Deriving DT Loss with Simple Example

1 Setup

Let's derive DT loss with simple example:

We have a two frame video: X_1, X_2

Each frame has a noise level: $K \sim \text{Uniform}(0, K)$ where 0 is no noise, K is full.

So $X_1^{K_1}, X_2^{K_2}$ are the two frames with their noise levels K_1, K_2 respectively.

2 Objective

Let us define our goal:

$$\arg\max_\theta \mathbb{E}_{X_1^0, X_2^0 \sim p_{\text{data}}} \left[\ln(p_\theta(X_1^0, X_2^0))\right] \tag{1}$$

We derive a surrogate objective via the ELBO:

$$\mathbb{E}_{X_1^0, X_2^0} \left[\ln(p_\theta(X_1^0, X_2^0))\right] \tag{2}$$

$$= \mathbb{E}_{X_1^0, X_2^0} \left[\ln \int p_\theta(X_1^0, X_2^0, X_1^{i:K}, X_2^{i:K}) dX_1^{i:K} dX_2^{i:K}\right] \tag{3}$$

Let's assume a Gaussian noising process:

$$\begin{aligned} q(X_i^k | X_i^0) &\sim \alpha_K X_i^0 + \beta_K \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) &(4)\\ &\sim \mathcal{N}(\alpha_K X_i^0, \beta_K I) &(5) \end{aligned}$$

$$\begin{aligned} &= \mathbb{E}_{X_1^0, X_2^0} \left[\ln \int p(X_1^{0:K}, X_2^{0:K}) \cdot \frac{q(X_1^{1:K}, X_2^{1:K} | X_1^0, X_2^0)}{q(X_1^{1:K}, X_2^{1:K} | X_1^0, X_2^0)} dX_1^{1:K} dX_2^{1:K}\right] &(6)\\ &= \mathbb{E}_{X_1^0, X_2^0 \sim p_{\text{data}}} \left[\ln \mathbb{E}_{X_1^{1:K}, X_2^{1:K} \sim q(\cdot | X_1^0, X_2^0)} \left[\frac{p(X_1^{0:K}, X_2^{0:K})}{q(X_1^{1:K}, X_2^{1:K} | X_1^0, X_2^0)}\right]\right] &(7)\\ &\geq \mathbb{E}_{X_1^0, X_2^0 \sim p_{\text{data}}, X_1^{1:K}, X_2^{1:K} \sim q(\cdot | X_1^0, X_2^0)} \left[\ln \left[\frac{p(X_1^{0:K}, X_2^{0:K})}{q(X_1^{1:K}, X_2^{1:K} | X_1^0, X_2^0)}\right]\right] &(8) \end{aligned}$$

$$p(X_1^{0:K}, X_2^{0:K}) = p(X_1^K, X_2^K) \prod_{i \in \tau} p(T_{i-1} | T_i), \quad \tau \sim \text{Traj}(2, K) \tag{9}$$

where $T_i = (X_1^{K_i}, X_2^{K_i})$, $T_0 = (X_1^0, X_2^0)$, $T_K = (X_1^K, X_2^K)$.

Traj$(2, K)$ is the set of all trajectories on a 2-dimensional cube with K delimiters. From 0 vertex to the opposite vertex. Note that we can construct the joint distribution in any order according to the denoising trajectories we allow, and we can assign any probability distribution over these trajectories. This flexibility allows us to model different generative processes by choosing different trajectory distributions. In the general case beyond the 2-frame example, we would replace 2 with n to get Traj(n, K), representing trajectories on an n-dimensional cube.

$$= \mathbb{E}_{X_1^{0:K}, X_2^{0:K}} \left[\ln p(X_1^K, X_2^K) + \sum_{i \in \text{Traj}(2,K)} \ln \left(\frac{p(T_{i-1}|T_i)}{q(T_i|T_{i-1})} \right) \right] \tag{10}$$

$$= \mathbb{E}_{X_1^{0:K}, X_2^{0:K}} \left[\ln p(X_1^K, X_2^K) \right] + \sum_{i \in \text{Traj}(2,K)} \mathbb{E}_{T_{i-1}, T_i} \left[\ln \left(\frac{p(T_{i-1}|T_i) \cdot q(T_{i-1})}{q(T_{i-1}|T_i) \cdot q(T_i)} \right) \right] \tag{11}$$

We take all constants w.r.t p out:

$$\mathbb{E}_{X_1^K, X_2^K} \left[\ln p(X_1^K, X_2^K) \right] + \sum_{i \in \text{Traj}(2,K)} \mathbb{E}_{T_{i-1}, T_i} \left[\ln \left(\frac{p(T_{i-1}|T_i)}{q(T_{i-1}|T_i)} \right) \right] \tag{12}$$

$$= \mathbb{E}_{X_1^K, X_2^K} \left[\ln p(X_1^K, X_2^K) \right] - \sum_{i \in \text{Traj}(2,K)} D_{KL}(q(T_{i-1}|T_i) || p(T_{i-1}|T_i)) \tag{13}$$

Note that $q(X_1^K, X_2^K) = p(X_1^K, X_2^K) = \mathcal{N}(0, I)$, and our model parameters typically do not change $p(X_1^K, X_2^K)$, so we can discard the first term. This gives us a simple sum of KL divergences over trajectories sampled from our trajectory distribution:

$$\mathcal{L} = - \sum_{\tau \sim \text{Traj}(2,K)} \sum_{i \in \tau} D_{KL}(q(T_{i-1}|T_i) || p(T_{i-1}|T_i)) \tag{14}$$

The probabilities assigned to each trajectory should correspond to the importance of each trajectory when sampling from the model.

We create indicator indicating if $X_1^{K_1}, X_2^{K_2} \leftrightarrow X_1^{K'_1}, X_2^{K'_2}$ is included in the trajectories of interest.

$\mathbb{I}(X_1^{K_1}, X_2^{K_2} \leftrightarrow X_1^{K'_1}, X_2^{K'_2})$

We can introduce probabilities of different trajectories: e.g. p_π

This modifies our loss by now having a weighted sum over the D_{KL} of our trajectories:

$$\mathbb{E}_{i \in \text{Traj}(2,K)_\pi} \left[D_{KL}(q(T_{i-1}|T_i) || p(T_{i-1}|T_i)) \right] \tag{15}$$

Now we sum over ALL possible 2^K trajectories:

$$= \sum_{i \in \text{Traj}(2,K)_\pi} p(\pi) \cdot D_{KL}(q(T_{i-1}|T_i) || p(T_{i-1}|T_i)) \tag{16}$$

3 Generalizing to the n-frame case

$$\sum_{i \in \text{Traj}(n,K)} P(T_i \to T_i') D_{KL}(q(T_{i-1}|T_i)||p(T_{i-1}|T_i)) \tag{17}$$

In DF we have $P(T_{i-1} \to T_i)$:

$$= \mathbb{I}\left(\left(K_1^{T_i}, K_2^{T_i}\right) - \left(K_1^{T_i-1}, K_2^{T_i-1}\right) = \left(K_1^{T_i}, K_2^{T_i} - 1\right)\right) \tag{18}$$

But indicator didn't draw all arrows, but indicator of all trajectories with

$$= (K-1)^n \tag{19}$$

but this should not be the case if we never double back and assign equal probability to each trajectory (you expect transitions of states near the diagonal to be weighted more by binomial...).

4 DF Weights

DF weights should be:

0	1	1	K
1	2		

trajectories containing $\{-, X_i^{K_i-1}, -\}$
$T_i \to T_i'$

We have:

$$\binom{K_1 + \cdots + K_n - 1}{K_1, K_2, \cdots, K_i - 1, \cdots, K_n} + \binom{(K-K_1) + \cdots + (K-K_n)}{K-K_1, \cdots, K-K_n} \tag{20}$$

$$= \binom{n \cdot K}{K, K, \cdots, K}_{n \text{ times}} \tag{21}$$

TF is encapsulated in this framework too.

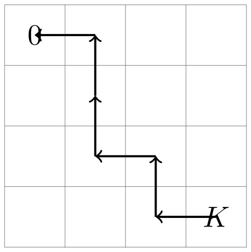
TF weights should be a constant traversing the edge of the cube from frame 1 $\to$ frame 2 $\to$ ... denoising which is:

$$\mathbb{I}\left(\left\{X_1^0, X_2^0, \ldots, X_i^{K_i}, X_{i+1}^K, \ldots, X_n^K\right\} \to \left\{X_1^0, X_2^0, \ldots, X_i^{K_i-1}, X_{i+1}^K, \ldots, X_n^{1K}\right\}\right) \tag{22}$$

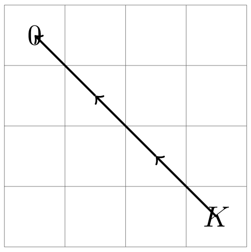
i.e. pick any frame i, every frame to the left should be clean, every frame to the right should be full noise, and the transition should be denoising the ith frame by 1 noise level.

But might seem like it is not what it is.

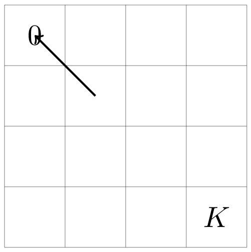
Transformers will simultaneously submit predictions for every frame, but in the run case, we just iterate through all frames (in any order) and denoise them by 1 step.



If we do architecture such as the transformer that predicts all frames at once, then our possible trajectories are just diagonals:



and at inference you just need



so that is the only trajectory trained in bi-directional/synchronous models.

5 Deriving the Self-Forcing Objective and Training Algorithm

Let's derive the self-forcing objective and training algorithm from the following:

Instead of $\arg\max_\theta \mathbb{E}_{X\sim p_{\text{data}}}\left[\ln(p_\theta(X))\right]$,

we swap p_{data} and p_θ to get:

$$\arg\max_\theta \mathbb{E}_{X_1^0, X_2^0 \sim p_\theta}\left[\ln(p_{\text{data}}(X_1^0, X_2^0))\right] \tag{23}$$

$$= \arg\max_\theta \mathbb{E}_{X_1^0, X_2^0 \sim p_\theta}\left[\ln \int p_{\text{data}}(X_1^{0:K}, X_2^{0:K}) \cdot \frac{p_\theta(X_1^{1:K}, X_2^{1:K}|X_1^0, X_2^0)}{p_\theta(X_1^{1:K}, X_2^{1:K}|X_1^0, X_2^0)} dX_1^{1:K} dX_2^{1:K}\right] \tag{24}$$

$$= \arg\max_\theta \mathbb{E}_{X_1^0, X_2^0 \sim p_\theta,}\left[\ln \mathbb{E}_{X_1^{1:K}, X_2^{1:K} \sim p_\theta(\cdot|X_1^0, X_2^0)}\left[\frac{p_{\text{data}}(X_1^{0:K}, X_2^{0:K})}{p_\theta(X_1^{1:K}, X_2^{1:K}|X_1^0, X_2^0)}\right]\right] \tag{25}$$

$$\geq \arg\max_\theta \mathbb{E}_{X_1^{0:K}, X_2^{0:K} \sim p_\theta}\left[\ln\left(\frac{p_{\text{data}}(X_1^{0:K}, X_2^{0:K})}{p_\theta(X_1^{1:K}, X_2^{1:K}|X_1^0, X_2^0)}\right)\right] \tag{26}$$