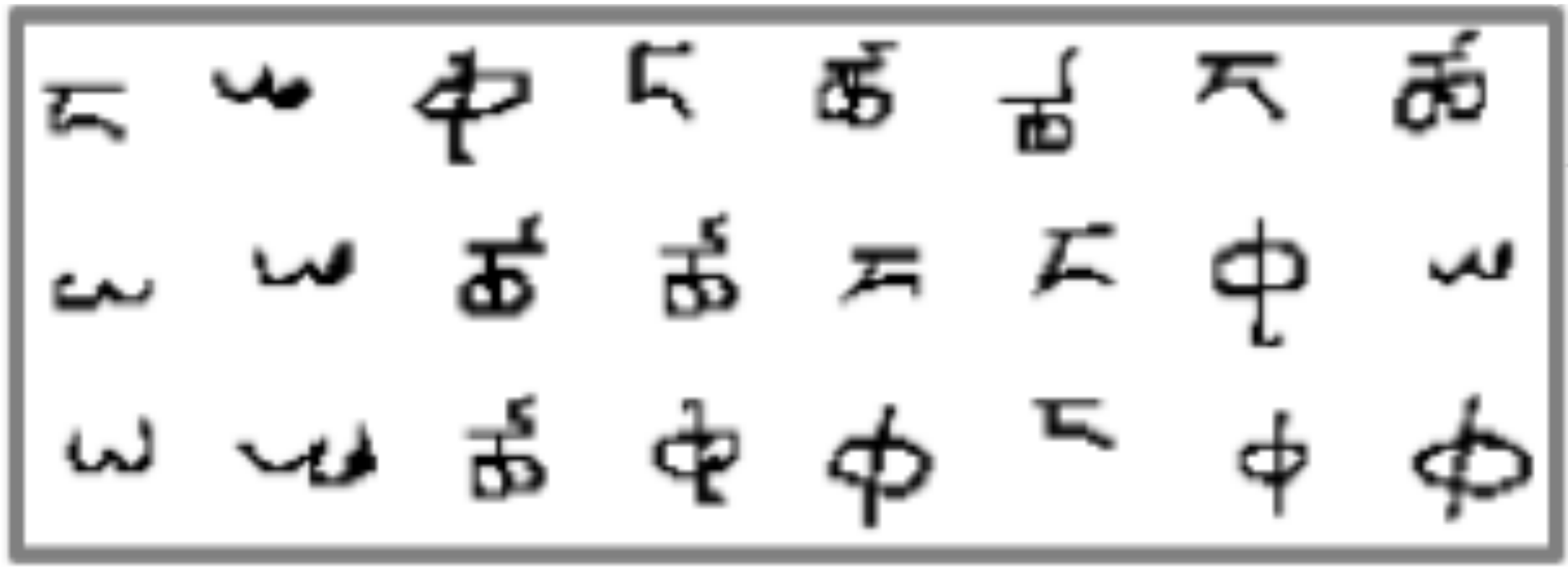
# The Kanerva Machine: A Generative Distributed Memory

Yan Wu, Greg Wayne, Alex Graves, Timothy Lillicrap

# The Kanerva Machine

- Few-shot learning task: store an *exchangeable episode* and recall all stored patterns

$X = \{x_1, x_2, \ldots, x_T\}$



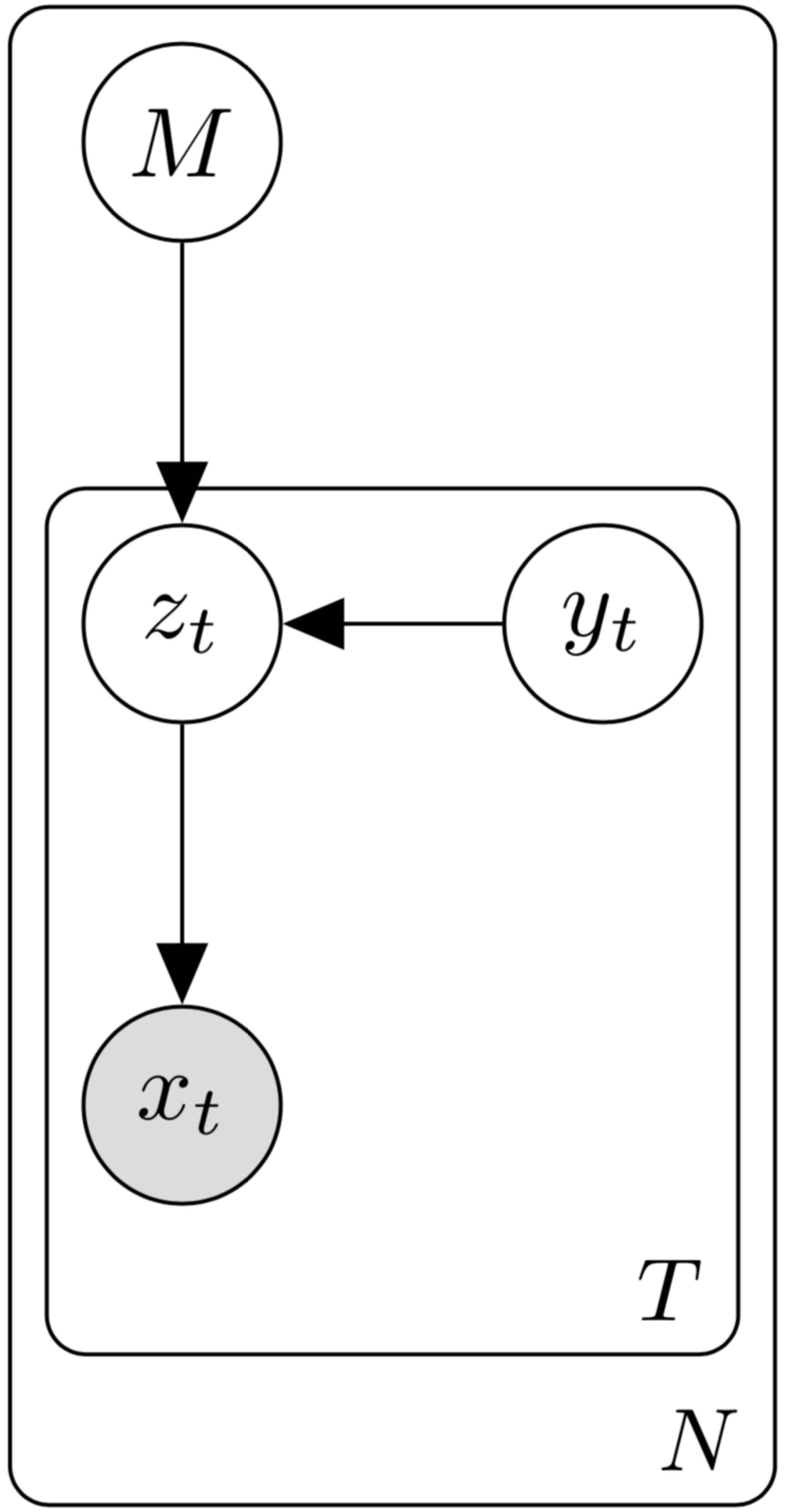
# The Kanerva Machine

$$p_\theta(X, Y, Z \,|\, M) = \prod_{t=1}^{T} p_\theta(x_t, y_t, z_t \,|\, M) = \prod_{t=1}^{T} p_\theta(x_t \,|\, z_t)\, p_\theta(z_t \,|\, y_t, M)\, p_\theta(y_t)$$

Read-outs

Data

Addresses



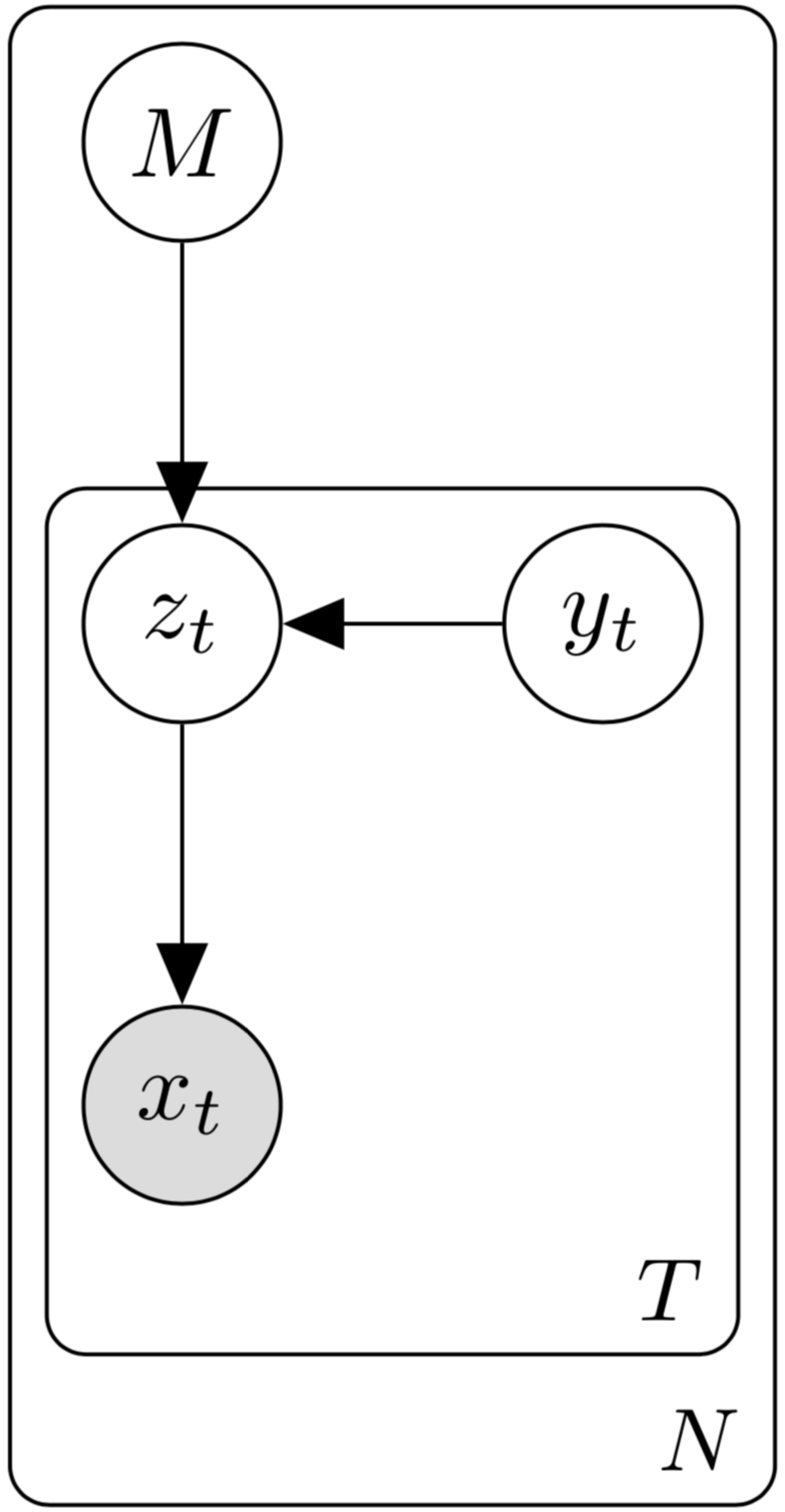
# The Kanerva Machine

$$p_\theta(X, Y, Z \,|\, M) = \prod_{t=1}^{T} p_\theta(x_t, y_t, z_t \,|\, M) = \prod_{t=1}^{T} p_\theta(x_t \,|\, z_t) p_\theta(z_t \,|\, y_t, M) p_\theta(y_t)$$

**Neural network**

???

$\mathcal{N}(0,I)$



# The Kanerva Machine

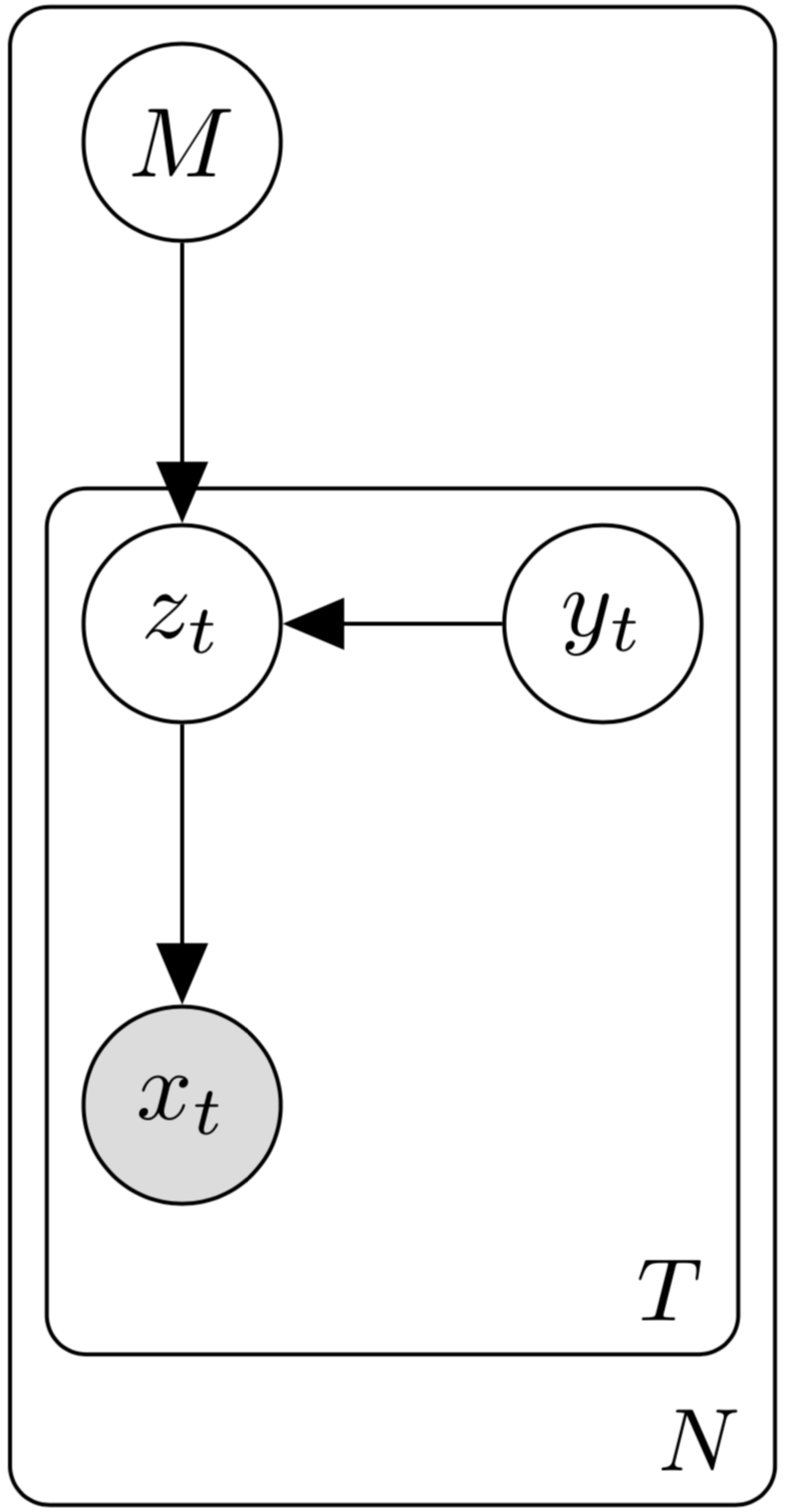
$$p_\theta(X, Y, Z \,|\, M) = \prod_{t=1}^{T} p_\theta(x_t, y_t, z_t \,|\, M) = \prod_{t=1}^{T} p_\theta(x_t \,|\, z_t) p_\theta(z_t \,|\, y_t, M) p_\theta(y_t)$$

**Neural network**

$\mathcal{N}(0,I)$

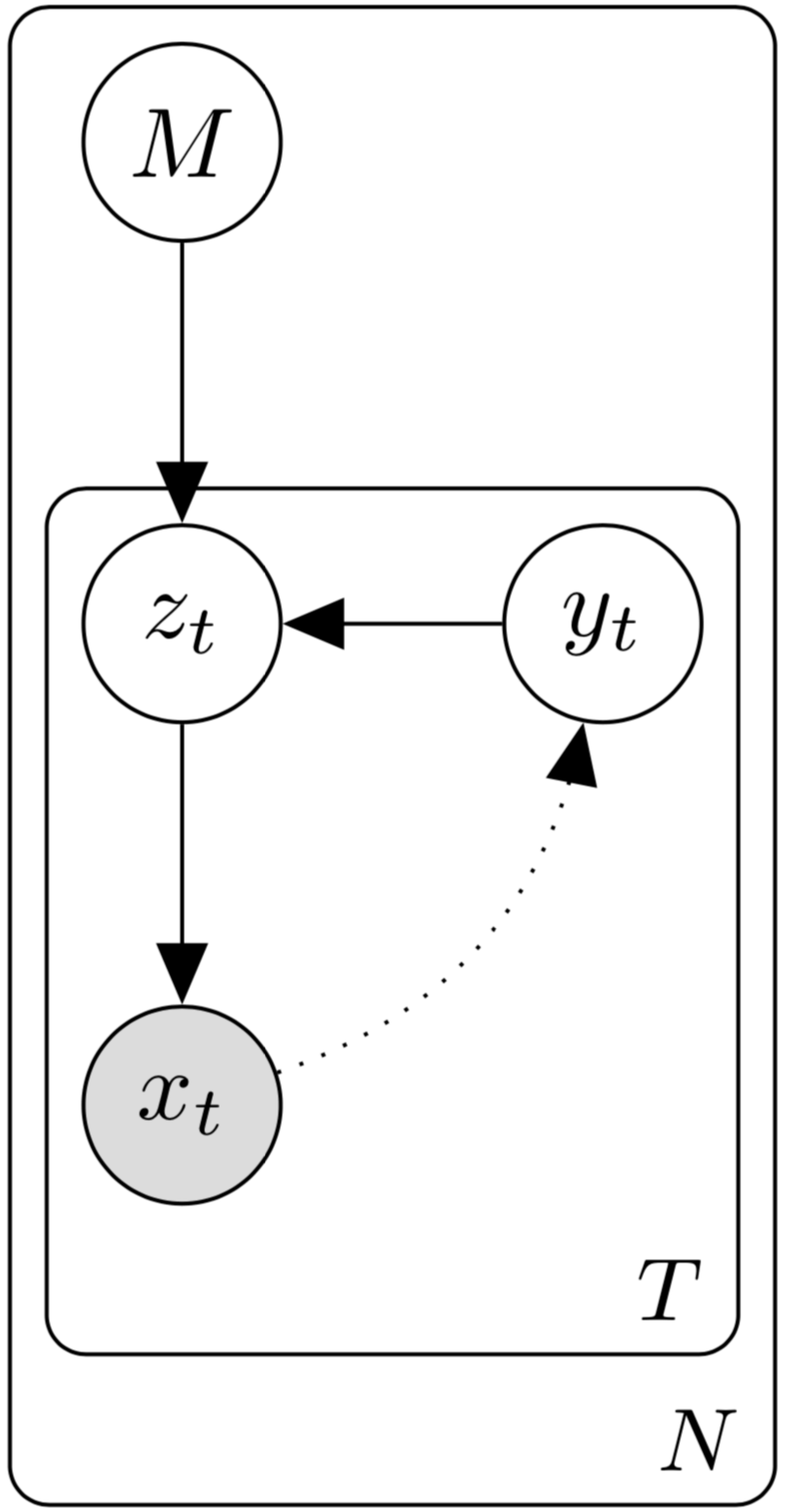
$$w_t = f^T(y_t) \cdot A$$

$$p_\theta(z_t \,|\, y_t, M) = \mathcal{N}\left(z_t \mid w_t^T \cdot M, \sigma^2 I\right)$$



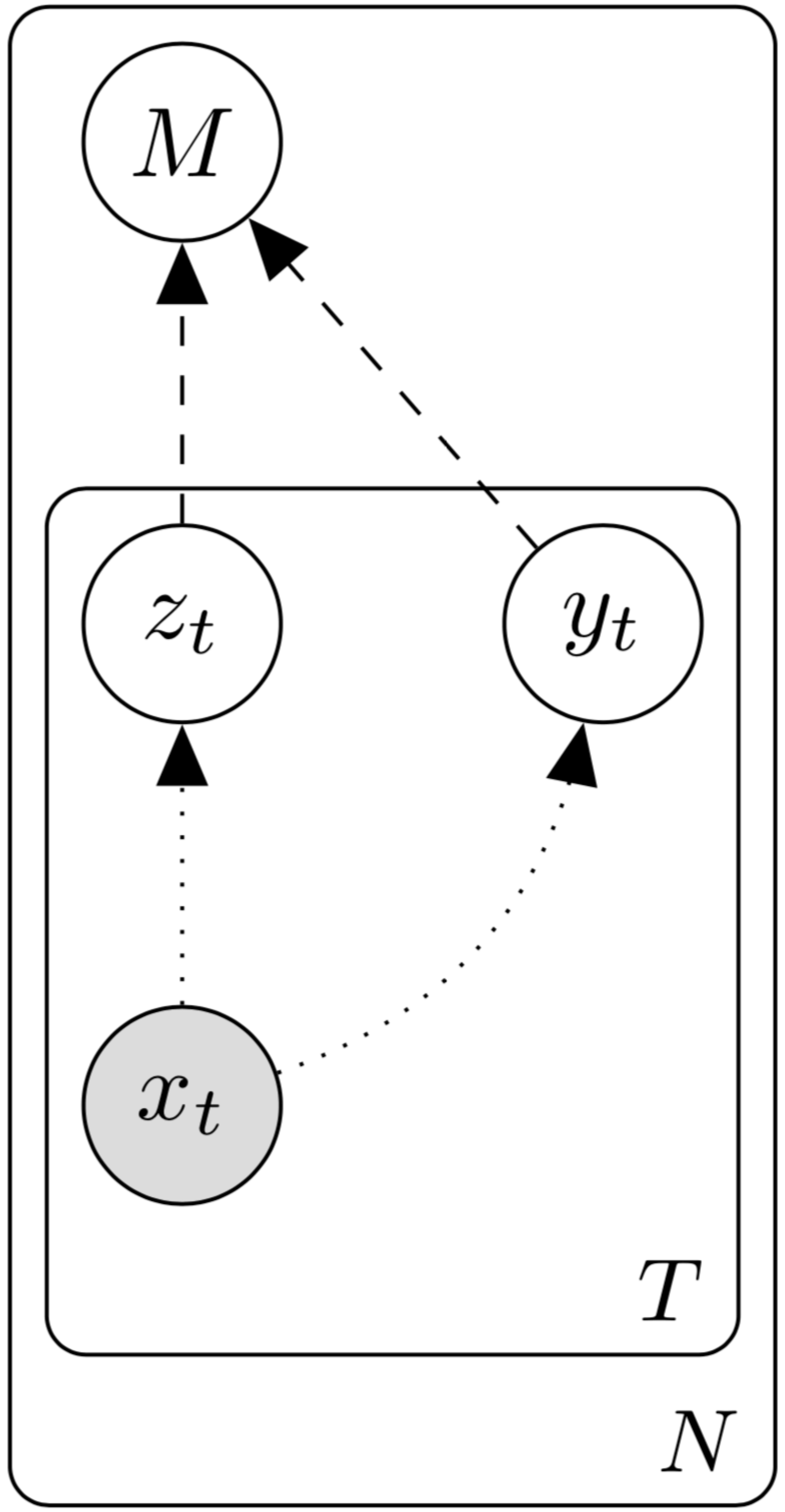
# Reading inference

$$q_\phi(Y, Z \mid X, M) = \prod_{t=1}^{T} q_\phi(y_t, z_t \mid x_t, M) = \prod_{t=1}^{T} q_\phi(z_t \mid x_t, y_t, M)\, q_\phi(y_t \mid x_t)$$



# Writing inference

$$
\begin{aligned}
q_\phi(M\,|\,X) &= \int p_\theta(M, Y, Z\,|\,X)\mathrm{d}Z\mathrm{d}Y \\
&= \int p_\theta(M\,|\,\{y_1, \ldots, y_T\}, \{z_1, \ldots, z_T\}) \prod_{t=1}^{T} q_\phi(z_t\,|\,x_t) q_\phi(y_t\,|\,x_t)\mathrm{d}z_t\mathrm{d}y_t \\
&\approx p_\theta(M\,|\,\{y_1, \ldots, y_T\}, \{z_1, \ldots, z_T\})\Big|_{y_t \sim q_\phi(y_t|x_t), z_t \sim q_\phi(z_t|x_t)}
\end{aligned}
$$

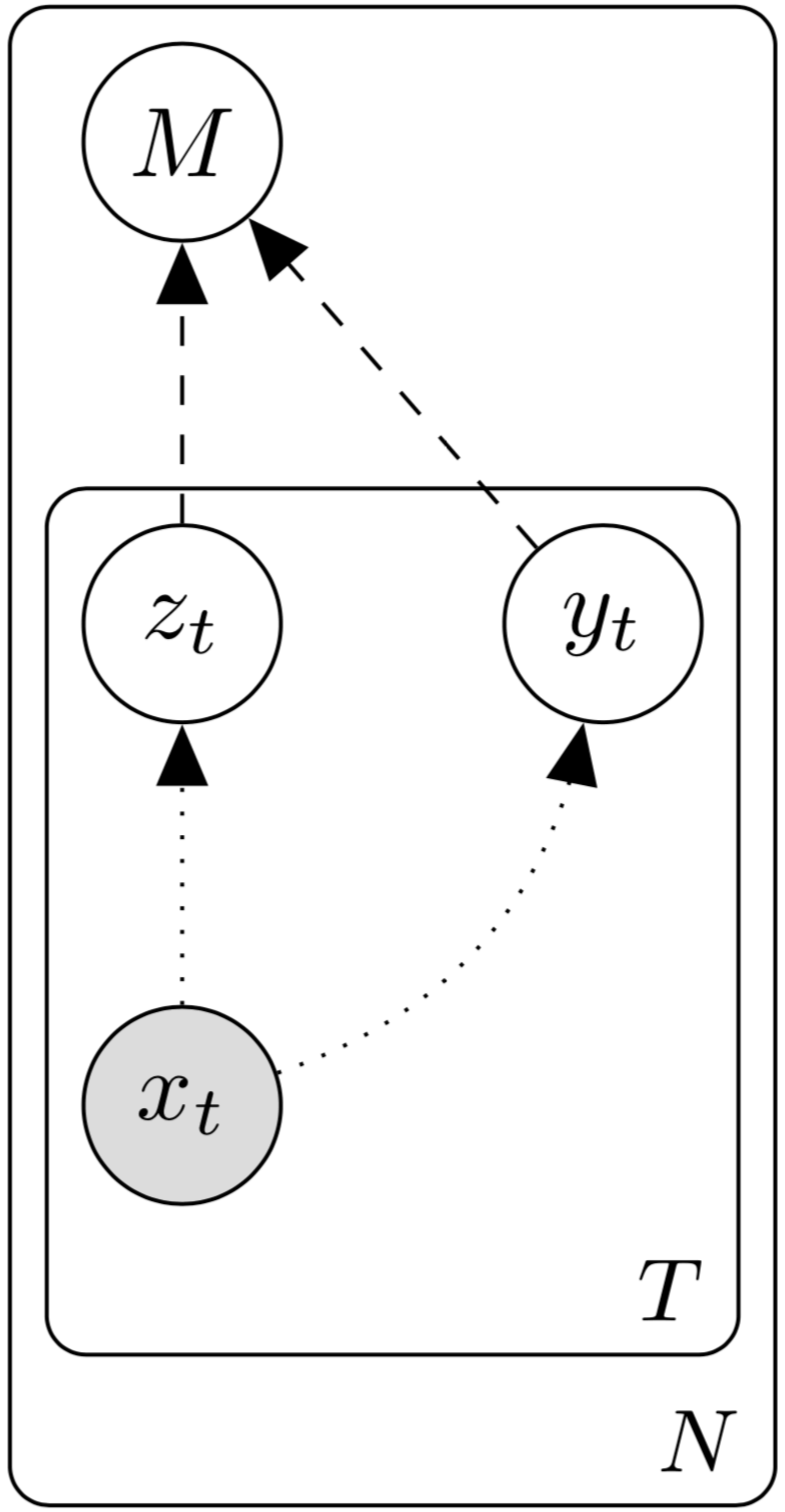


# Writing inference

$$p_\theta(M \mid Y, Z) = ?$$

$$w_t = f^T(y_t) \cdot A$$

$$p_\theta(z_t \,|\, y_t, M) = \mathcal{N}\left(z_t \mid w_t^T \cdot M, \sigma^2 I\right)$$



# Distribution over matrices

- Matrix normal distribution

$$p(M) = \mathscr{M}\mathscr{N}(\overset{\textbf{Mean}}{R}, \underset{\textbf{Covariance of rows}}{U}, \underset{\textbf{Covariance of columns}}{V})$$

# Distribution over matrices

- Matrix normal distribution

**Mean**

$$p(M) = \mathscr{M}\mathscr{N}(R, U, V) \quad \Leftrightarrow \quad p\left(\mathrm{vec}(M)\right) = \mathscr{N}(\mathrm{vec}(M) \mid \mathrm{vec}(R), V \otimes U)$$

**Covariance of rows**

**Covariance of columns**

# Distribution over matrices

- Matrix normal distribution

$$p(M) = \mathscr{M}\mathscr{N}(R, U, V) \propto \exp\left(-\frac{1}{2}Tr\left(V^{-1}\,(X-R)^T\,U^{-1}(X-R\right)\right)$$

$$= \exp\left(-\frac{1}{2}\left\langle (X-R)V^{-1}, U^{-1}(X-R)\right\rangle\right)$$

$V = I$ **— no covariance between columns**

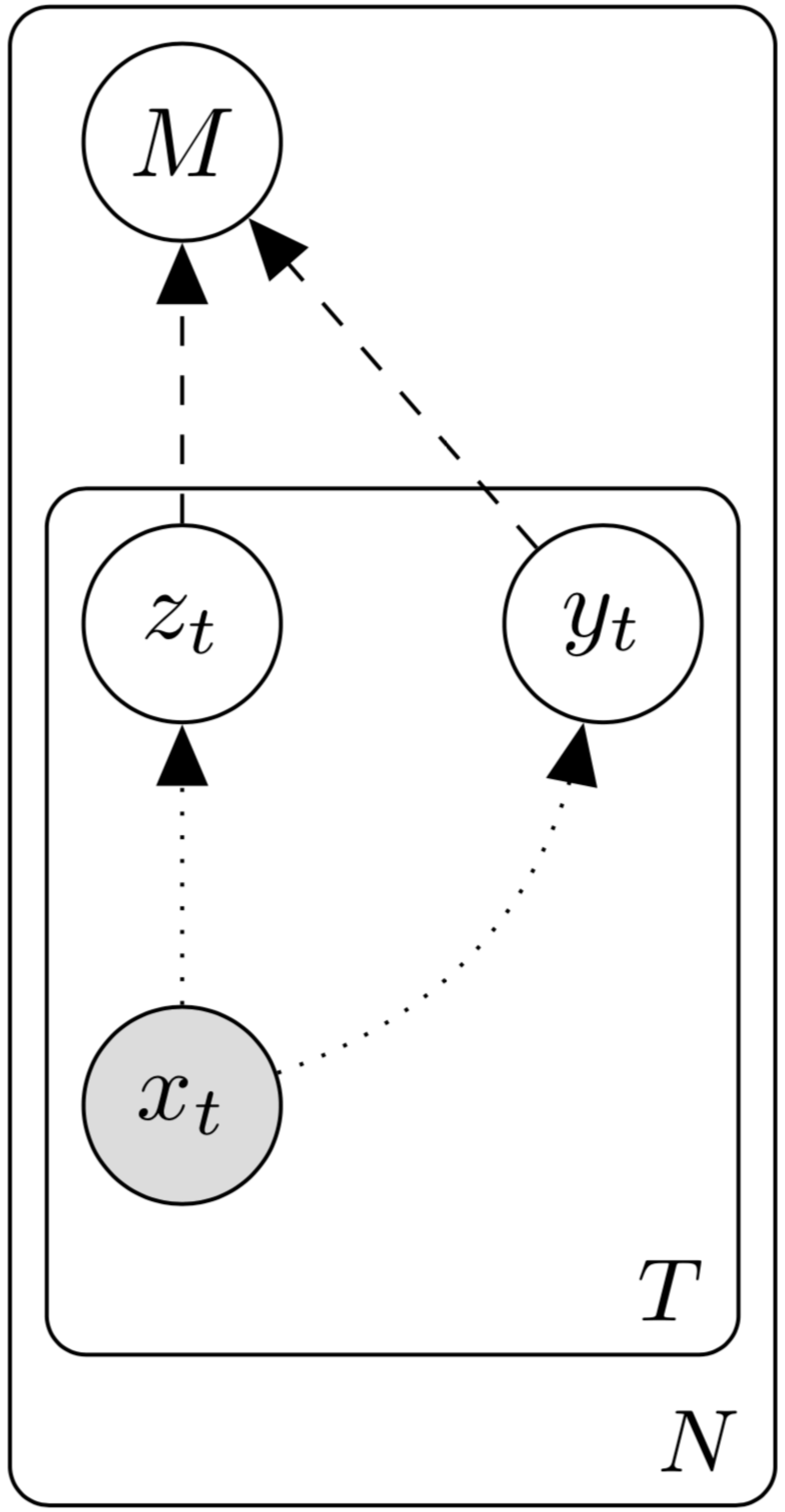
# Writing inference

$$p_\theta(M \mid Y, Z) = ?$$

$$w_t = f^T(y_t) \cdot A$$

$$p_\theta(z_t | y_t, M) = \mathcal{N}\left(z_t \mid w_t^T \cdot M, \sigma^2 I\right)$$

$$p(M) = \mathscr{M}\mathcal{N}(R, U, V)$$



# Writing inference

$$p_\theta(M \mid Y, Z) = ?$$

$$w_t = f^T(y_t) \cdot A$$

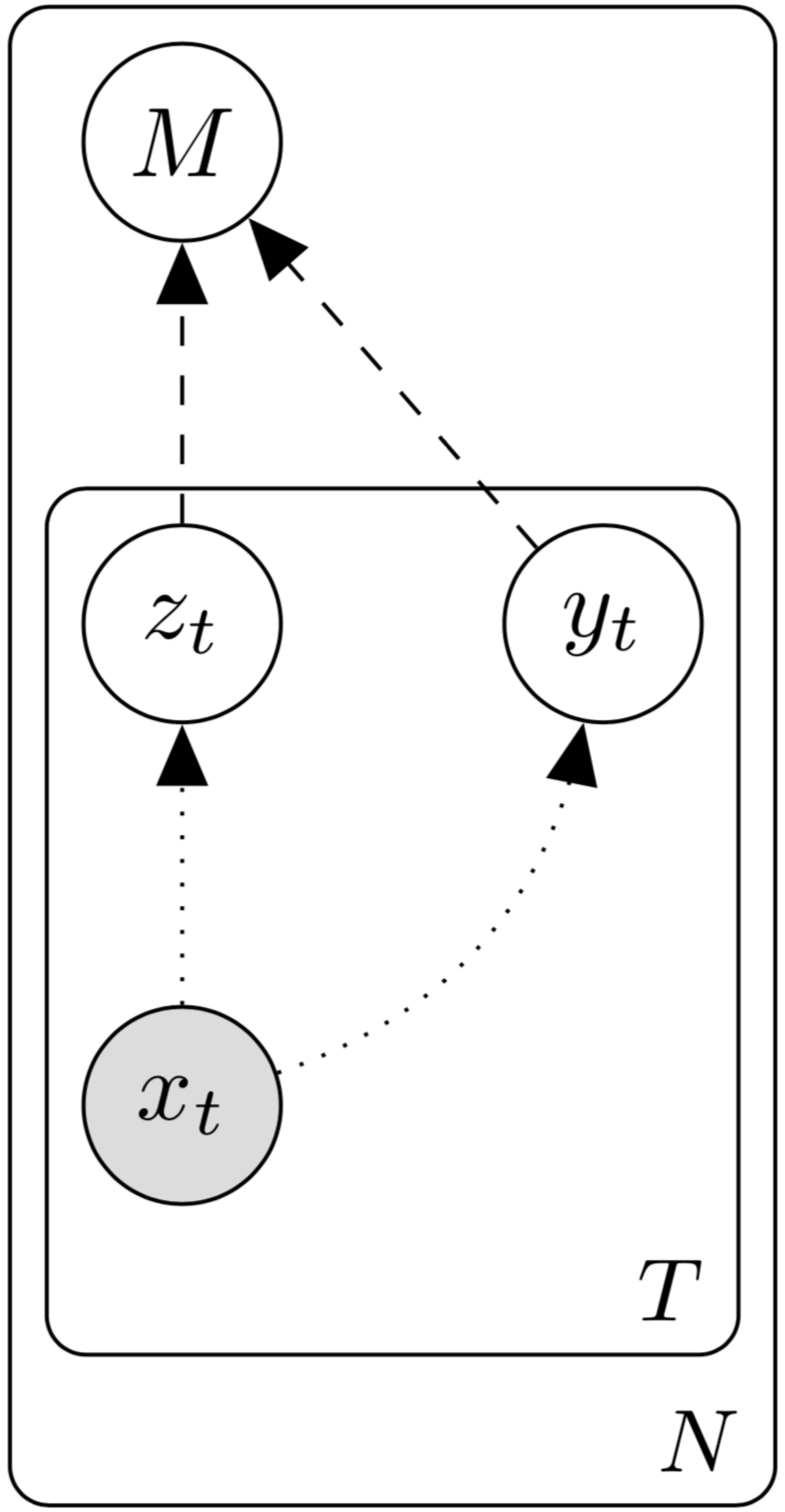
$$p_\theta(z_t \mid y_t, M) = \mathcal{N}\left(z_t \mid w_t^T \cdot M, \sigma^2 I\right)$$

$$p(M) = \mathcal{MN}(R, U, V)$$

$$\Delta \leftarrow Z - W R$$

$$\Sigma_c \leftarrow W U \qquad \Sigma_z \leftarrow W U W^T + \Sigma_\xi$$

$$R \leftarrow R + \Sigma_c^T \Sigma_z^{-1} \Delta \qquad U \leftarrow U - \Sigma_c^T \Sigma_z^{-1} \Sigma_c$$



# Writing inference

K x C

$$p_\theta(M \mid Y, Z) = \,?$$

$$\boxed{\begin{aligned} & w_t = f^T(y_t) \cdot A \\ & p_\theta(z_t \,|\, y_t, M) = \mathcal{N}\left(z_t \mid w_t^T \cdot M, \sigma^2 I\right) \end{aligned}}$$

$$p(M) = \mathscr{M}\mathcal{N}(R, U, V)$$

K x K

K x C

$$\Delta \leftarrow Z - W R$$

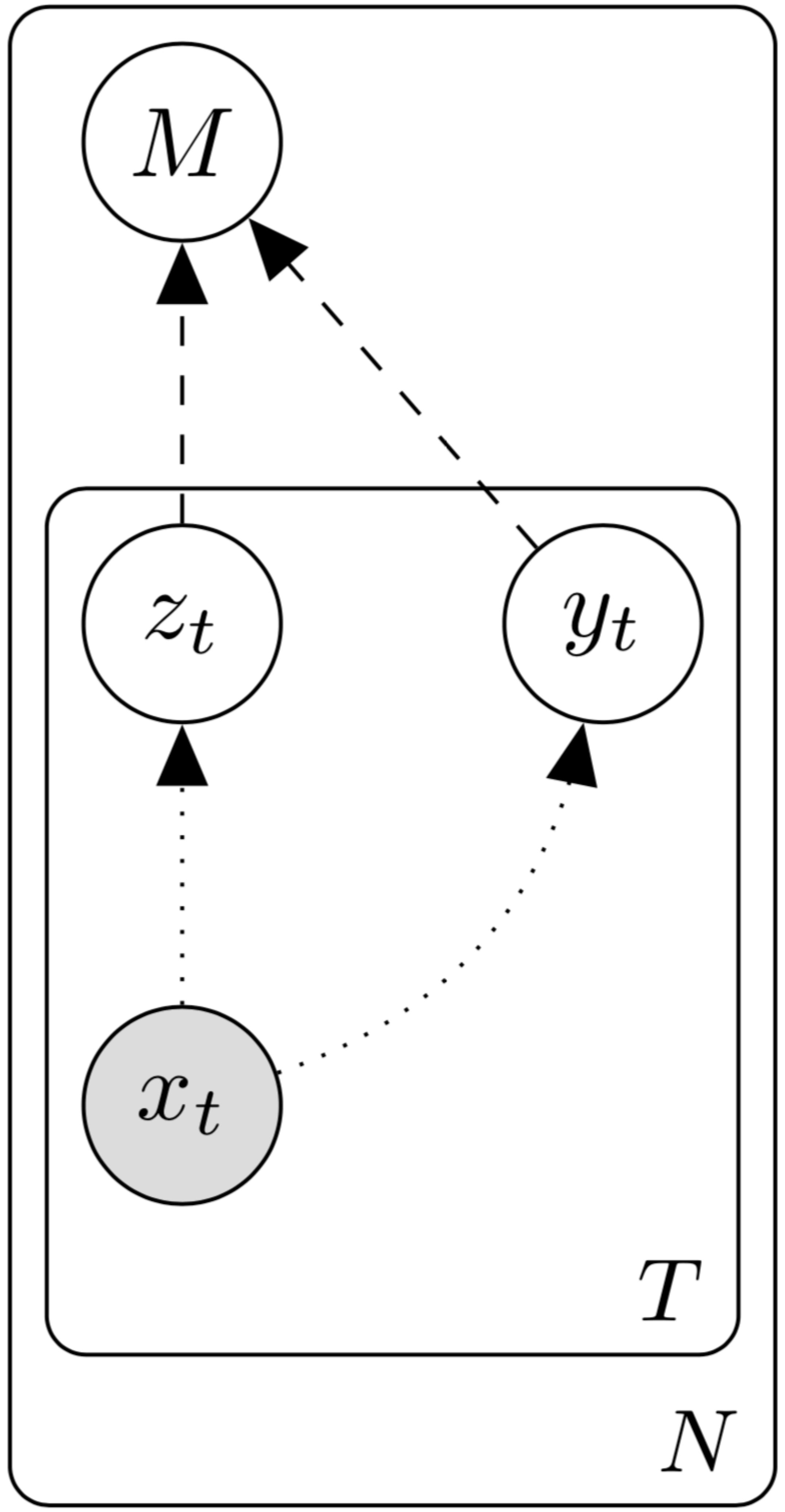
T x K

$$\Sigma_c \leftarrow W U$$

$$R \leftarrow R + \Sigma_c^T \Sigma_z^{-1} \Delta$$

$$\Sigma_z \leftarrow W U W^T + \Sigma_\xi$$

$$U \leftarrow U - \Sigma_c^T \Sigma_z^{-1} \Sigma_c$$



# Writing inference

K x C

$$p_\theta(M \mid Y, Z) = \,?$$

$$\boxed{\begin{aligned} w_t &= f^T(y_t) \cdot A \\ p_\theta(z_t \mid y_t, M) &= \mathcal{N}\left(z_t \mid w_t^T \cdot M, \sigma^2 I\right) \end{aligned}}$$

$$p(M) = \mathcal{M}\mathcal{N}(R, U, V)$$

K x C

K x K

$$\Delta \leftarrow Z - W R$$

T x K

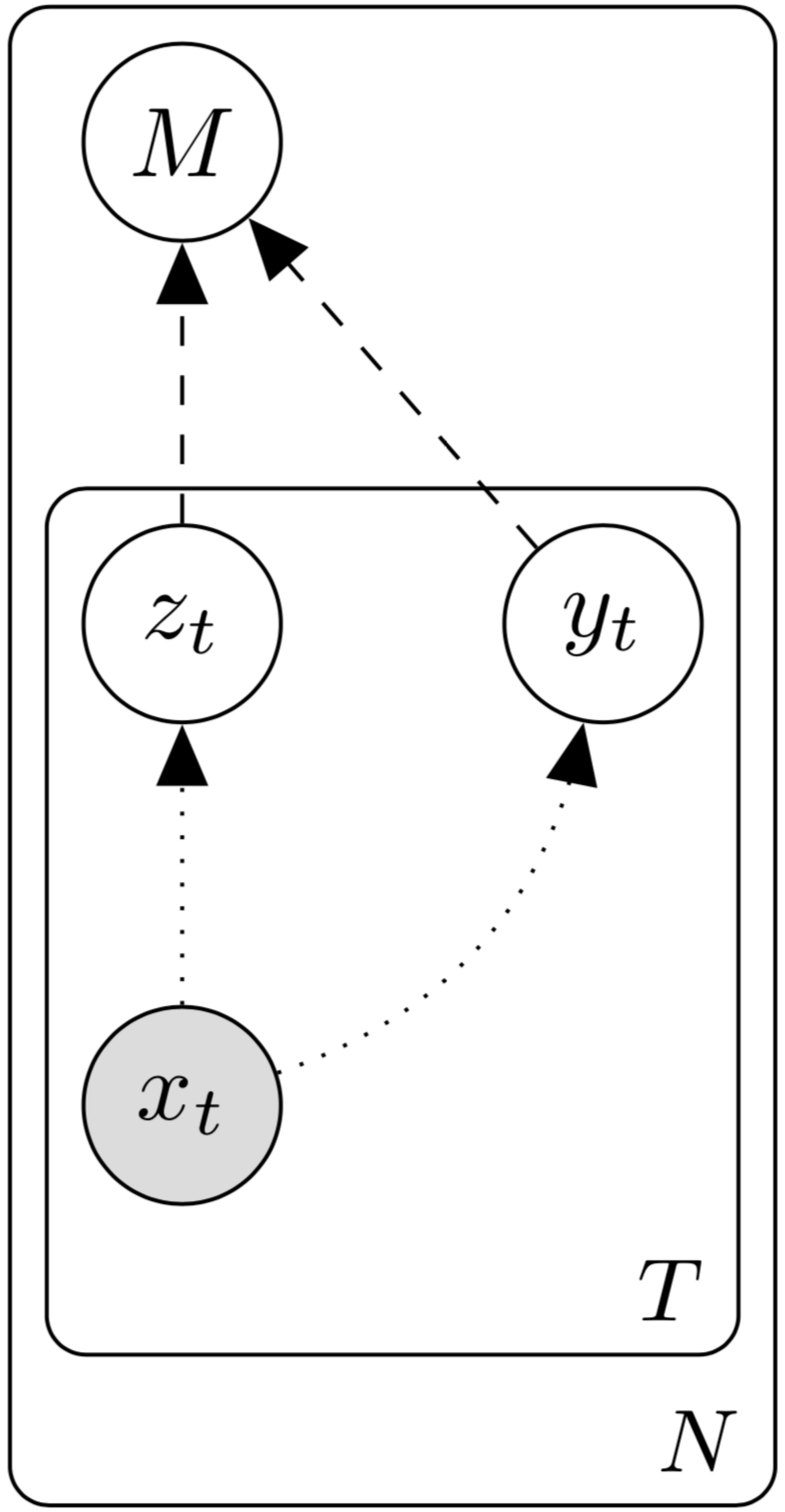
$$\Sigma_c \leftarrow W U$$

$$R \leftarrow R + \Sigma_c^T \Sigma_z^{-1} \Delta$$

$$\Sigma_z \leftarrow W U W^T + \Sigma_\xi$$

$$U \leftarrow U - \Sigma_c^T \Sigma_z^{-1} \Sigma_c$$

T x T



# Writing inference

K x C

$$p_\theta(M \mid Y, Z) = \,?$$

$$w_t = f^T(y_t) \cdot A$$

$$p_\theta(z_t \mid y_t, M) = \mathcal{N}\left(z_t \mid w_t^T \cdot M, \sigma^2 I\right)$$

$$p(M) = \mathcal{MN}(R, U, V)$$

K x K

K x C

$$\Delta \leftarrow Z - W R$$

T x K

$$\Sigma_c \leftarrow W U$$

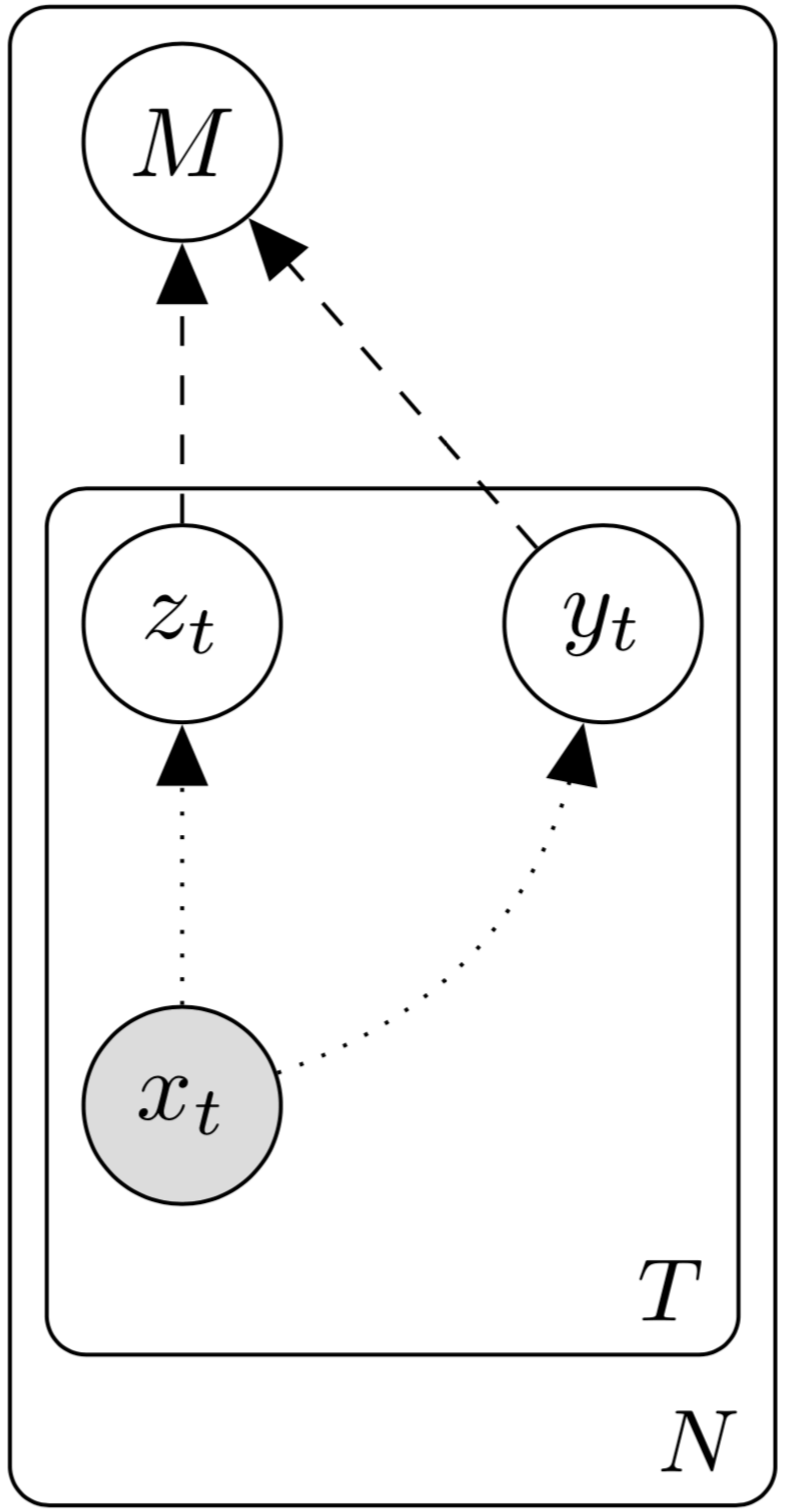
$$R \leftarrow R + \Sigma_c^T \Sigma_z^{-1} \Delta$$

$$\Sigma_z \leftarrow W U W^T + \Sigma_\xi$$

$$U \leftarrow U - \Sigma_c^T \Sigma_z^{-1} \Sigma_c$$

T x T

**Iterative writing reduces complexity!**



# Training

$$\mathscr{J} = \text{const} + \mathbb{E}_{p(X)p(M|X)} \sum_{t=1}^{T} \log p_\theta(x_t \,|\, M)\mathrm{d}M\mathrm{d}X \geq \text{const} + \mathscr{L}$$

$$\mathscr{L} = \mathbb{E}_{q_\phi(M|X)p(X)} \sum_{t=1}^{T} \Big\{ \mathbb{E}_{q_\phi(y_t,z_t|x_t,M)} \log p_\theta(x_t \,|\, z_t)$$

$$-\mathrm{KL}(q_\phi(y_t \,|\, x_t) \| p_\theta(y_t)) - \mathrm{KL}(q_\phi(z_t \,|\, x_t, y_t, M) \| p_\phi(z_t \,|\, y_t, M)) \Big\}$$

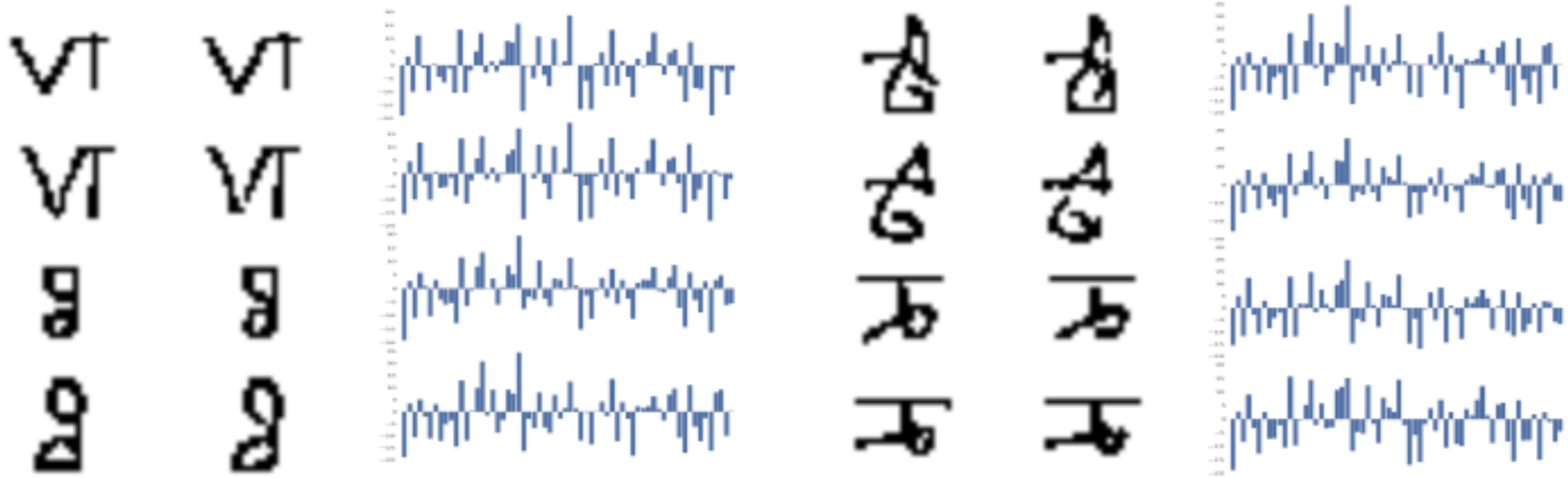
- During training, $q_\phi(M \,|\, X) = \delta(R)$

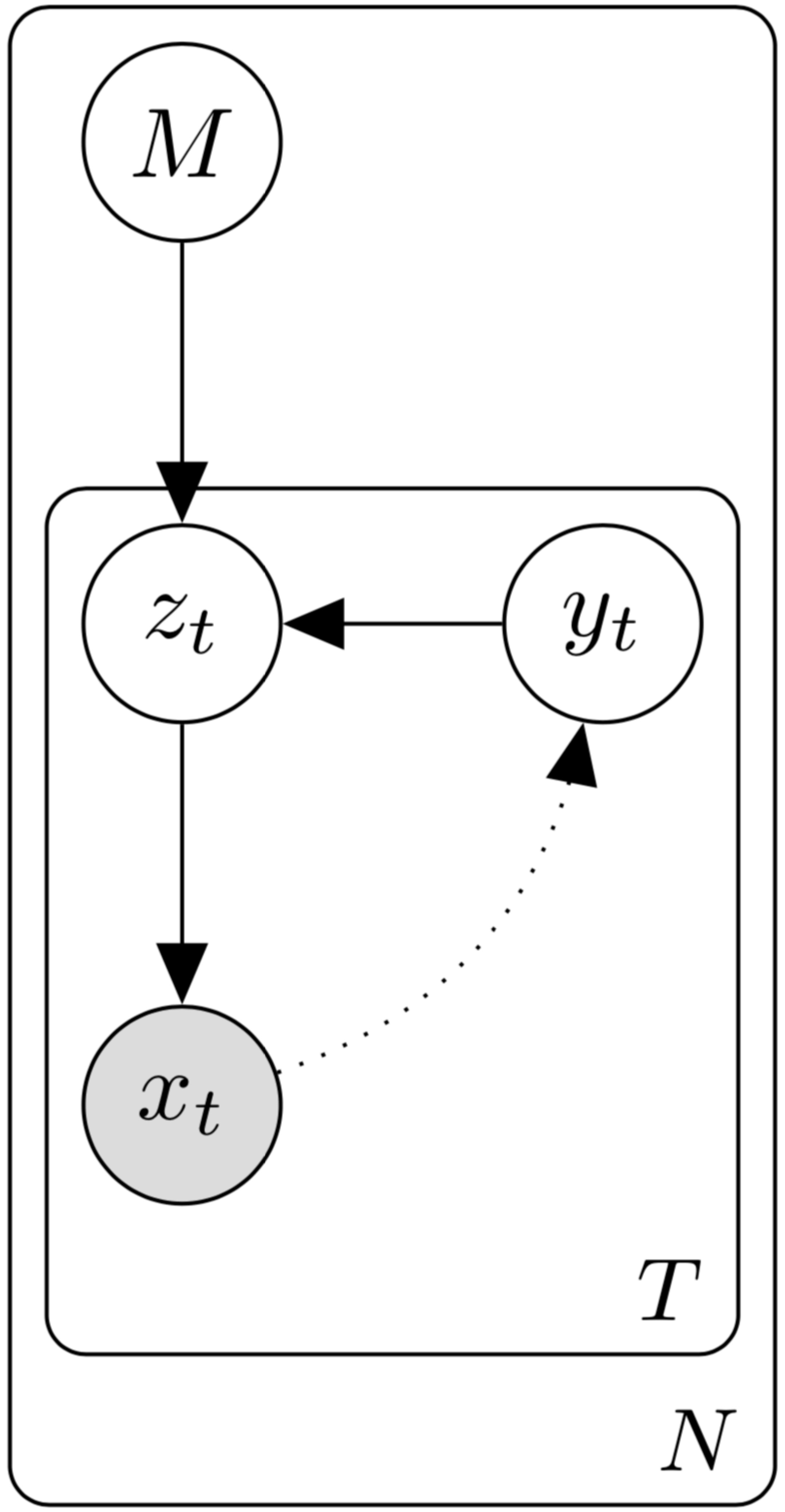
# Experiments

The Kanerva Machine: A Generative Distributed Memory

# Reading

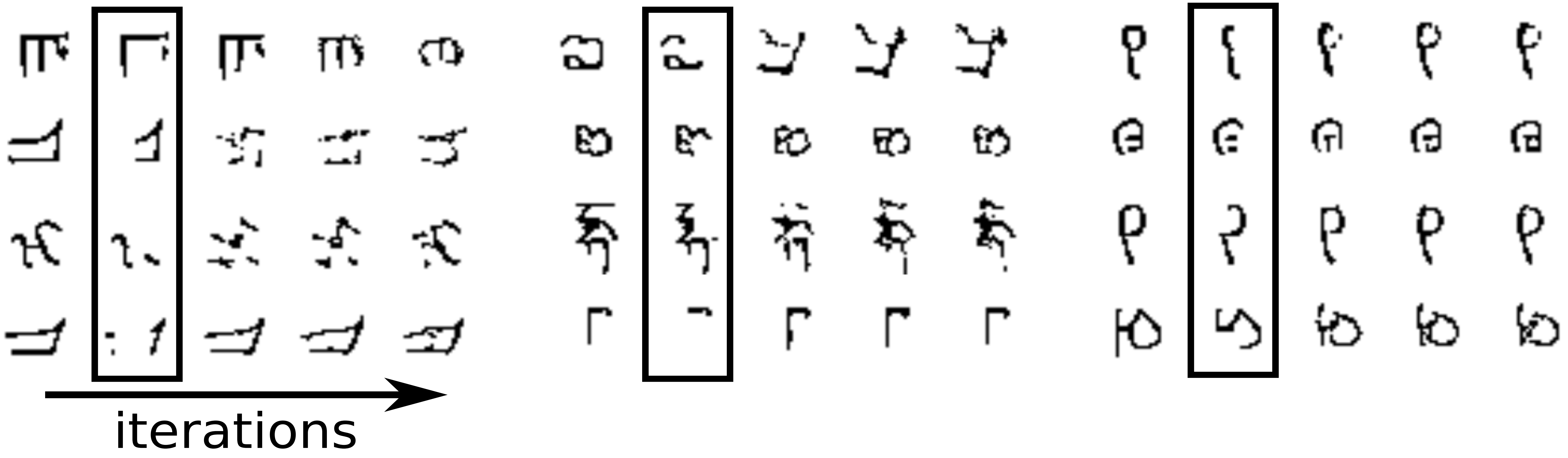
$$q_\phi(Y, Z \mid X, M) = \prod_{t=1}^{T} q_\phi(y_t, z_t \mid x_t, M) = \prod_{t=1}^{T} q_\phi(z_t \mid x_t, y_t, M)\, q_\phi(y_t \mid x_t)$$

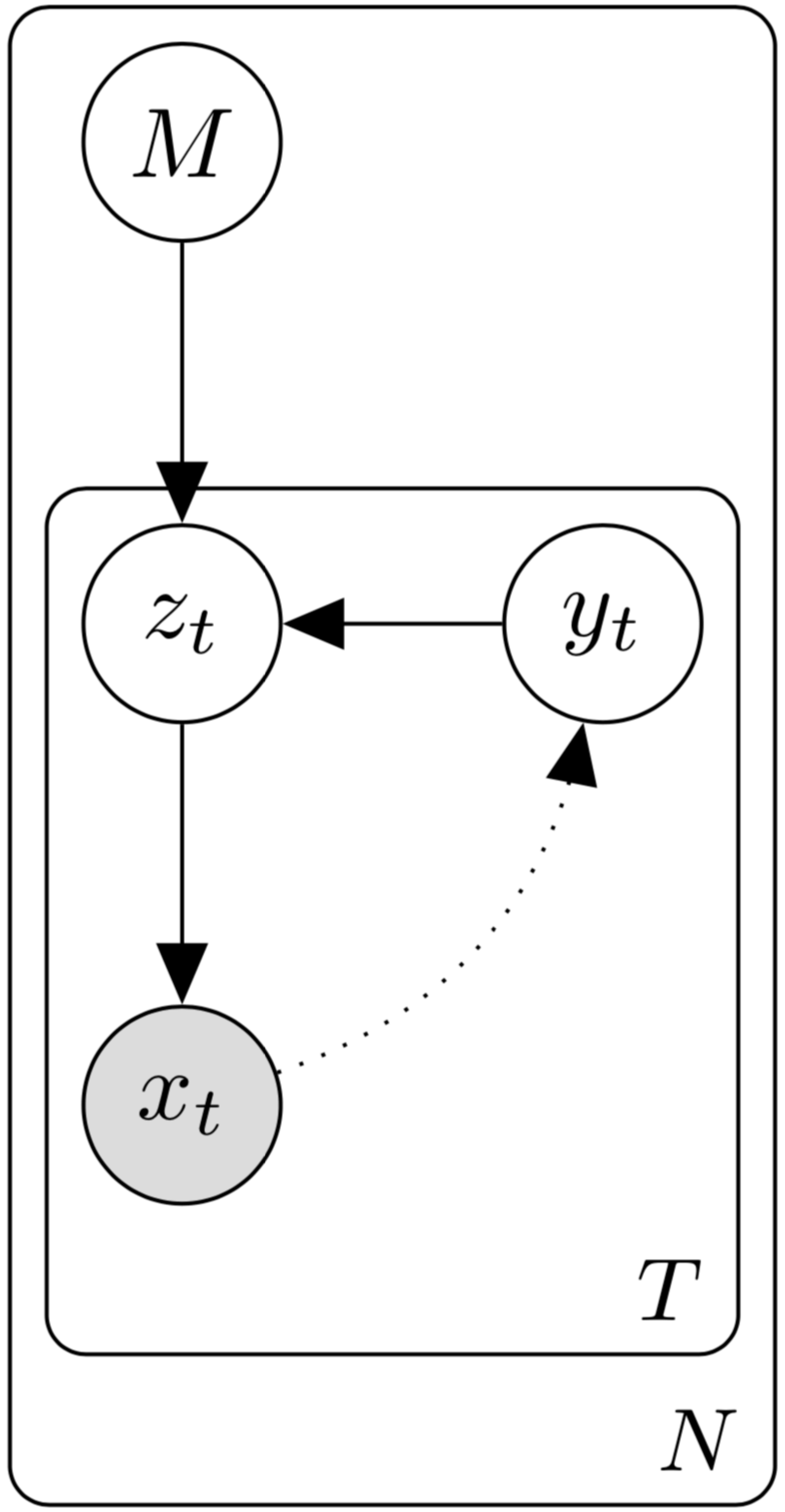




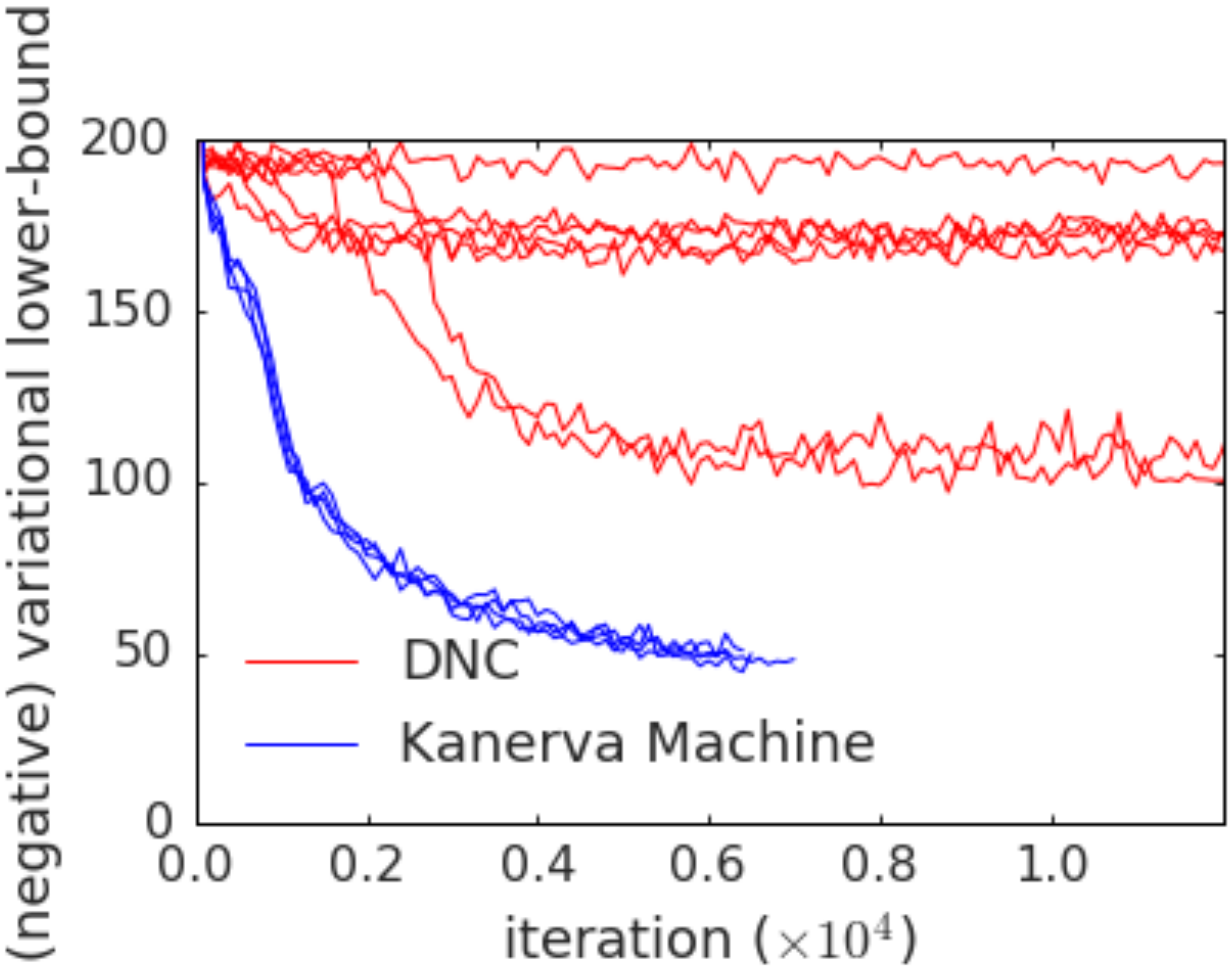
# Iterative Reading

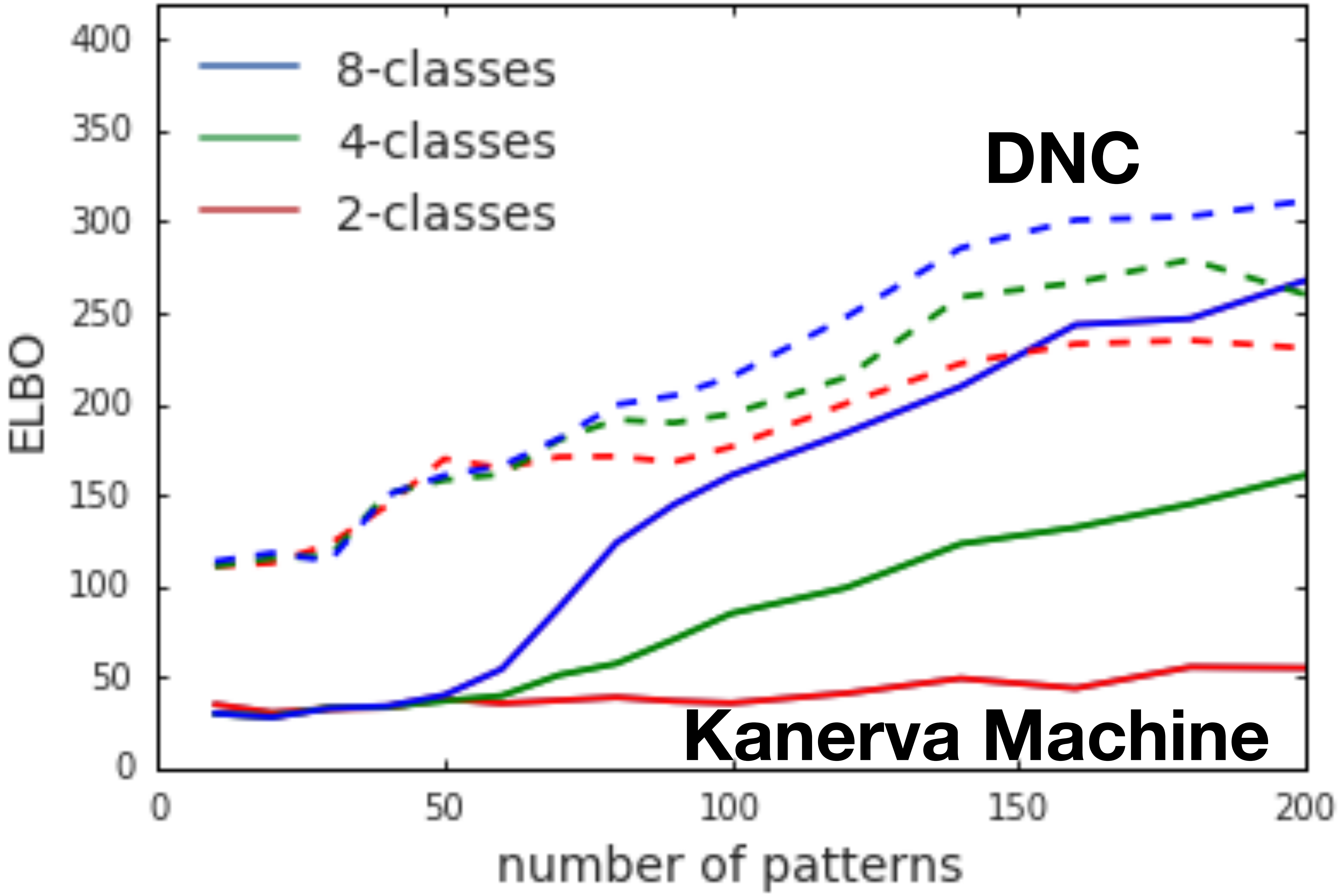
$$q_\phi(Y, Z \,|\, X, M) = \prod_{t=1}^{T} q_\phi(y_t, z_t \,|\, x_t, M) = \prod_{t=1}^{T} q_\phi(z_t \,|\, x_t, y_t, M)\, q_\phi(y_t \,|\, x_t)$$



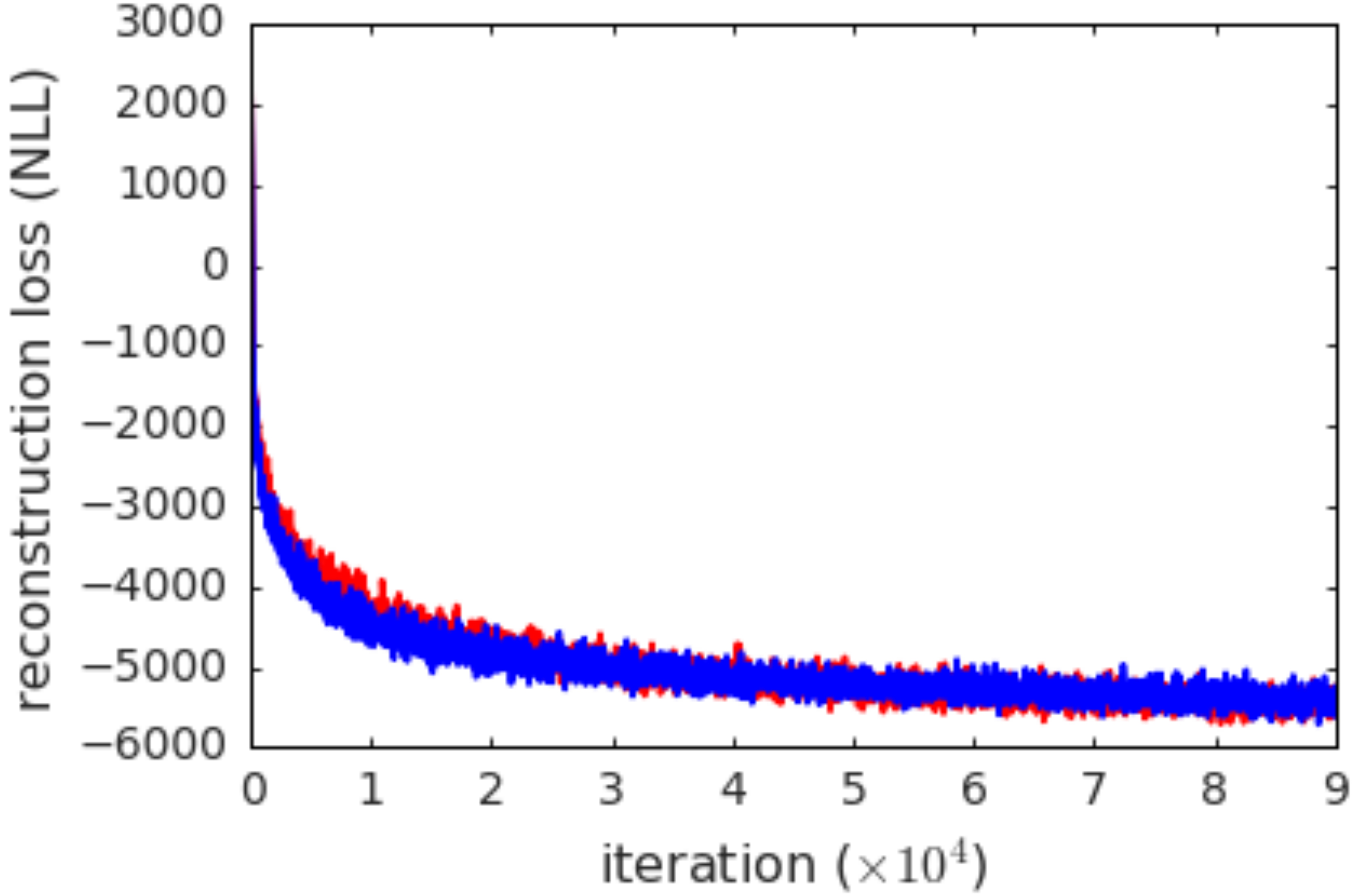


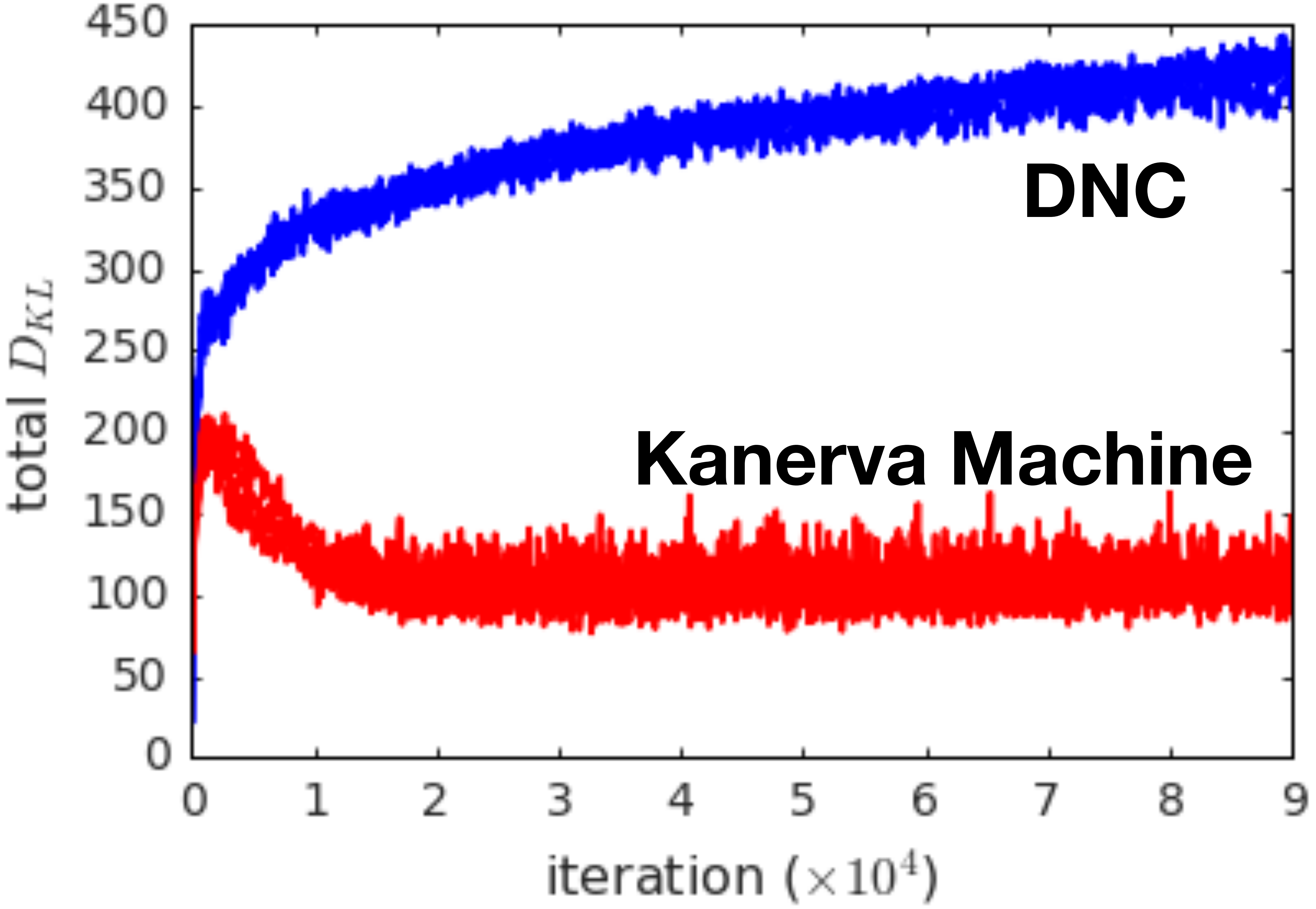
# Kanerva Machine vs DNC





# Kanerva Machine vs DNC





# Learning Attractor Dynamics for Generative Memory

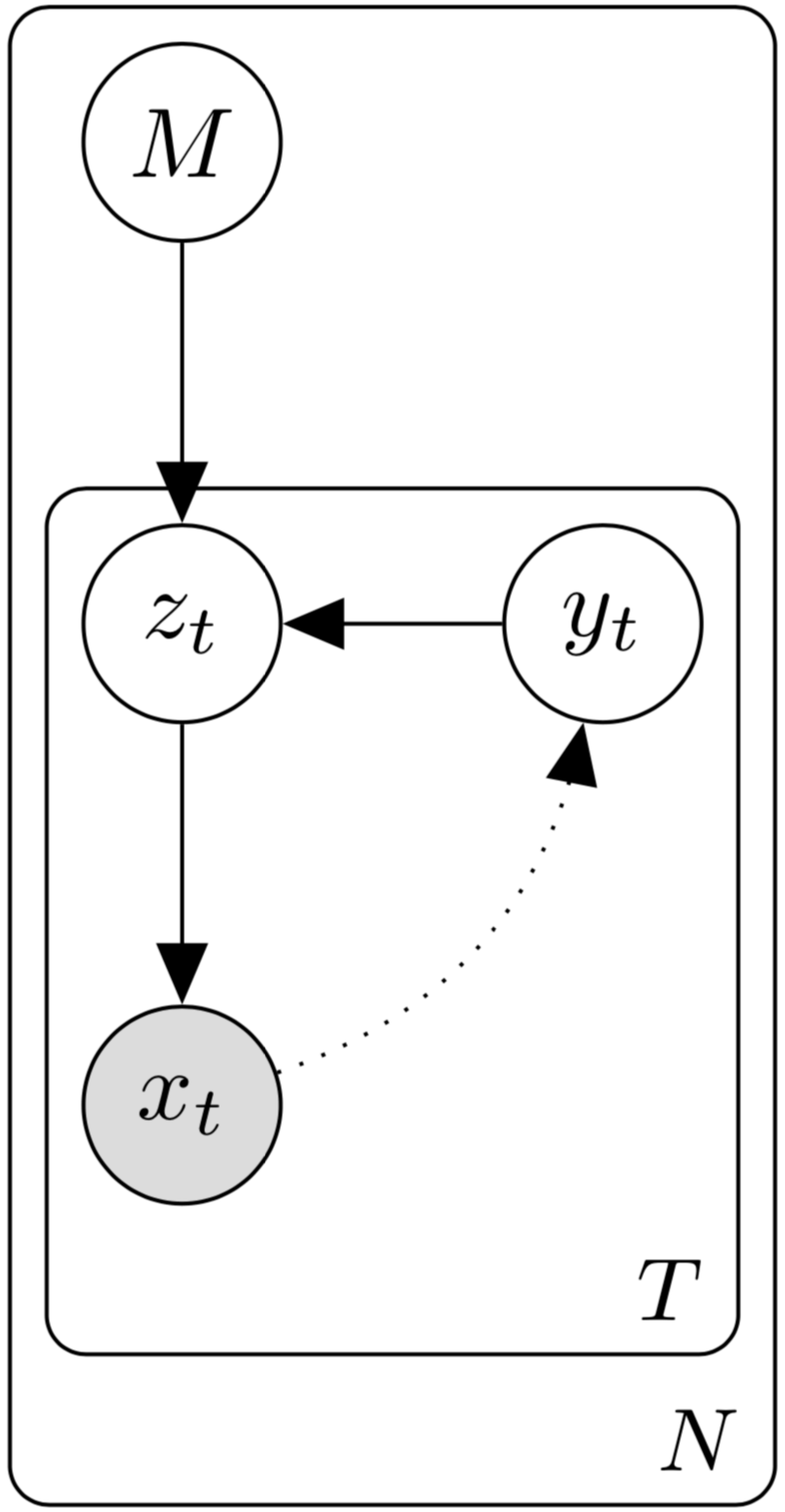
Yan Wu, Greg Wayne, Karol Gregor, Timothy Lillicrap

# Attractor dynamics

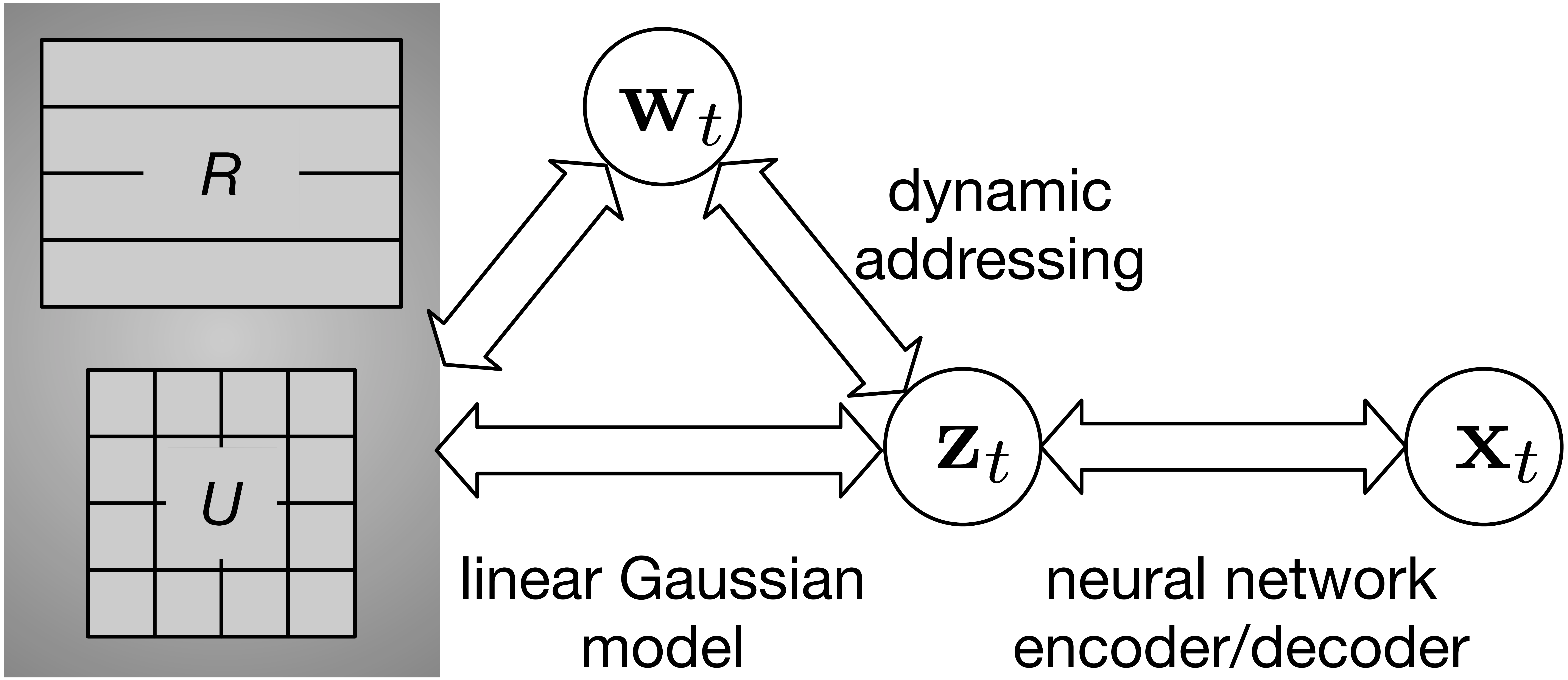
- Iterative reading improves samples during evaluation, but is not used during training

- Idea: use iterative reading during training

- Propagating through «repeat until converged» is hard—vanishing gradients



# Dynamic Kanerva Machine



# Training

$$\ln p(x_{\leqslant T}) = \mathscr{L}_T + \sum_{t=1}^{T} \mathbb{E}_{q(M)} KL(q(w_t) \| p(w_t \mid x_t, M)) + KL(q(M) \| p(M \mid x_{\leq T}))$$

$$\mathscr{L}_T = \sum_{t=1}^{T} \left( \mathbb{E}_{q(w_t)\, q(M)} \log p(x_t \mid w_t, M) - KL(q(w_t) \| p(w_t)) \right) - KL(q(M) \| p(M))$$

# Training

$$\ln p(x_{\leqslant T}) = \mathscr{L}_T + \sum_{t=1}^{T} \mathbb{E}_{q(M)} KL(q(w_t) \| p(w_t \,|\, x_t, M)) + KL(q(M) \| p(M \,|\, x_{\leq T}))$$

Step 1

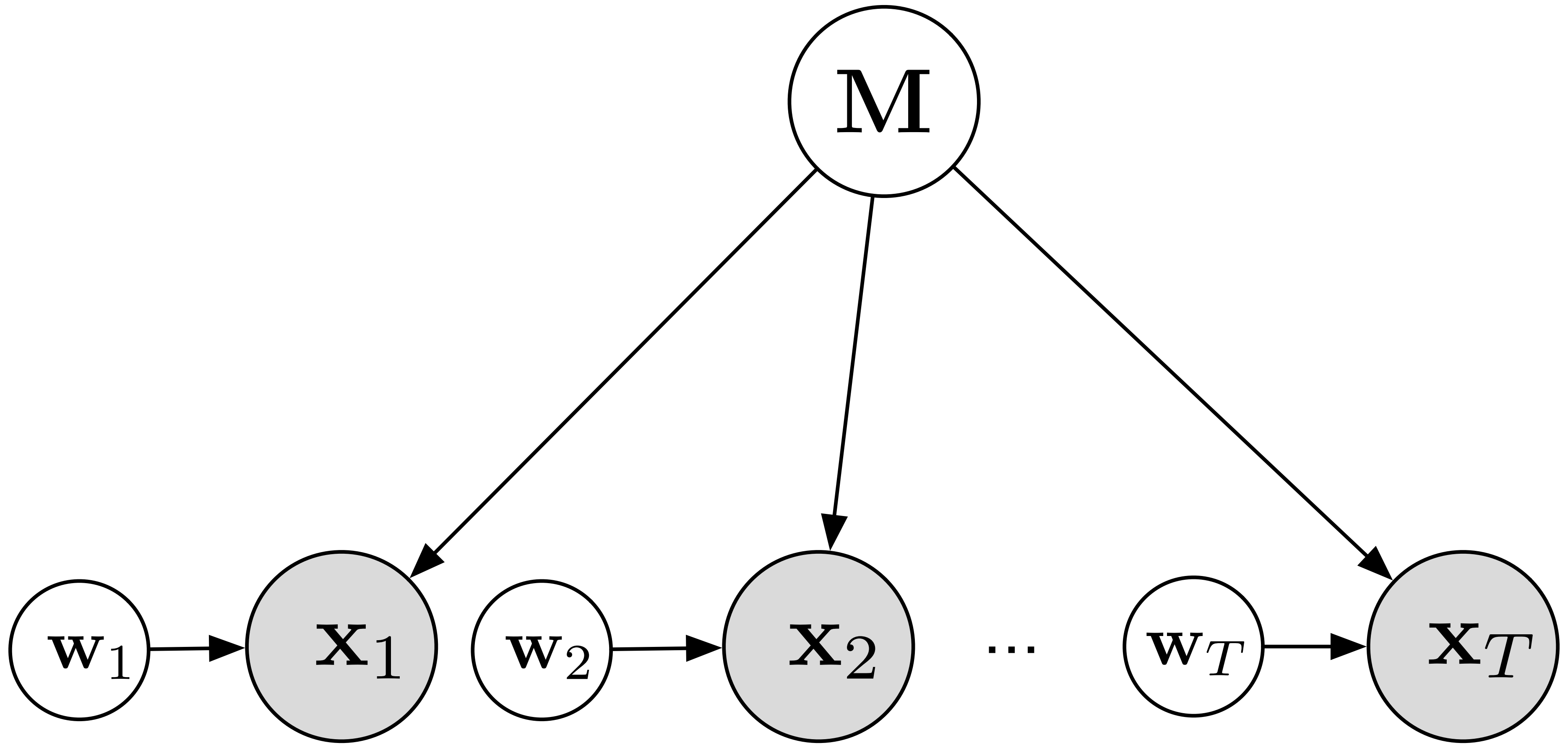
Step 2

$$\mathscr{L}_T = \sum_{t=1}^{T} \left( \mathbb{E}_{q(w_t)\, q(M)} \log p(x_t \,|\, w_t, M) - KL(q(w_t) \| p(w_t)) \right) - KL(q(M) \| p(M))$$

Step 3

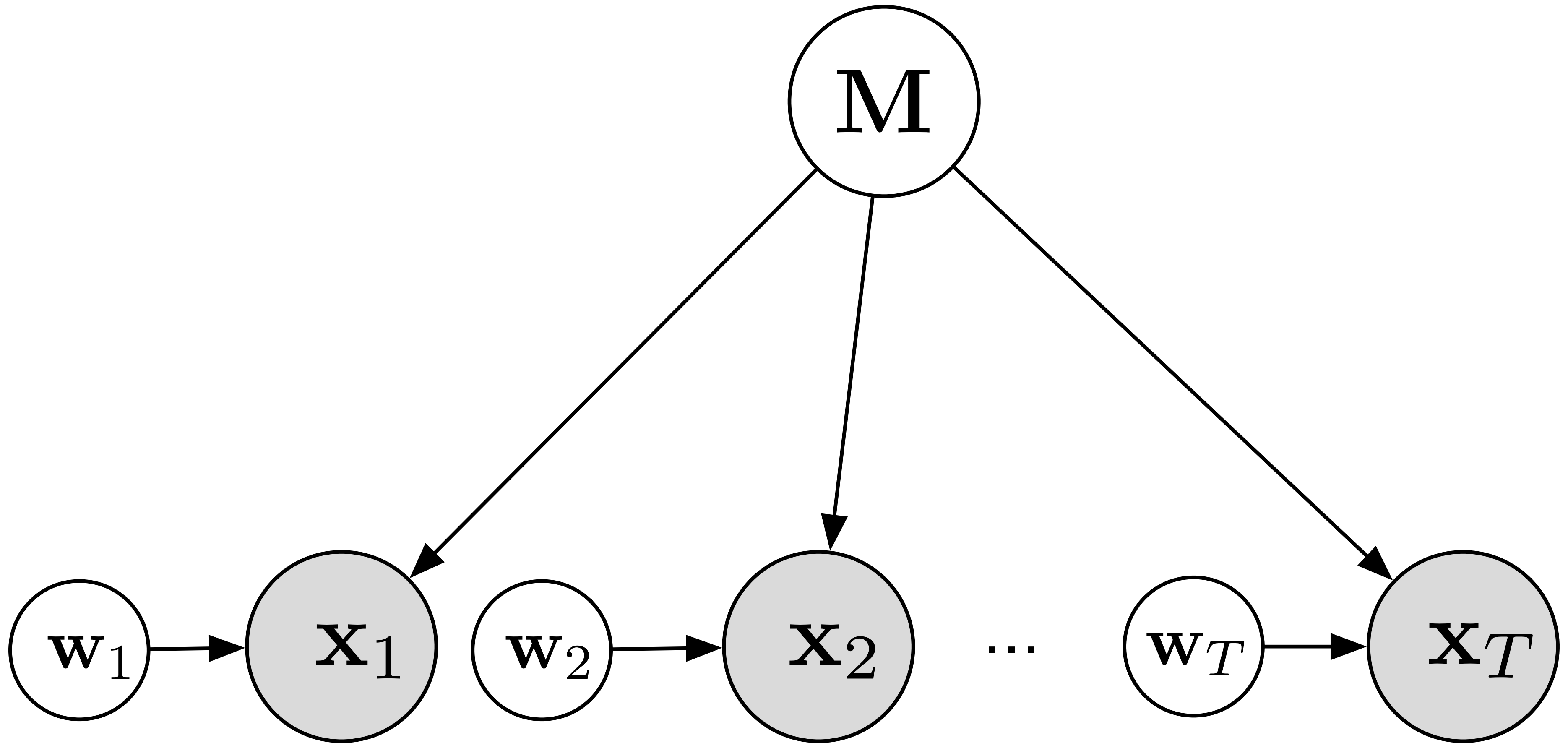
# Step 1. Dynamic addressing

$$\min_{\mu_{w_t}} KL\left(q(w_t) \| p(w_t \mid x_t, M)\right) \quad q(w_t) \sim \mathcal{N}(\mu_{w_t}, \sigma_w^2 I)$$



# Step 1. Dynamic addressing

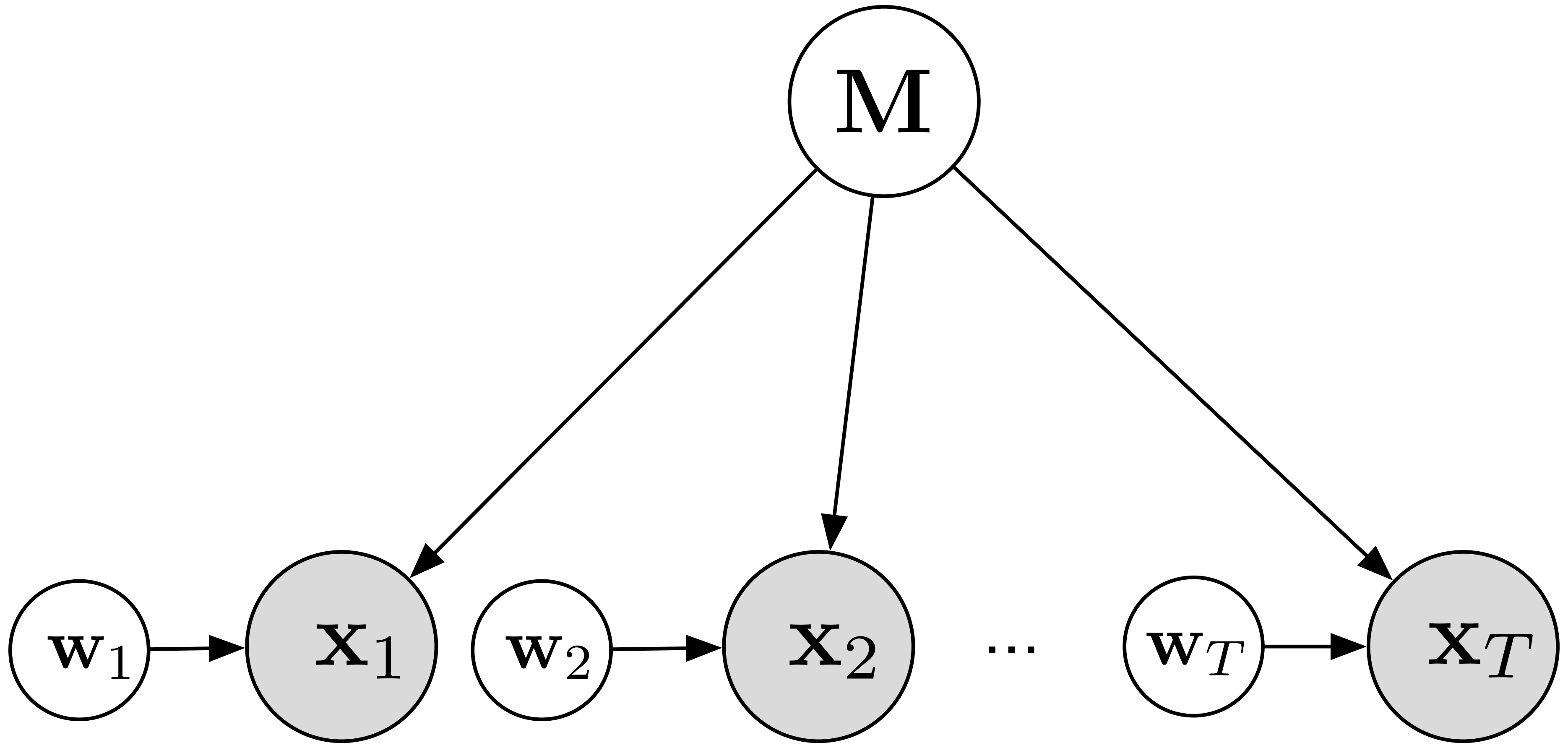
$$\min_{\mu_{w_t}} KL\left(q(w_t) \| p(w_t \,|\, x_t, M)\right) \qquad q(w_t) \sim \mathcal{N}(\mu_{w_t}, \sigma_w^2 I)$$



$$KL(q(w) \| p(w \,|\, x, M)) \approx -\frac{||e(x) - M^T \cdot \mu_w||^2}{2\sigma_\xi^2} - \frac{1}{2}\|\mu_w||^2 + \ldots$$

# Step 1. Dynamic addressing

$$\min_{\mu_{w_t}} KL\left(q(w_t)\|p(w_t\,|\,x_t, M)\right) \quad q(w_t) \sim \mathcal{N}(\mu_{w_t}, \sigma_w^2 I)$$



$$KL(q(w)\|p(w\,|\,x, M)) \approx -\frac{||\,e(x) - M^T \cdot \mu_w\,||^2}{2\sigma_\xi^2} - \frac{1}{2}\|\mu_w\,||^2 + \dots$$

$$\mu_w \leftarrow (MM^T + \sigma_\xi^2 \cdot I)^{-1} M^T e(x)$$

# Training

$$\ln p(x_{\leqslant T}) = \mathscr{L}_T + \sum_{t=1}^{T} \mathbb{E}_{q(M)} KL(q(w_t) \| p(w_t \,|\, x_t, M)) + KL(q(M) \| p(M \,|\, x_{\leq T}))$$

Step 1

Step 2

$$\mathscr{L}_T = \sum_{t=1}^{T} \left( \mathbb{E}_{q(w_t)\, q(M)} \log p(x_t \,|\, w_t, M) - KL(q(w_t) \| p(w_t)) \right) - KL(q(M) \| p(M))$$

Step 3

# Step 2. Bayesian Memory Update

$$\min_{q(M)} KL\left(q(M) \| p(M \mid x_{\leq T})\right)$$

# Step 2. Bayesian Memory Update

$$\xcancel{\min_{q(M)} KL\left(q(M) \| p(M \mid x_{\leq T})\right)}$$

$$\min_{q(M)} KL\left(q(M) \| p(M \mid x_{\leq T}, w_{\leq T})\right) \Leftrightarrow q(M) = p(M \mid x_{\leq T}, w_{\leq T})$$

# Step 2. Bayesian Memory Update

$$\xcancel{\min_{q(M)} KL\left(q(M) \| p(M \mid x_{\leq T})\right)}$$

$$\min_{q(M)} KL\left(q(M) \| p(M \mid x_{\leq T}, w_{\leq T})\right) \Leftrightarrow q(M) = p(M \mid x_{\leq T}, w_{\leq T})$$

- Solve by iteratively writing data to the memory:
    - $\mu_{w_t} = \arg\min_{\mu_{w_t}} KL\left(q(w_t) \| p(w_t \mid x_t, M_{t-1})\right)$
    - $q(M_t) \approx p(M_t \mid x_t, \mu_{w_t}, M_{t-1})$

# Training

$$\ln p(x_{\leqslant T}) = \mathscr{L}_T + \sum_{t=1}^{T} \mathbb{E}_{q(M)} KL(q(w_t) \| p(w_t \mid x_t, M)) + KL(q(M) \| p(M \mid x_{\leq T}))$$

Step 1

Step 2

$$\mathscr{L}_T = \sum_{t=1}^{T} \left( \mathbb{E}_{q(w_t)\, q(M)} \log p(x_t \mid w_t, M) - KL(q(w_t) \| p(w_t)) \right) - KL(q(M) \| p(M))$$

Step 3

# Training

$$\ln p(x_{\leqslant T}) = \mathscr{L}_T + \sum_{t=1}^{T} \mathbb{E}_{q(M)} KL(q(w_t) \| p(w_t \,|\, x_t, M)) + KL(q(M) \| p(M \,|\, x_{\leq T}))$$

**Step 1** **Step 2**

$$\mathscr{L}_T = \sum_{t=1}^{T} \left( \mathbb{E}_{q(w_t)\,q(M)} \log p(x_t \,|\, w_t, M) - KL(q(w_t) \| p(w_t)) \right) - KL(q(M) \| p(M))$$

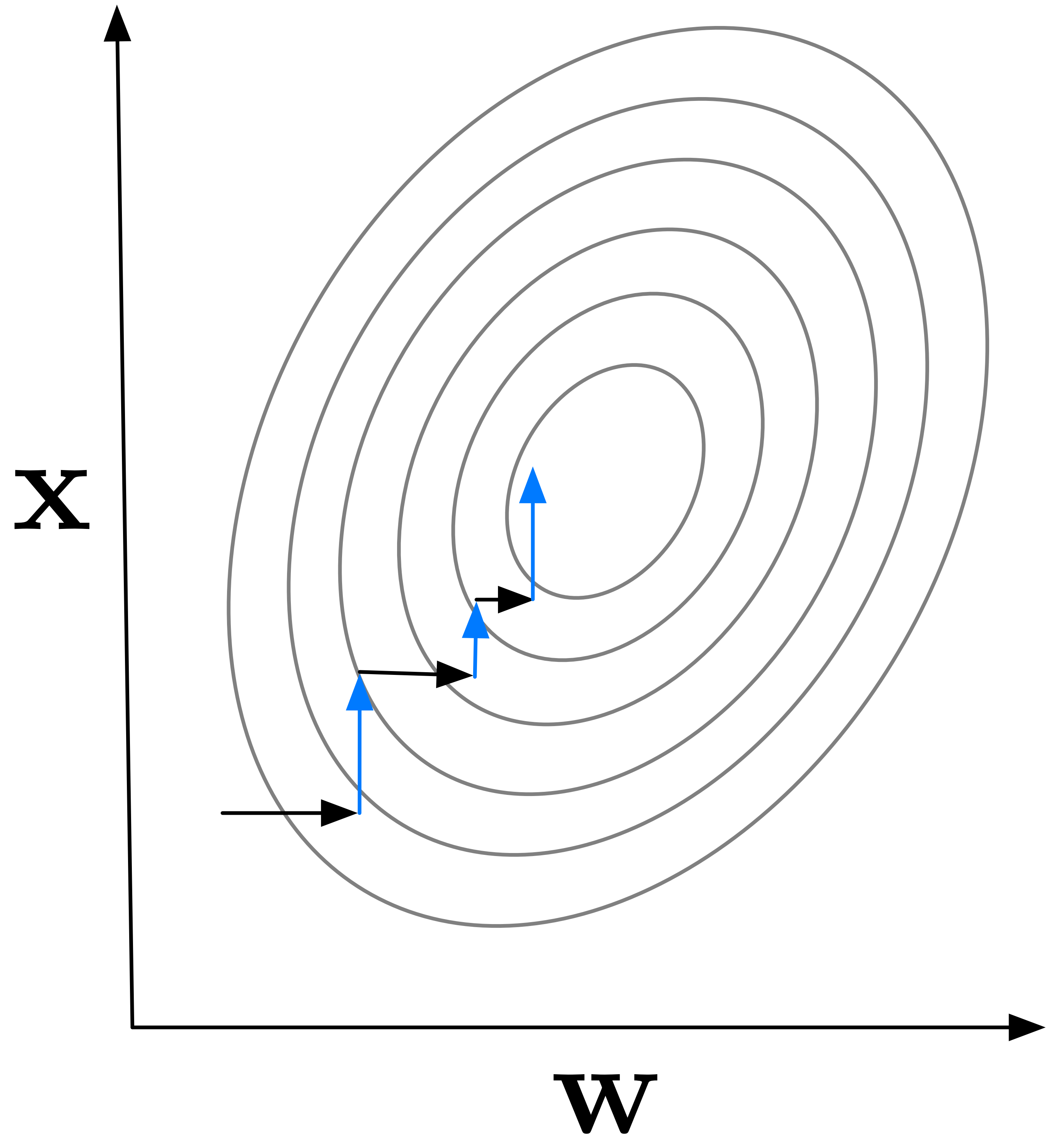
**Step 3** $\min \left[ \mathscr{L}_T + \mathscr{L}_{AE} \right]$ $\mathscr{L}_{AE} = \mathbb{E}_{p(X)} \log d\left(e(x)\right)$

# Attractor dynamics

$$\mathscr{L}_T = \sum_{t=1}^{T} \Big( \mathbb{E}_{q(w_t)\,q(M)} \log p(x_t \,|\, w_t, M) - KL(q(w_t) \| p(w_t)) \Big) - KL(q(M) \| p(M))$$

$$\mathscr{E}(x, q(w)) = -\,\mathbb{E}_{q(M)q(w)} \log p(x \mid w, M) + KL\left(q(w) \| p(w)\right)$$

$\mathcal{E}(\mathbf{x}, \mathbf{w})$



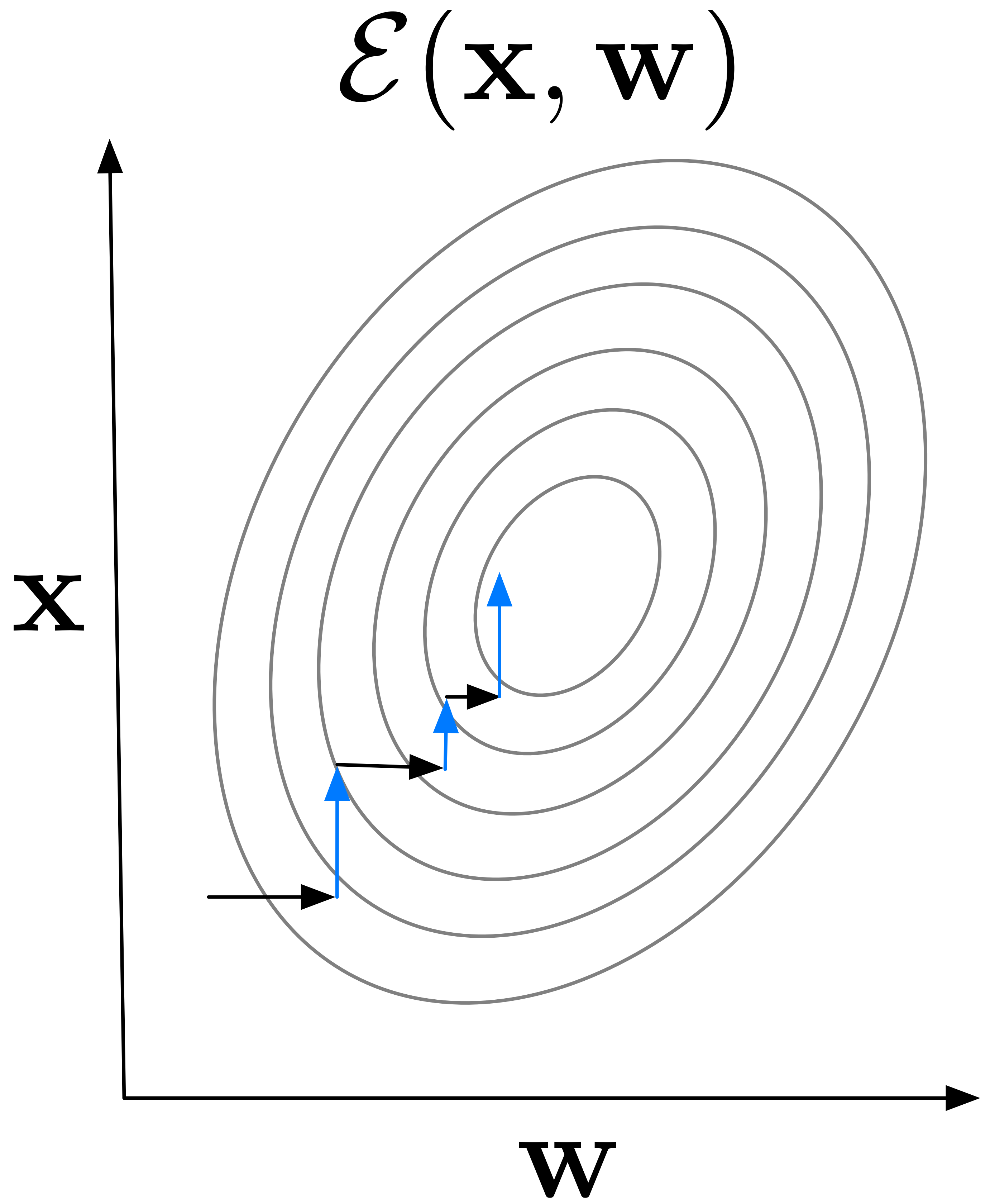
# Attractor dynamics

$$\mathscr{L}_T = \sum_{t=1}^{T} \left( \mathbb{E}_{q(w_t)\,q(M)} \log p(x_t \,|\, w_t, M) - KL(q(w_t) \| p(w_t)) \right) - KL(q(M) \| p(M))$$

$$\mathscr{E}(x, q(w)) = -\mathbb{E}_{q(M)q(w)} \log p(x \mid w, M) + KL\left(q(w) \| p(w)\right)$$

- Coordinate descent:

$$x_{t+1} = \arg\max_{x_{t+1}} p(x_{t+1} \mid x_t, M)$$



# Attractor dynamics

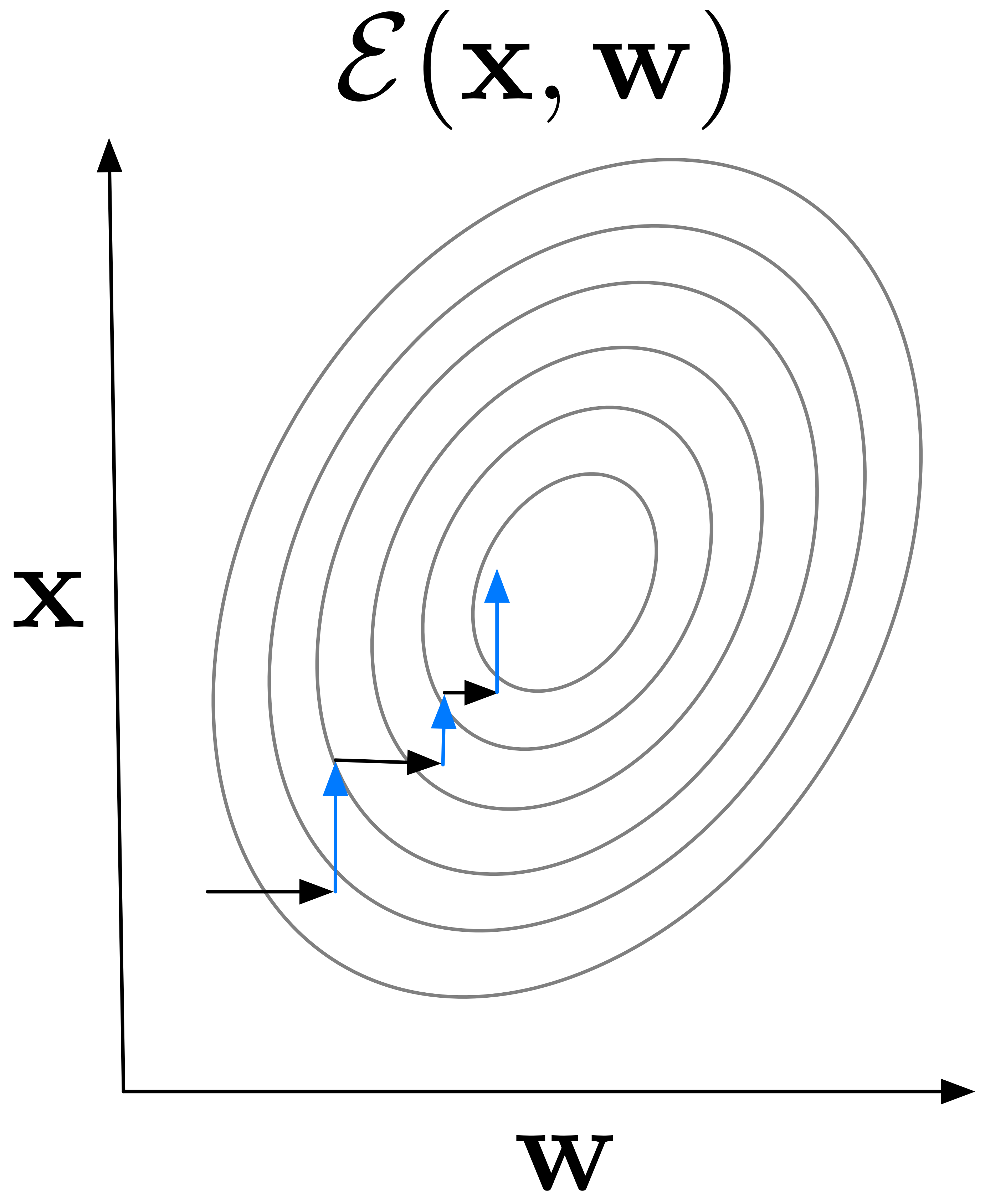
$$\mathscr{L}_T = \sum_{t=1}^{T} \left( \mathbb{E}_{q(w_t)\,q(M)} \log p(x_t \,|\, w_t, M) - KL(q(w_t) \| p(w_t)) \right) - KL(q(M) \| p(M))$$

$$\mathscr{E}(x, q(w)) = -\,\mathbb{E}_{q(M)q(w)} \log p(x \mid w, M) + KL\left(q(w) \| p(w)\right)$$

- Coordinate descent:

$$x_{t+1} = \arg\max_{x_{t+1}} p(x_{t+1} \mid x_t, M)$$

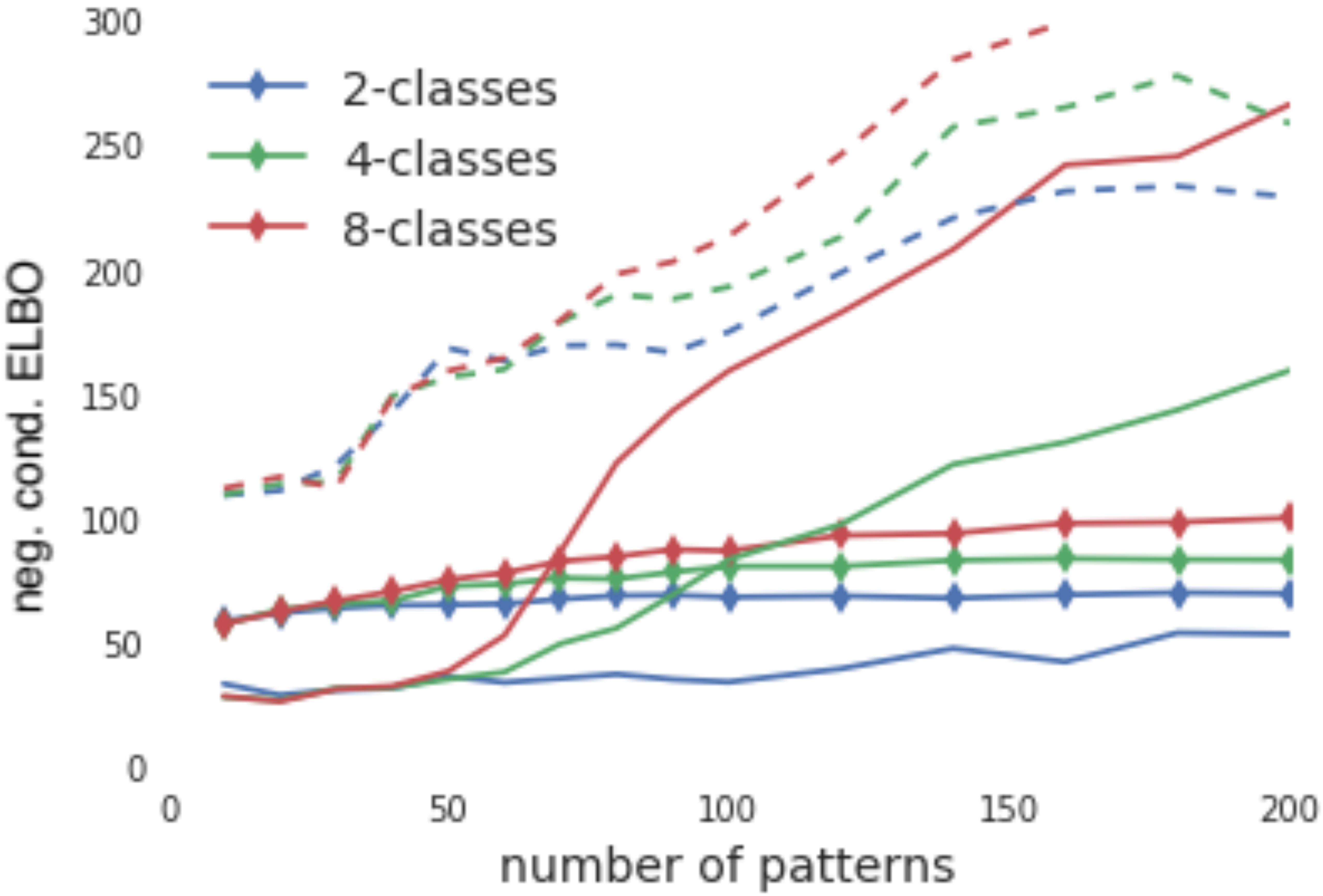
$$\begin{cases} \mu_{t,w} \leftarrow (MM^T + \sigma_\xi^2 \cdot I)^{-1} M^T e(x_t) \\ x_{t+1} = \arg\max_{x_{t+1}} \left[\log p(x_{t+1} \mid \mu_{t,w}, M)\right] \end{cases}$$



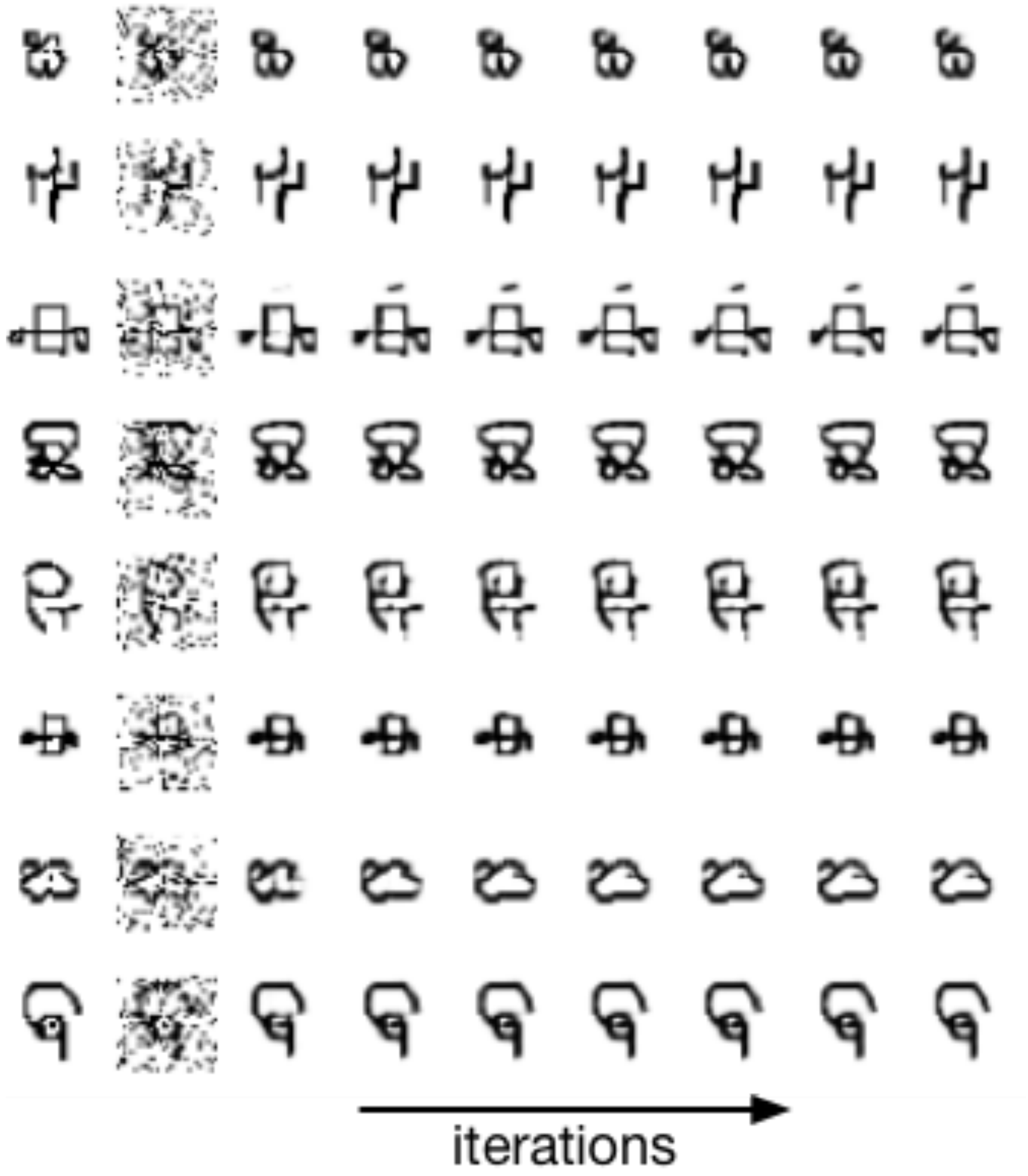
# Experiments

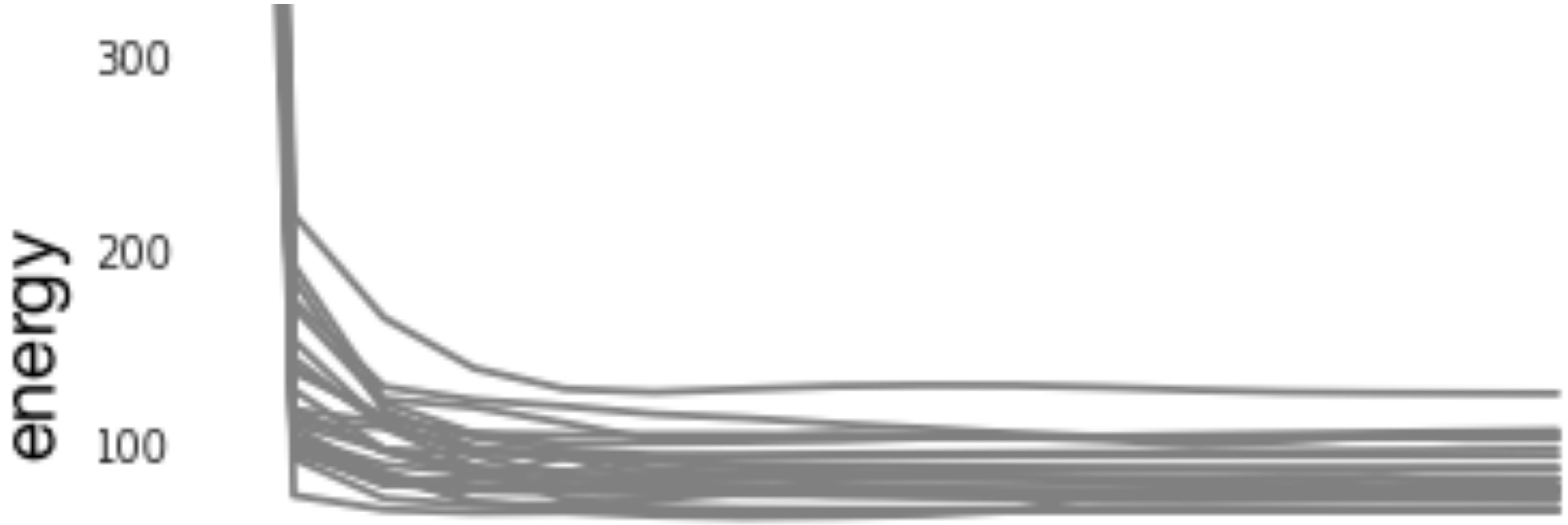
Learning Attractor Dynamics for Generative Memory

# Capacity

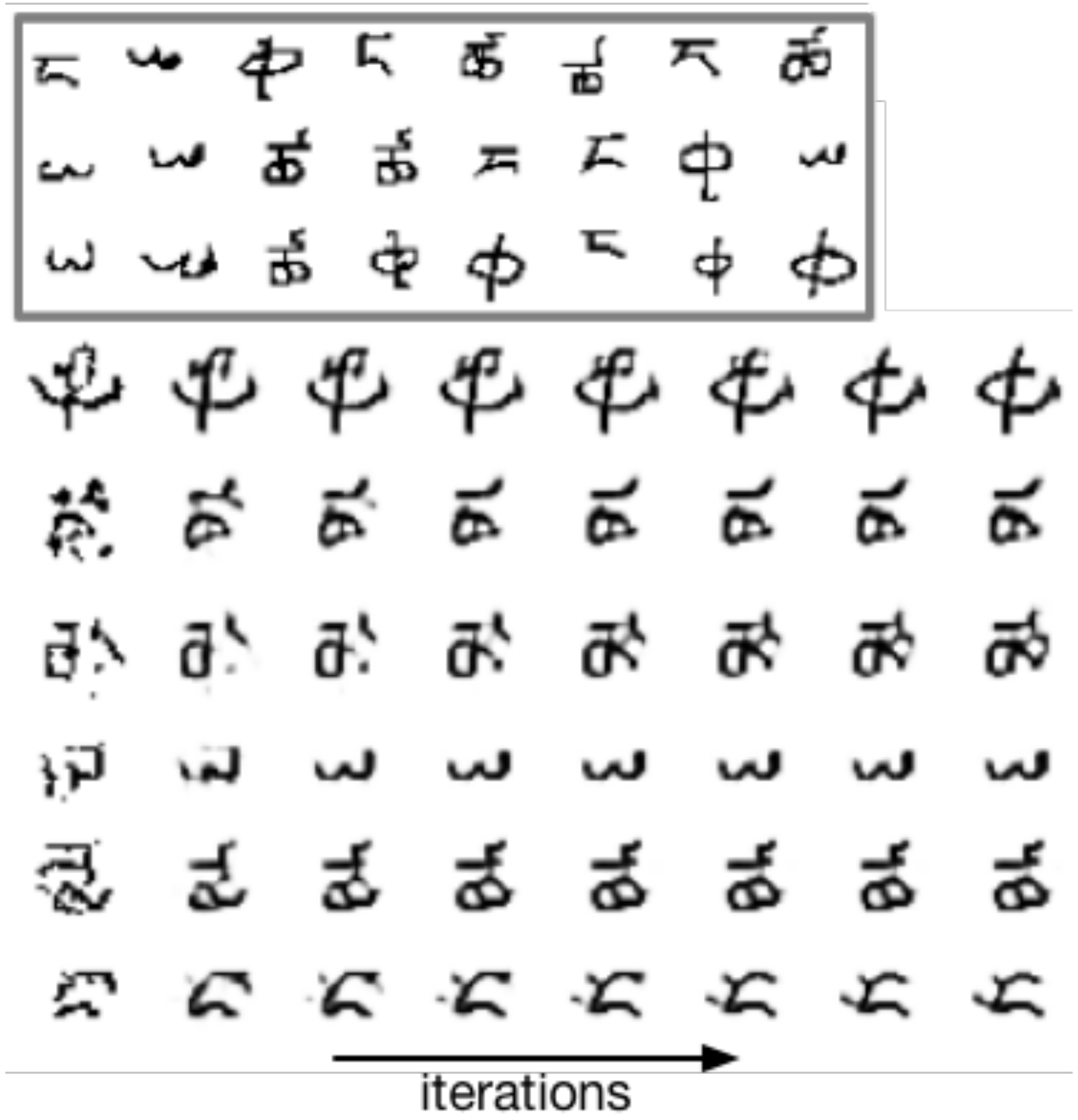

2-classes
4-classes
8-classes
neg. cond. ELBO
300
250
200
150
100
50
0
0
50
100
150
200
number of patterns

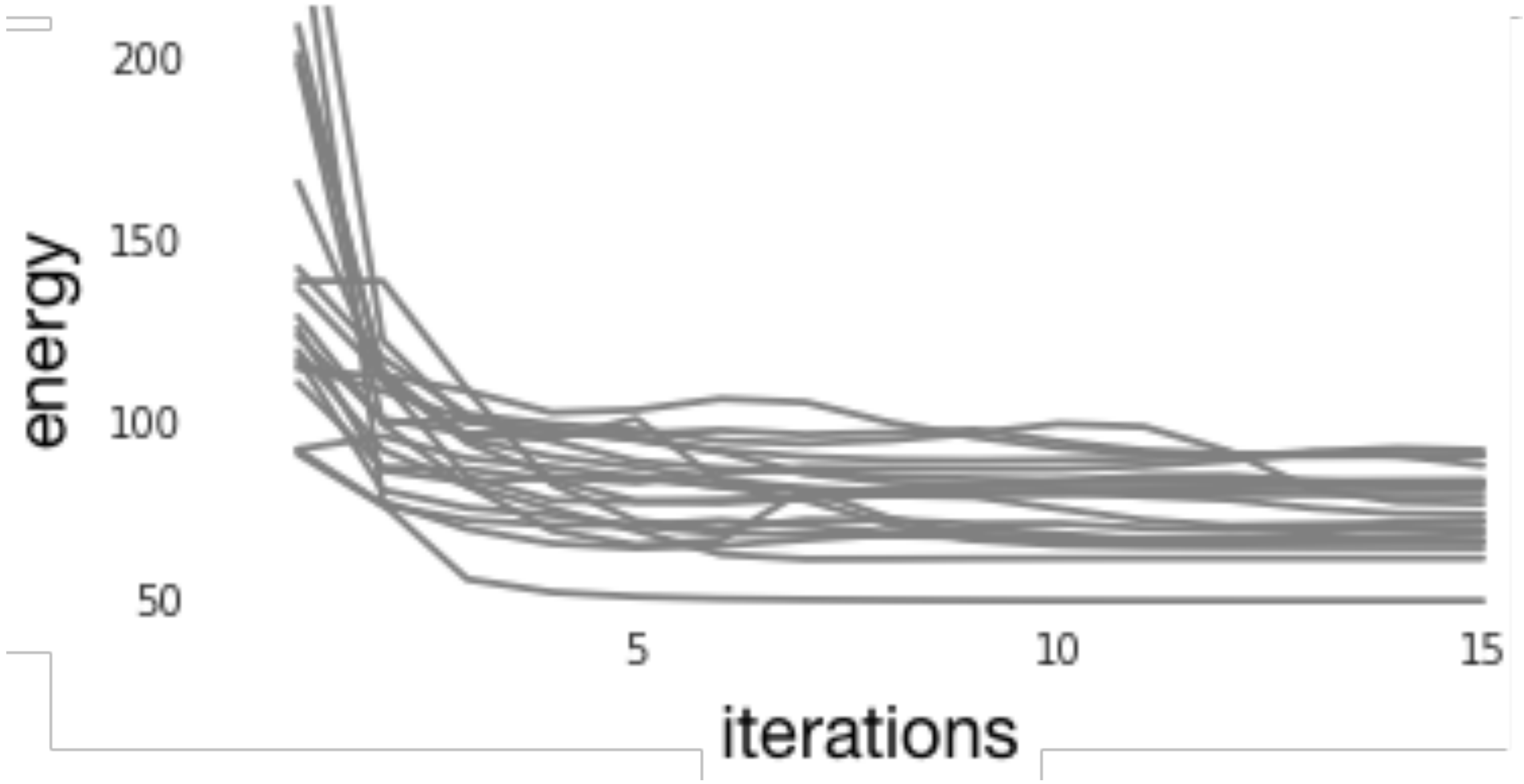
# Denoising





# Sampling

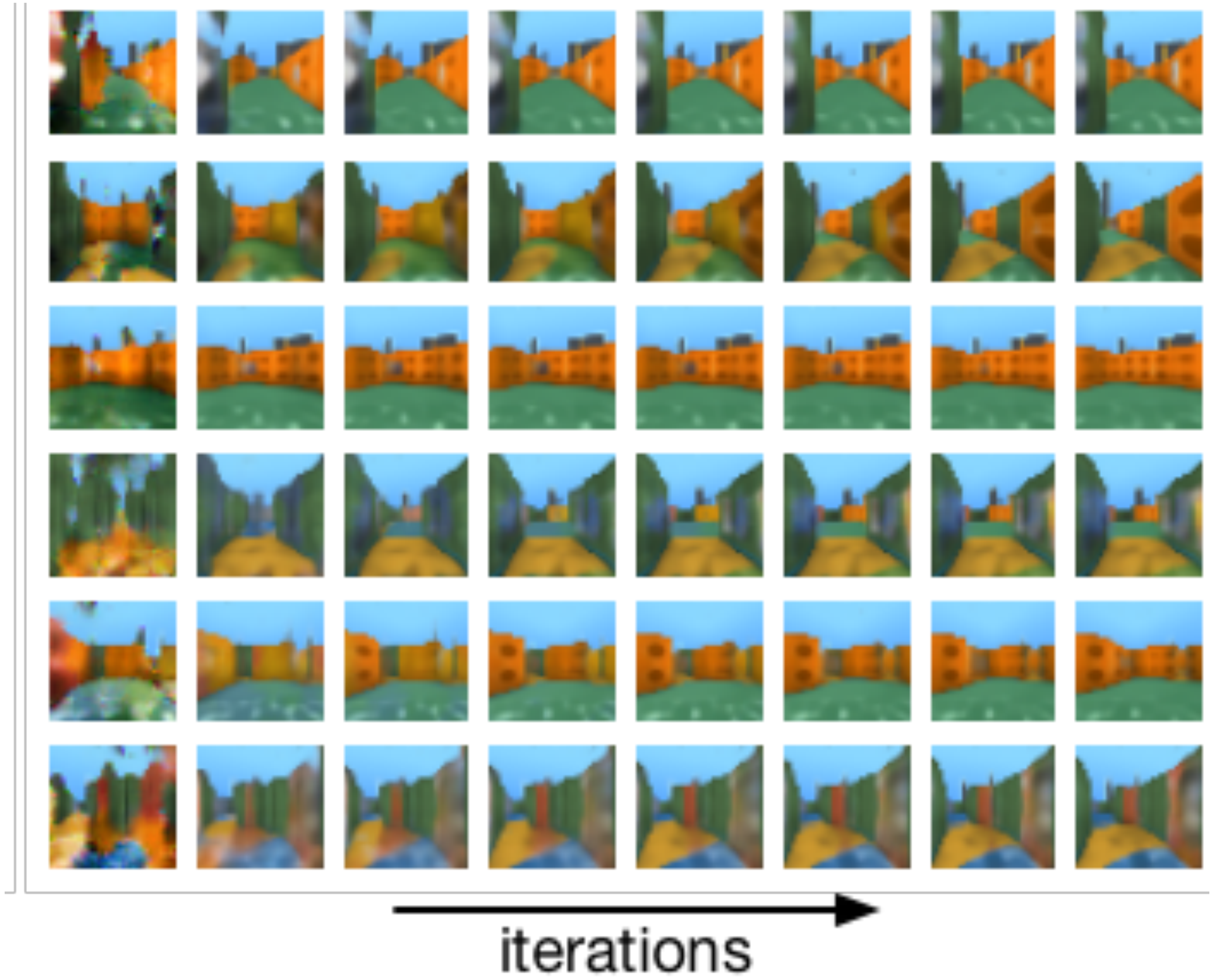


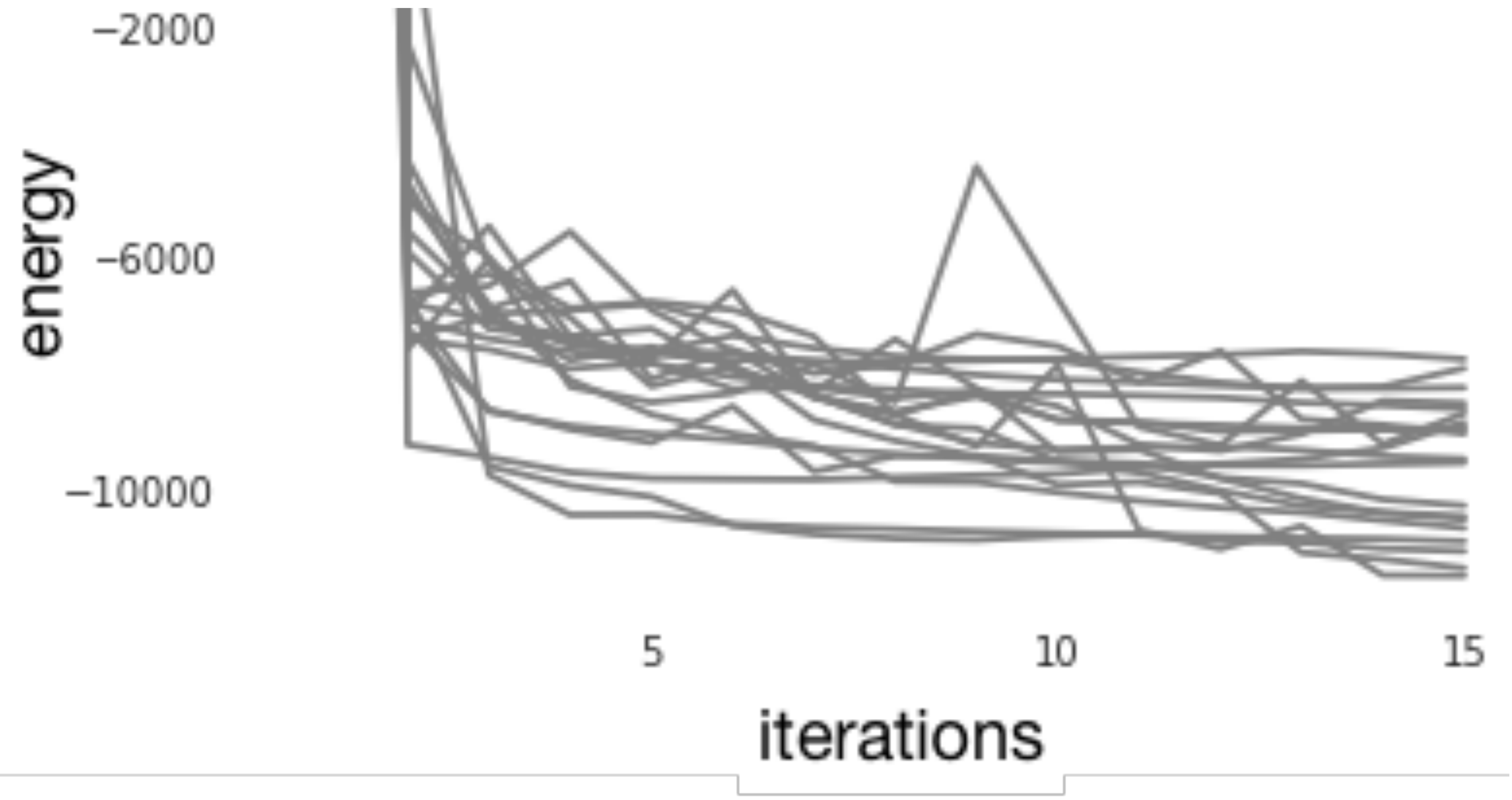


# Denoising

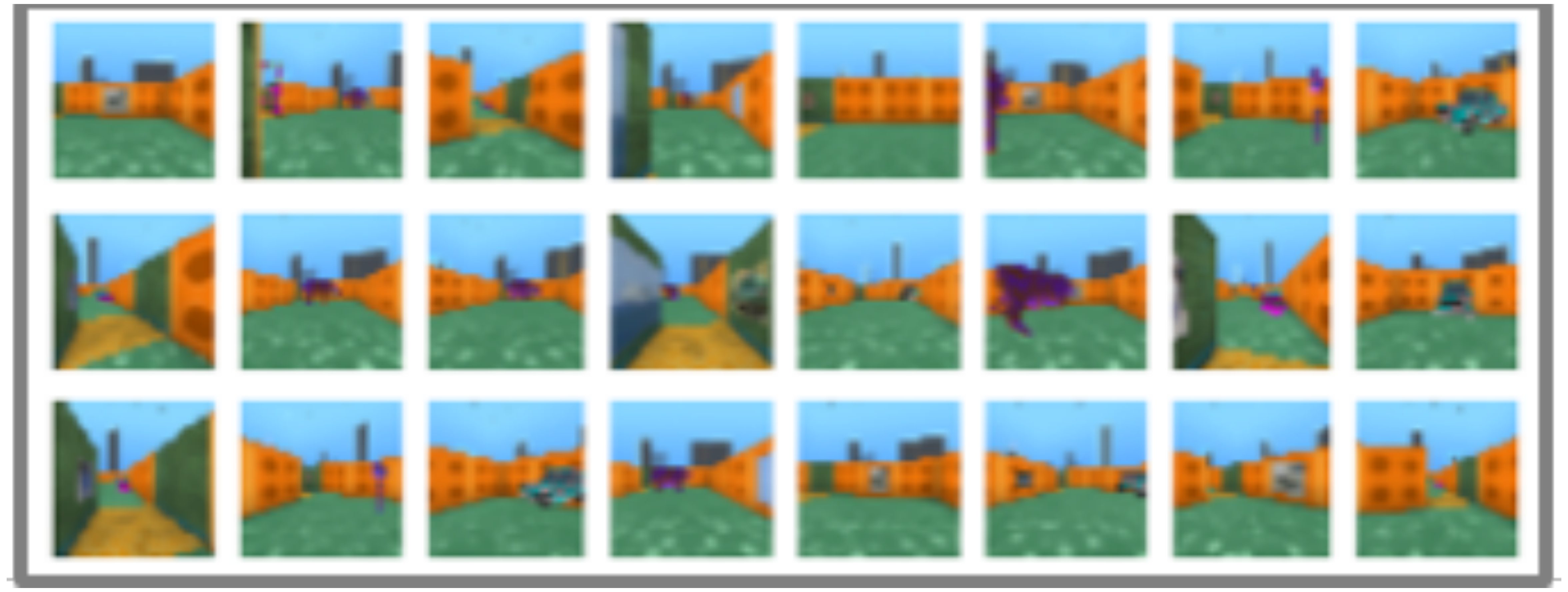
iterations
energy
−6000
-10000
5
10
15

# Sampling





**Input images**



# Summary

- **The Kanerva Machine: A Generative Distributed Memory**
    - Distributed read/write operations
    - Writing as inference
- **Learning Attractor Dynamics for Generative Memory**
    - Finds «optimal» reading weights
    - Iterative reading to restore written objects