Attention via log $\sum$ exp energy

Alexander Tschantz

January 24, 2025

1 General Framework

Setup. We consider a single set of nodes $\boldsymbol{v} = \{\boldsymbol{v}_a : a \in \{1, 2, \ldots, A\}\}$, where each node $\boldsymbol{v}_a \in \mathbb{R}^d$. The relationships between these nodes are defined by a set of M energy functions $\{E_m : m \in \{1, 2, \ldots, M\}\}$. Each energy function E_m defines a subset of nodes acting as *children* $C_m \subseteq \{1, 2, \ldots, A\}$ and a subset acting as *parents* $P_m \subseteq \{1, 2, \ldots, A\}$, which may overlap.

Energy. Each energy function E_m defines a similarity function:

$$\operatorname{sim}\big(\boldsymbol{v}_c, \boldsymbol{v}_p\big) \quad : \quad \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}, \tag{1}$$

which produces a scalar similarity between a child $\boldsymbol{v}_c$ and a parent $\boldsymbol{v}_p$. Using $\{\boldsymbol{v}_c\} = \{\boldsymbol{v}_c : c \in C_m\}$ and $\{\boldsymbol{v}_p\} = \{\boldsymbol{v}_p : p \in P_m\}$, the energy for E_m is defined as:

$$E_m\big(\{\boldsymbol{v}_c\}, \{\boldsymbol{v}_p\}\big) \;=\; -\sum_{c \in C_m} \ln\Big(\sum_{p \in P_m} \exp\big(\operatorname{sim}(\boldsymbol{v}_c, \boldsymbol{v}_p)\big)\Big). \tag{2}$$

The global energy sums over all energy functions:

$$E\big(\{\boldsymbol{v}\}\big) \;=\; \sum_{m=1}^{M} E_m\big(\{\boldsymbol{v}_c\}, \{\boldsymbol{v}_p\}\big). \tag{3}$$

Gradient Updates. For a single node $\boldsymbol{v}_a$, the gradient of the global energy E w.r.t. $\boldsymbol{v}_a$ decomposes into two terms. Let $\mathcal{M}_c(a) = \{m : a \in C_m\}$ denote the energy functions where $\boldsymbol{v}_a$ acts as a *child*, and $\mathcal{M}_p(a) = \{m : a \in P_m\}$ the energy functions where $\boldsymbol{v}_a$ acts as a *parent*. Then:

$$\begin{aligned} -\frac{\partial E}{\partial \boldsymbol{v}_a} \;=\; & \underbrace{\sum_{m \in \mathcal{M}_c(a)} \sum_{p \in P_m} \operatorname{softmax}_p\Big(\operatorname{sim}(\boldsymbol{v}_a, \boldsymbol{v}_p)\Big)\, \frac{\partial}{\partial \boldsymbol{v}_a} \operatorname{sim}\big(\boldsymbol{v}_a, \boldsymbol{v}_p\big)}_{\boldsymbol{v}_a \text{ acting as a child}} \\ & + \underbrace{\sum_{m \in \mathcal{M}_p(a)} \sum_{c \in C_m} \operatorname{softmax}_a\Big(\operatorname{sim}(\boldsymbol{v}_c, \boldsymbol{v}_a)\Big)\, \frac{\partial}{\partial \boldsymbol{v}_a} \operatorname{sim}\big(\boldsymbol{v}_c, \boldsymbol{v}_a\big)}_{\boldsymbol{v}_a \text{ acting as a parent}}. \end{aligned} \tag{4}$$

The first term captures contributions from $\boldsymbol{v}_a$ being explained by its parents, while the second term captures contributions from $\boldsymbol{v}_a$ explaining its children.

2 Gaussian Mixture Models

Setup. We have N data points (children) $\boldsymbol{x}_i \in \mathbb{R}^d$, $i \in C = \{1, \ldots, N\}$, and K mixture components (parents), each with mean $\boldsymbol{\mu}_k \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma}_k$, $k \in P = \{1, \ldots, K\}$. Let π_k be the mixing proportion.

Similarity function. We define

$$\operatorname{sim}\big(\boldsymbol{x}_i, \boldsymbol{\mu}_k\big) \;=\; \ln \pi_k \;-\; \tfrac{1}{2}\big(\boldsymbol{x}_i - \boldsymbol{\mu}_k\big)^\top \boldsymbol{\Sigma}_k^{-1}\big(\boldsymbol{x}_i - \boldsymbol{\mu}_k\big).$$

Energy.

$$E^{\text{GMM}}\Big(\{\boldsymbol{x}_i\},\{\boldsymbol{\mu}_k\}\Big) \;=\; -\sum_{i=1}^{N}\ln\Big(\sum_{k=1}^{K}\exp\big(\text{sim}(\boldsymbol{x}_i,\boldsymbol{\mu}_k)\big)\Big). \tag{5}$$

Gradients. If we differentiate w.r.t. $\boldsymbol{\mu}_k$, then

$$-\frac{\partial E^{\text{GMM}}}{\partial \boldsymbol{\mu}_k} \;=\; \sum_{i=1}^{N}\text{softmax}_k\big(\boldsymbol{A}_{ik}\big)\;\boldsymbol{\Sigma}_k^{-1}\big(\boldsymbol{x}_i-\boldsymbol{\mu}_k\big).$$

Setting this gradient to zero yields the usual GMM M-step:

$$\boldsymbol{\mu}_k \;=\; \frac{\sum_{i=1}^{N}\text{softmax}_k\big(\boldsymbol{A}_{ik}\big)\;\boldsymbol{x}_i}{\sum_{i=1}^{N}\text{softmax}_k\big(\boldsymbol{A}_{ik}\big)}.$$

3 Cross Attention

Setup. We have a set of child vectors (queries) $\boldsymbol{Q}\in\mathbb{R}^{d\times N_Q}$ and a set of parent vectors (keys) $\boldsymbol{K}\in\mathbb{R}^{d\times N_K}$. Let

$$C \;=\; \{1,\dots,N_Q\},\quad P \;=\; \{1,\dots,N_K\},$$

so $\boldsymbol{v}_c=\boldsymbol{q}_c$ is the c-th query, and $\boldsymbol{v}_p=\boldsymbol{k}_p$ is the p-th key. Suppose we have learnable weight matrices $\boldsymbol{W}^Q,\boldsymbol{W}^K\in\mathbb{R}^{d\times d}$. Then

$$\boldsymbol{q}_c \;=\; \boldsymbol{W}^Q\boldsymbol{x}_c^Q,\quad \boldsymbol{k}_p \;=\; \boldsymbol{W}^K\boldsymbol{x}_p^K,$$

where $\boldsymbol{x}_c^Q$ is the raw c-th query token and $\boldsymbol{x}_p^K$ the raw p-th key token.

Similarity function.

$$\text{sim}\big(\boldsymbol{q}_c,\boldsymbol{k}_p\big) \;=\; \boldsymbol{q}_c^\top\boldsymbol{k}_p.$$

Energy.

$$E^{\text{Cross}}\Big(\{\boldsymbol{q}_c\},\{\boldsymbol{k}_p\}\Big) \;=\; -\sum_{c=1}^{N_Q}\ln\Big(\sum_{p=1}^{N_K}\exp\big(\boldsymbol{q}_c^\top\,\boldsymbol{k}_p\big)\Big). \tag{6}$$

Gradients.

$$-\frac{\partial E^{\text{Cross}}}{\partial \boldsymbol{q}_c} \;=\; \sum_{p=1}^{N_K}\text{softmax}_p\Big(\boldsymbol{q}_c^\top\boldsymbol{k}_p\Big)\,\boldsymbol{k}_p. \tag{7}$$

$$-\frac{\partial E^{\text{Cross}}}{\partial \boldsymbol{k}_p} \;=\; \sum_{c=1}^{N_Q}\text{softmax}_p\Big(\boldsymbol{q}_c^\top\boldsymbol{k}_p\Big)\,\boldsymbol{q}_c. \tag{8}$$

When mapping back to the raw tokens $\boldsymbol{x}_c^Q$ or $\boldsymbol{x}_p^K$, chain-rule multiplies by $\boldsymbol{W}^Q$ or $\boldsymbol{W}^K$, respectively.

4 Hopfield Networks

Setup. We have a set of *children* data vectors $\boldsymbol{x}_i\in\mathbb{R}^d$, $i\in C=\{1,\dots,N\}$, and a set of *parent* memory vectors $\boldsymbol{m}_\mu\in\mathbb{R}^d$, $\mu\in P=\{1,\dots,K\}$.

Similarity function.

$$\text{sim}\big(\boldsymbol{x}_i,\boldsymbol{m}_\mu\big) \;=\; \boldsymbol{x}_i^\top\boldsymbol{m}_\mu.$$

Energy.

$$E^{\text{Hopfield}}\Big(\{\boldsymbol{x}_i\},\{\boldsymbol{m}_\mu\}\Big) \;=\; -\sum_{i=1}^{N}\ln\Big(\sum_{\mu=1}^{K}\exp\big(\boldsymbol{x}_i^\top\boldsymbol{m}_\mu\big)\Big). \tag{9}$$

Gradients.

$$-\frac{\partial E^{\text{Hopfield}}}{\partial \boldsymbol{x}_i} \;=\; \sum_{\mu=1}^{K} \text{softmax}_\mu\big(\boldsymbol{x}_i^\top \boldsymbol{m}_\mu\big)\, \boldsymbol{m}_\mu. \tag{10}$$

$$-\frac{\partial E^{\text{Hopfield}}}{\partial \boldsymbol{m}_\mu} \;=\; \sum_{i=1}^{N} \text{softmax}_\mu\big(\boldsymbol{x}_i^\top \boldsymbol{m}_\mu\big)\, \boldsymbol{x}_i. \tag{11}$$

5 Slot Attention

Setup. Let $\boldsymbol{x}_j \in \mathbb{R}^d$, $j \in C = \{1, \ldots, N\}$ be the children (tokens), and $\boldsymbol{\mu}_i \in \mathbb{R}^d$, $i \in P = \{1, \ldots, S\}$ be the parents (slots). We typically apply linear transforms $\boldsymbol{W}_K, \boldsymbol{W}_Q \in \mathbb{R}^{d\times d}$ to form

$$\text{sim}\big(\boldsymbol{x}_j, \boldsymbol{\mu}_i\big) \;=\; \big(\boldsymbol{W}_K\, \boldsymbol{x}_j\big)^\top \big(\boldsymbol{W}_Q\, \boldsymbol{\mu}_i\big).$$

Energy.

$$E^{\text{Slot}}\Big(\{\boldsymbol{x}_j\}, \{\boldsymbol{\mu}_i\}\Big) \;=\; -\sum_{j=1}^{N} \ln\Big(\sum_{i=1}^{S} \exp\big(\text{sim}(\boldsymbol{x}_j, \boldsymbol{\mu}_i)\big)\Big). \tag{12}$$

Gradients.

$$-\frac{\partial E^{\text{Slot}}}{\partial \boldsymbol{x}_j} \;=\; \sum_{i=1}^{S} \text{softmax}_i\Big(\text{sim}(\boldsymbol{x}_j, \boldsymbol{\mu}_i)\Big)\, \boldsymbol{W}_K^\top \boldsymbol{W}_Q\, \boldsymbol{\mu}_i. \tag{13}$$

$$-\frac{\partial E^{\text{Slot}}}{\partial \boldsymbol{\mu}_i} \;=\; \sum_{j=1}^{N} \text{softmax}_i\Big(\text{sim}(\boldsymbol{x}_j, \boldsymbol{\mu}_i)\Big)\, \boldsymbol{W}_Q^\top \boldsymbol{W}_K\, \boldsymbol{x}_j. \tag{14}$$

6 Self-Attention

Setup. In self-attention, every node can act as both a child (query) and a parent (key). Concretely, let us have N tokens $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$. We form

$$\boldsymbol{q}_i \;=\; \boldsymbol{W}^Q\, \boldsymbol{x}_i, \quad \boldsymbol{k}_i \;=\; \boldsymbol{W}^K\, \boldsymbol{x}_i,$$

for $i = 1, \ldots, N$. Thus the set $C = \{1, \ldots, N\}$ and $P = \{1, \ldots, N\}$ coincide, with

$$\text{sim}\big(\boldsymbol{x}_c, \boldsymbol{x}_p\big) \;=\; \big(\boldsymbol{W}^Q \boldsymbol{x}_c\big)^\top \big(\boldsymbol{W}^K \boldsymbol{x}_p\big).$$

Energy.

$$E^{\text{SA}}\Big(\{\boldsymbol{x}_i\}\Big) \;=\; -\sum_{c=1}^{N} \ln\Big(\sum_{p=1}^{N} \exp\Big((\boldsymbol{W}^Q \boldsymbol{x}_c)^\top (\boldsymbol{W}^K \boldsymbol{x}_p)\Big)\Big). \tag{15}$$

Gradients. Since each $\boldsymbol{x}_i$ is *both* a child and a parent, its gradient is a sum of two terms (the child side and the parent side). Writing it out explicitly:

$$\begin{aligned} -\frac{\partial E^{\text{SA}}}{\partial \boldsymbol{x}_i} \;=\; & \underbrace{\sum_{p=1}^{N} \text{softmax}_p\Big((\boldsymbol{W}^Q \boldsymbol{x}_i)^\top (\boldsymbol{W}^K \boldsymbol{x}_p)\Big)\, \boldsymbol{W}_Q^\top \boldsymbol{W}_K\, \boldsymbol{x}_p}_{\text{child } i \text{ being explained by parents } p} \\ & + \underbrace{\sum_{c=1}^{N} \text{softmax}_i\Big((\boldsymbol{W}^Q \boldsymbol{x}_c)^\top (\boldsymbol{W}^K \boldsymbol{x}_i)\Big)\, \boldsymbol{W}_K^\top \boldsymbol{W}_Q\, \boldsymbol{x}_c}_{\text{parent } i \text{ explaining children } c}. \end{aligned} \tag{16}$$

7 Predictive coding

Setup. We again have child vectors $\{\boldsymbol{x}_i\}_{i=1}^N$ and a set of parent $\boldsymbol{\mu}_k \in \mathbb{R}^d, k = 1, \dots, K$. However, rather than a direct difference $\boldsymbol{x}_i - \boldsymbol{\mu}_k$, let us assume the *model* maps $\boldsymbol{\mu}_k$ through some non-linear function $f_\phi(\cdot)$ before comparing to $\boldsymbol{x}_i$. For instance, f_ϕ could be a neural network.

Similarity function. Define

$$\mathrm{sim}\big(\boldsymbol{x}_i, \boldsymbol{\mu}_k\big) \;=\; -\tfrac{1}{2}\,\big\|\boldsymbol{x}_i \;-\; f_\phi(\boldsymbol{\mu}_k)\big\|^2.$$

Energy.

$$E^{\mathrm{PC}}\Big(\{\boldsymbol{x}_i\}, \{\boldsymbol{\mu}_k\}\Big) \;=\; -\sum_{i=1}^{N} \ln\Big(\sum_{k=1}^{K} \exp\Big(-\tfrac{1}{2}\,\big\|\boldsymbol{x}_i - f_\phi(\boldsymbol{\mu}_k)\big\|^2\Big)\Big). \tag{17}$$

Gradients.

$$\alpha_{i,k} \;=\; \mathrm{softmax}_k\Big(-\tfrac{1}{2}\,\big\|\boldsymbol{x}_i - f_\phi(\boldsymbol{\mu}_k)\big\|^2\Big).$$

$$-\frac{\partial E^{\mathrm{PC}}}{\partial \boldsymbol{x}_i} \;=\; \sum_{k=1}^{K} \alpha_{i,k}\,\big(\boldsymbol{x}_i - f_\phi(\boldsymbol{\mu}_k)\big).$$

$$-\frac{\partial E^{\mathrm{PC}}}{\partial \boldsymbol{\mu}_k} \;=\; \sum_{i=1}^{N} \alpha_{i,k}\;\underbrace{\frac{\partial f_\phi(\boldsymbol{\mu}_k)}{\partial \boldsymbol{\mu}_k}}_{\text{Jacobian of } f_\phi}\;\big(\boldsymbol{x}_i - f_\phi(\boldsymbol{\mu}_k)\big).$$

Here, $\frac{\partial f_\phi(\boldsymbol{\mu}_k)}{\partial \boldsymbol{\mu}_k}$ is the $d \times d$ Jacobian (or more general shape if $\boldsymbol{\mu}_k$ and $f_\phi(\boldsymbol{\mu}_k)$ differ in dimension).

8 Switching Linear Dynamical System

Setup. Consider a time series of observations $\{\boldsymbol{x}_t\}_{t=1}^T$, where each $\boldsymbol{x}_t \in \mathbb{R}^d$. We assume there are K distinct (linear) dynamical modes, each with parameters $\{\boldsymbol{A}_k, \boldsymbol{b}_k, \boldsymbol{\Sigma}_k\}$. Let π_k be the mixing weight of mode k. A typical *switching linear dynamical system* (SLDS) posits:

$$\boldsymbol{x}_{t+1} \;\approx\; \boldsymbol{A}_k\,\boldsymbol{x}_t \;+\; \boldsymbol{b}_k, \quad k \in \{1, \dots, K\},$$

with Gaussian noise $\boldsymbol{\Sigma}_k$. We treat $\boldsymbol{x}_{t+1}$ as a *child* and the $\{\boldsymbol{A}_k, \boldsymbol{b}_k\}$ (together with $\boldsymbol{x}_t$) as *parents* in a mixture-of-linear-dynamics fashion.

Similarity function. Define, for each mode k,

$$\mathrm{sim}\big(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t; \boldsymbol{A}_k, \boldsymbol{b}_k\big) \;=\; \ln \pi_k \;-\; \tfrac{1}{2}\,\big(\boldsymbol{x}_{t+1} - \boldsymbol{A}_k\,\boldsymbol{x}_t - \boldsymbol{b}_k\big)^\top \boldsymbol{\Sigma}_k^{-1}\big(\boldsymbol{x}_{t+1} - \boldsymbol{A}_k\,\boldsymbol{x}_t - \boldsymbol{b}_k\big).$$

Energy. Summing over all time steps $t = 1, \dots, T-1$, we write

$$E^{\mathrm{SLDS}}\Big(\{\boldsymbol{x}_t\}, \{\boldsymbol{A}_k, \boldsymbol{b}_k\}\Big) \;=\; -\sum_{t=1}^{T-1} \ln\Big(\sum_{k=1}^{K} \exp\Big(\mathrm{sim}\big(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t; \boldsymbol{A}_k, \boldsymbol{b}_k\big)\Big)\Big). \tag{18}$$

Gradients. Let

$$\alpha_{t,k} \;=\; \mathrm{softmax}_k\Big(\mathrm{sim}\big(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t; \boldsymbol{A}_k, \boldsymbol{b}_k\big)\Big),$$

i.e. the normalized exponent for mode k.

$$-\frac{\partial E^{\mathrm{SLDS}}}{\partial \boldsymbol{x}_{t+1}} \;=\; \sum_{k=1}^{K} \alpha_{t,k}\,\boldsymbol{\Sigma}_k^{-1}\big(\boldsymbol{x}_{t+1} - \boldsymbol{A}_k\,\boldsymbol{x}_t - \boldsymbol{b}_k\big).$$

$$-\frac{\partial E^{\mathrm{SLDS}}}{\partial \boldsymbol{x}_t} \;=\; \sum_{k=1}^{K} \alpha_{t,k}\,\big(-\boldsymbol{A}_k^\top\,\boldsymbol{\Sigma}_k^{-1}\big)\,\big(\boldsymbol{x}_{t+1} - \boldsymbol{A}_k\,\boldsymbol{x}_t - \boldsymbol{b}_k\big) \quad (t = 1, \dots, T-1).$$

$$-\frac{\partial E^{\mathrm{SLDS}}}{\partial \boldsymbol{A}_k} \;=\; \sum_{t=1}^{T-1} \alpha_{t,k}\, \boldsymbol{\Sigma}_k^{-1}\big(\boldsymbol{x}_{t+1} - \boldsymbol{A}_k\, \boldsymbol{x}_t - \boldsymbol{b}_k\big)\, \boldsymbol{x}_t^\top.$$

$$-\frac{\partial E^{\mathrm{SLDS}}}{\partial \boldsymbol{b}_k} \;=\; \sum_{t=1}^{T-1} \alpha_{t,k}\, \boldsymbol{\Sigma}_k^{-1}\big(\boldsymbol{x}_{t+1} - \boldsymbol{A}_k\, \boldsymbol{x}_t - \boldsymbol{b}_k\big).$$

9 Layer Normalization

Setup. We consider a batch of input vectors $\{\boldsymbol{x}_i\}_{i=1}^B$, where each vector $\boldsymbol{x}_i \in \mathbb{R}^D$. Each vector is normalized by subtracting the mean and dividing by the standard deviation, with learnable scaling and bias parameters $\gamma, \delta \in \mathbb{R}^D$. For numerical stability, a small constant $\epsilon > 0$ is added to the variance.

Energy. The energy for layer normalization is given by:

$$E^{\mathrm{LN}}(\{\boldsymbol{x}_i\}) = \sum_{i=1}^{B} \left[\gamma \sqrt{\frac{1}{D}\sum_{j=1}^{D}(x_{ij} - \bar{\boldsymbol{x}}_i)^2 + \epsilon} + \sum_{j=1}^{D} \delta_j x_{ij} \right],$$

where:

$$\bar{\boldsymbol{x}}_i = \frac{1}{D}\sum_{j=1}^{D} x_{ij}.$$

Derivative. The normalized outputs are obtained as the derivative of the energy with respect to the inputs:

$$\frac{\partial E^{\mathrm{LN}}}{\partial x_{ij}} = \gamma_j \frac{x_{ij} - \bar{\boldsymbol{x}}_i}{\sqrt{\frac{1}{D}\sum_{k=1}^{D}(x_{ik} - \bar{\boldsymbol{x}}_i)^2 + \epsilon}} + \delta_j.$$

$$\bar{\boldsymbol{x}}_i = \frac{1}{D}\sum_{k=1}^{D} x_{ik}.$$

10 Spatiotemporal Self-Attention

We have K slots and T time steps, giving $K \times T$ total tokens. However, *instead of* using a single global set of projection matrices $\boldsymbol{W}^Q, \boldsymbol{W}^K, \boldsymbol{W}^V$ shared by all tokens, we now assume *each slot* k has its own distinct parameters $\boldsymbol{W}_k^Q, \boldsymbol{W}_k^K, \boldsymbol{W}_k^V$, but crucially **these parameters do not depend on time**. That is, slot k applies the same set of parameters $\boldsymbol{W}_k^Q, \boldsymbol{W}_k^K, \boldsymbol{W}_k^V$ at every time step t.

Setup. For each object-slot $k \in \{1, \ldots, K\}$ and time step $t \in \{1, \ldots, T\}$, we have a vector $\boldsymbol{x}_{k,t} \in \mathbb{R}^d$. We define:

$$\boldsymbol{q}_{k,t} \;=\; \boldsymbol{W}_k^Q\, \boldsymbol{x}_{k,t}, \quad \boldsymbol{k}_{k,t} \;=\; \boldsymbol{W}_k^K\, \boldsymbol{x}_{k,t}, \quad \boldsymbol{v}_{k,t} \;=\; \boldsymbol{W}_k^V\, \boldsymbol{x}_{k,t}.$$

Each slot k has its own learnable $\{\boldsymbol{W}_k^Q,\, \boldsymbol{W}_k^K,\, \boldsymbol{W}_k^V\}$, and these same matrices are used for all time steps t. Thus the model can specialize each slot's query-key-value mapping without introducing separate time-dependent parameters.

Full spatiotemporal attention. We concatenate all tokens (k, t) into a single set $\mathcal{S}$ of size $K \times T$. As before, we can define the pairwise similarity of token (k, t) to (k', t') by:

$$\mathrm{sim}\big(\boldsymbol{q}_{k,t}, \boldsymbol{k}_{k',t'}\big) \;=\; \boldsymbol{q}_{k,t}^\top\, \boldsymbol{k}_{k',t'}.$$

We then compute the attention "weights" (via a softmax over all (k', t')) and form a new representation $\widehat{\boldsymbol{x}}_{k,t}$ as a weighted sum of the corresponding $\boldsymbol{v}_{k',t'}$. Concretely:

$$\alpha_{(k,t)\to(k',t')} \;=\; \frac{\exp\big(\boldsymbol{q}_{k,t}^\top\, \boldsymbol{k}_{k',t'}\big)}{\sum_{(\tilde{k},\tilde{t})\in\mathcal{S}} \exp\big(\boldsymbol{q}_{k,t}^\top\, \boldsymbol{k}_{\tilde{k},\tilde{t}}\big)}, \quad \widehat{\boldsymbol{x}}_{k,t} \;=\; \sum_{(k',t')\in\mathcal{S}} \alpha_{(k,t)\to(k',t')}\, \boldsymbol{v}_{k',t'}. \tag{19}$$

Hence each token (k, t) attends freely to all other objects and all other time steps, yet the *projection* parameters are "slot-specific" but *time-invariant.* This can help each slot learn specialized ways of querying or encoding its own signals across time, while still using a single global attention mechanism to mix information among all (k', t').

Complexity. The computational cost remains $\mathcal{O}((KT)^2)$ for forming all pairwise dot products. Parameter-count-wise, we now have K distinct sets of $\boldsymbol{W}^Q, \boldsymbol{W}^K, \boldsymbol{W}^V$, each of size $d \times d$, but this still does *not* scale with T. We thus gain a "per-slot" specialization (rather than a single universal projection) but remain invariant to the number of time steps.

11 Coordinate ascent

Consider an energy of the form

$$E(\{\boldsymbol{x}_j\}, \{\boldsymbol{\mu}_i\}; \theta) \;=\; -\sum_{j=1}^{N} \ln\Big(\sum_{i=1}^{S} \exp\big(\mathrm{sim}_\theta(\boldsymbol{x}_j, \boldsymbol{\mu}_i)\big)\Big),$$

with θ denoting the parameters of the similarity function. In an EM procedure, we alternate:

- **E-step**: Update latent variables using fixed θ.
- **M-step**: Update parameters θ in closed form given fixed latent assignments.

For *slot attention*, we have:

$$\theta = \{\boldsymbol{W}_K, \boldsymbol{W}_Q\},$$
$$\mathrm{sim}_\theta(\boldsymbol{x}_j, \boldsymbol{\mu}_i) \;=\; (\boldsymbol{W}_K \boldsymbol{x}_j)^\top (\boldsymbol{W}_Q \boldsymbol{\mu}_i).$$
$$\mathbf{A} \quad \text{with entries} \quad A_{ji} \;=\; \mathrm{softmax}_i\Big((\boldsymbol{W}_K \boldsymbol{x}_j)^\top (\boldsymbol{W}_Q \boldsymbol{\mu}_i)\Big).$$

E-step: Update Slots

Fix $\boldsymbol{W}_K, \boldsymbol{W}_Q$. The gradient for each slot $\boldsymbol{\mu}_i$ is

$$-\frac{\partial E^{\mathrm{Slot}}}{\partial \boldsymbol{\mu}_i} = \sum_{j=1}^{N} A_{ji}\, \boldsymbol{W}_Q^\top \boldsymbol{W}_K \boldsymbol{x}_j.$$

Use this gradient to iteratively update $\{\boldsymbol{\mu}_i\}$ until convergence.

M-step: Update Parameters

Given fixed slots $\{\boldsymbol{\mu}_i\}$ and attention matrix $\mathbf{A}$, we aim to update the parameters $\boldsymbol{W}_K, \boldsymbol{W}_Q$. This is motivated by setting the gradient of the energy with respect to these parameters to zero:

$$-\frac{\partial E^{\mathrm{Slot}}}{\partial \boldsymbol{W}_K} = 0, \qquad -\frac{\partial E^{\mathrm{Slot}}}{\partial \boldsymbol{W}_Q} = 0.$$

Under the fixed assignments provided by $\mathbf{A}$ and slots $\{\boldsymbol{\mu}_i\}$, these conditions are equivalent to solving a weighted least-squares problem. Specifically, we consider minimizing the objective

$$\min_{\boldsymbol{W}_K, \boldsymbol{W}_Q} \sum_{j=1}^{N} \sum_{i=1}^{S} A_{ji} \left\| \boldsymbol{W}_K \boldsymbol{x}_j - \boldsymbol{W}_Q \boldsymbol{\mu}_i \right\|^2,$$

since the stationary point of this quadratic form corresponds to zero gradients with respect to $\boldsymbol{W}_K, \boldsymbol{W}_Q$.

Update $\boldsymbol{W}_Q$: Differentiate w.r.t. $\boldsymbol{W}_Q$, set to zero:

$$\sum_{j,i} A_{ji} \Big(\boldsymbol{W}_K \boldsymbol{x}_j - \boldsymbol{W}_Q \boldsymbol{\mu}_i\Big) \boldsymbol{\mu}_i^\top \;=\; 0,$$

yielding

$$\boldsymbol{W}_Q \;=\; \left(\sum_{j,i} A_{ji}\, \boldsymbol{W}_K \boldsymbol{x}_j\, \boldsymbol{\mu}_i^\top\right) \left(\sum_{j,i} A_{ji}\, \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top\right)^{-1}.$$

Update $\boldsymbol{W}_K$: Similarly, differentiate w.r.t. $\boldsymbol{W}_K$, set to zero:

$$\sum_{j,i} A_{ji} \Big(\boldsymbol{W}_K \boldsymbol{x}_j - \boldsymbol{W}_Q \boldsymbol{\mu}_i\Big) \boldsymbol{x}_j^\top \;=\; 0,$$

yielding

$$\boldsymbol{W}_K \;=\; \left(\sum_{j,i} A_{ji}\, \boldsymbol{W}_Q \boldsymbol{\mu}_i\, \boldsymbol{x}_j^\top\right) \left(\sum_{j,i} A_{ji}\, \boldsymbol{x}_j \boldsymbol{x}_j^\top\right)^{-1}.$$

Iterate: Alternate between the E-step (updating $\{\boldsymbol{\mu}_i\}$) and the M-step (updating $\boldsymbol{W}_K, \boldsymbol{W}_Q$) until convergence.

A Appendix: Derivation of Gradient Updates

Let us consider a generic term:

$$-\ln\Big(\sum_{m=1}^{M}\exp\big(f_m(\boldsymbol{x})\big)\Big),$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is some variable, and each f_m is a scalar function. We compute its gradient:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{x}}\Big[-\ln\Big(\sum_{m=1}^{M}\exp\big(f_m(\boldsymbol{x})\big)\Big)\Big] &= -\frac{1}{\sum_{m'}\exp\big(f_{m'}(\boldsymbol{x})\big)}\sum_{m=1}^{M}\exp\big(f_m(\boldsymbol{x})\big)\,\frac{\partial f_m(\boldsymbol{x})}{\partial \boldsymbol{x}}\\ &= -\sum_{m=1}^{M}\Big[\frac{\exp\big(f_m(\boldsymbol{x})\big)}{\sum_{m'}\exp\big(f_{m'}(\boldsymbol{x})\big)}\Big]\,\frac{\partial f_m(\boldsymbol{x})}{\partial \boldsymbol{x}}.\end{aligned}$$

Defining $\mathrm{softmax}_m\big(f(\boldsymbol{x})\big) = \frac{\exp(f_m(\boldsymbol{x}))}{\sum_{m'}\exp(f_{m'}(\boldsymbol{x}))}$, this is

$$-\sum_{m=1}^{M}\mathrm{softmax}_m\big(f(\boldsymbol{x})\big)\,\frac{\partial f_m(\boldsymbol{x})}{\partial \boldsymbol{x}},$$

which matches the softmax-weighted gradient structure.

B Appendix

Here, we collect the explicit partial derivatives of sim for each model discussed.

Gaussian Mixture Models

$$\mathrm{sim}\big(\boldsymbol{x}_i,\boldsymbol{\mu}_k\big) \;=\; \ln \pi_k \;-\; \tfrac{1}{2}\big(\boldsymbol{x}_i-\boldsymbol{\mu}_k\big)^{\top}\boldsymbol{\Sigma}_k^{-1}\big(\boldsymbol{x}_i-\boldsymbol{\mu}_k\big).$$

$$\frac{\partial}{\partial \boldsymbol{\mu}_k}\mathrm{sim}(\boldsymbol{x}_i,\boldsymbol{\mu}_k) \;=\; \boldsymbol{\Sigma}_k^{-1}\,\big(\boldsymbol{x}_i-\boldsymbol{\mu}_k\big).$$

Cross Attention

$$\mathrm{sim}\big(\boldsymbol{q}_c,\boldsymbol{k}_p\big) \;=\; \boldsymbol{q}_c^{\top}\boldsymbol{k}_p.$$

$$\frac{\partial}{\partial \boldsymbol{q}_c}\mathrm{sim}(\boldsymbol{q}_c,\boldsymbol{k}_p) \;=\; \boldsymbol{k}_p,\quad \frac{\partial}{\partial \boldsymbol{k}_p}\mathrm{sim}(\boldsymbol{q}_c,\boldsymbol{k}_p) \;=\; \boldsymbol{q}_c.$$

Hopfield Networks

$$\mathrm{sim}\big(\boldsymbol{x}_i,\boldsymbol{m}_\mu\big) \;=\; \boldsymbol{x}_i^{\top}\boldsymbol{m}_\mu.$$

$$\frac{\partial}{\partial \boldsymbol{x}_i}\mathrm{sim}(\boldsymbol{x}_i,\boldsymbol{m}_\mu) \;=\; \boldsymbol{m}_\mu,\quad \frac{\partial}{\partial \boldsymbol{m}_\mu}\mathrm{sim}(\boldsymbol{x}_i,\boldsymbol{m}_\mu) \;=\; \boldsymbol{x}_i.$$

Slot Attention

$$\mathrm{sim}\big(\boldsymbol{x}_j,\boldsymbol{\mu}_i\big) \;=\; (\boldsymbol{W}_K\,\boldsymbol{x}_j)^{\top}(\boldsymbol{W}_Q\,\boldsymbol{\mu}_i).$$

$$\frac{\partial}{\partial \boldsymbol{x}_j}\mathrm{sim}(\boldsymbol{x}_j,\boldsymbol{\mu}_i) \;=\; \boldsymbol{W}_K^{\top}\,\boldsymbol{W}_Q\,\boldsymbol{\mu}_i,\quad \frac{\partial}{\partial \boldsymbol{\mu}_i}\mathrm{sim}(\boldsymbol{x}_j,\boldsymbol{\mu}_i) \;=\; \boldsymbol{W}_Q^{\top}\,\boldsymbol{W}_K\,\boldsymbol{x}_j.$$

Self-Attention

$$\mathrm{sim}\big(\boldsymbol{x}_c,\boldsymbol{x}_p\big) \;=\; (\boldsymbol{W}^{Q}\boldsymbol{x}_c)^{\top}\,(\boldsymbol{W}^{K}\boldsymbol{x}_p).$$

$$\frac{\partial}{\partial \boldsymbol{x}_c}\mathrm{sim}(\boldsymbol{x}_c,\boldsymbol{x}_p) \;=\; \boldsymbol{W}^{Q\top}\,\boldsymbol{W}^{K}\,\boldsymbol{x}_p,\quad \frac{\partial}{\partial \boldsymbol{x}_p}\mathrm{sim}(\boldsymbol{x}_c,\boldsymbol{x}_p) \;=\; \boldsymbol{W}^{K\top}\,\boldsymbol{W}^{Q}\,\boldsymbol{x}_c.$$

Switching Linear Dynamical System

$$\mathrm{sim}\big(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t; \boldsymbol{A}_k, \boldsymbol{b}_k\big) \;=\; \ln \pi_k \;-\; \tfrac{1}{2}\left(\boldsymbol{x}_{t+1} - \boldsymbol{A}_k\,\boldsymbol{x}_t - \boldsymbol{b}_k\right)^{\top} \boldsymbol{\Sigma}_k^{-1}\big(\boldsymbol{x}_{t+1} - \boldsymbol{A}_k\,\boldsymbol{x}_t - \boldsymbol{b}_k\big).$$

$$\frac{\partial}{\partial \boldsymbol{x}_{t+1}} \mathrm{sim}(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t; \boldsymbol{A}_k, \boldsymbol{b}_k) \;=\; \boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{x}_{t+1} - \boldsymbol{A}_k\,\boldsymbol{x}_t - \boldsymbol{b}_k\right),$$

$$\frac{\partial}{\partial \boldsymbol{x}_{t}} \mathrm{sim}(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t; \boldsymbol{A}_k, \boldsymbol{b}_k) \;=\; -\,\boldsymbol{A}_k^{\top}\,\boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{x}_{t+1} - \boldsymbol{A}_k\,\boldsymbol{x}_t - \boldsymbol{b}_k\right),$$

$$\frac{\partial}{\partial \boldsymbol{A}_k} \mathrm{sim}(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t; \boldsymbol{A}_k, \boldsymbol{b}_k) \;=\; \boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{x}_{t+1} - \boldsymbol{A}_k\,\boldsymbol{x}_t - \boldsymbol{b}_k\right)\boldsymbol{x}_t^{\top},$$

$$\frac{\partial}{\partial \boldsymbol{b}_k} \mathrm{sim}(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t; \boldsymbol{A}_k, \boldsymbol{b}_k) \;=\; \boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{x}_{t+1} - \boldsymbol{A}_k\,\boldsymbol{x}_t - \boldsymbol{b}_k\right).$$

Predictive Coding

$$\mathrm{sim}\big(\boldsymbol{x}_i, \boldsymbol{\mu}_k\big) \;=\; -\tfrac{1}{2}\left\|\boldsymbol{x}_i \;-\; f_\phi(\boldsymbol{\mu}_k)\right\|^2.$$

$$\frac{\partial}{\partial \boldsymbol{x}_i}\,\mathrm{sim}(\boldsymbol{x}_i, \boldsymbol{\mu}_k) \;=\; \boldsymbol{x}_i - f_\phi(\boldsymbol{\mu}_k),$$

$$\frac{\partial}{\partial \boldsymbol{\mu}_k}\,\mathrm{sim}(\boldsymbol{x}_i, \boldsymbol{\mu}_k) \;=\; \frac{\partial f_\phi(\boldsymbol{\mu}_k)}{\partial \boldsymbol{\mu}_k}\,\big(\boldsymbol{x}_i - f_\phi(\boldsymbol{\mu}_k)\big).$$