Proof Sketch: SF+GAN Minimizes Expected Jensen–Shannon Divergence

1 Objective

To prove that **SF+GAN** minimizes

$$\mathbb{E}_{noise}\Big[\,\text{JSD}\left(p_{\text{data}}(x|t)\,\|\,p_g(x|t)\right)\Big], \tag{1}$$

we adapt the original GAN proof to the video domain.

2 Adapting the GAN Framework

Our training procedure for SF+GAN creates a minimax game between the generator G and discriminator D:

$$\min_G \max_D V(G, D), \tag{2}$$

where

$$V(G, D) = \int_x p_{\text{data}}(x) \log D(x)\, dx + \int_z p_z(z) \log\big(1 - D(G(z))\big)\, dz. \tag{3}$$

During training, we sample minibatches from both the noise and data distributions and perform gradient ascent/descent updates:

$$\nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^{m} \left[\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))\right], \tag{4}$$

$$\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)}))). \tag{5}$$

3 Extension to Videos and Noise Levels

In SF+GAN, each x corresponds not to an image but to a **video**:

$$x_{0:T} = (x_0, x_1, \ldots, x_T), \tag{6}$$

and we perform the minimax game over noise levels $k_{0:T} \sim p(k)$.

Our sampling procedure is:

$$z_{0:T}^{(i)} \sim p_z, \quad \text{(noise)} \tag{7}$$

$$x_{0:T}^{(i)} \sim p_{\text{data}}, \quad \text{(real video)}. \tag{8}$$

For each batch, we randomly sample noise levels independently for real and generated videos:

$$k_{0:T}^{(i),\text{real}} \sim p(k), \tag{9}$$

$$k_{0:T}^{(i),\text{fake}} \sim p(k), \tag{10}$$

where each $k_{0:T} = (k_0, k_1, \ldots, k_T)$ specifies the noise level for each frame.

The generator G produces **clean frames** from noise:

$$\tilde{x}_{0:T}^{(i)} = G(z_{0:T}^{(i)}), \tag{11}$$

where $\tilde{x}_{0:T}^{(i)}$ represents the generated clean video.

Forward Process Definition

We define the **forward process** $F(x_{0:T}, k_{0:T})$ as a noising operation that takes clean frames and adds noise according to the schedule $k_{0:T}$. For each frame $t \in \{0, \ldots, T\}$, the forward process applies:

$$F(x_t, k_t) = \alpha_{k_t} x_t + \beta_{k_t} \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I), \tag{12}$$

where α_{k_t} and β_{k_t} are schedule-dependent coefficients that control the amount of noise added. More compactly, for a video $x_{0:T}$ and schedule $k_{0:T}$:

$$F(x_{0:T}, k_{0:T}) = \big(F(x_0, k_0), F(x_1, k_1), \ldots, F(x_T, k_T)\big). \tag{13}$$

During training, we apply the forward process to **both** real and generated videos using independently sampled noise schedules:

$$x_{0:T}^{(i),\text{noisy}} = F(x_{0:T}^{(i)}, k_{0:T}^{(i),\text{real}}), \tag{14}$$

$$\tilde{x}_{0:T}^{(i),\text{noisy}} = F(\tilde{x}_{0:T}^{(i)}, k_{0:T}^{(i),\text{fake}}) = F(G(z_{0:T}^{(i)}), k_{0:T}^{(i),\text{fake}}). \tag{15}$$

4 Training Updates

The SF+GAN training procedure is summarized in Algorithm 1. We update the discriminator by ascending:

$$\nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^{m} \Big[\log D\big(F(x_{0:T}^{(i)}, k_{0:T}^{(i),\text{real}}); k_{0:T}^{(i),\text{real}}\big) + \log\big(1 - D(F(G(z_{0:T}^{(i)}), k_{0:T}^{(i),\text{fake}}); k_{0:T}^{(i),\text{fake}})\big) \Big], \tag{16}$$

and update the generator by descending:

$$\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} \log\big(1 - D(F(G(z_{0:T}^{(i)}), k_{0:T}^{(i),\text{fake}}); k_{0:T}^{(i),\text{fake}})\big). \tag{17}$$

Note that the generator G produces clean frames $\tilde{x}_{0:T}^{(i)} = G(z_{0:T}^{(i)})$, and then the forward process F is applied independently to the real clean videos $x_{0:T}^{(i)}$ and the generated clean videos $\tilde{x}_{0:T}^{(i)}$ using their respective randomly sampled noise schedules.

Algorithm 1: SF+GAN Training for Videos

```
Input: Number of training iterations $N$, batch size $m$, noise distribution $p_z$, data
       distribution $p_{\text{data}}$, noise schedule distribution $p(k)$, forward process $F$, generator
       $G$, discriminator $D$
Output: Trained generator $G$
Initialize generator $G$ and discriminator $D$ with random weights;
for training iteration $n = 1, 2, \ldots, N$ do
    Sample minibatch of noise: $\{z_{0:T}^{(i)}\}_{i=1}^{m} \sim p_z$;
    Sample minibatch of real videos: $\{x_{0:T}^{(i)}\}_{i=1}^{m} \sim p_{\text{data}}$;
    Sample noise schedules for real videos: $\{k_{0:T}^{(i),\text{real}}\}_{i=1}^{m} \sim p(k)$;
    Sample noise schedules for fake videos: $\{k_{0:T}^{(i),\text{fake}}\}_{i=1}^{m} \sim p(k)$;
    Generate clean videos: $\tilde{x}_{0:T}^{(i)} = G(z_{0:T}^{(i)})$ for $i = 1, \ldots, m$;
    Apply forward process to real videos: $x_{0:T}^{(i),\text{noisy}} = F(x_{0:T}^{(i)}, k_{0:T}^{(i),\text{real}})$ for $i = 1, \ldots, m$;
    Apply forward process to generated videos: $\tilde{x}_{0:T}^{(i),\text{noisy}} = F(\tilde{x}_{0:T}^{(i)}, k_{0:T}^{(i),\text{fake}})$ for $i = 1, \ldots, m$;
    Update discriminator by ascending gradient:;
    $\nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D(x_{0:T}^{(i),\text{noisy}}; k_{0:T}^{(i),\text{real}}) + \log(1 - D(\tilde{x}_{0:T}^{(i),\text{noisy}}; k_{0:T}^{(i),\text{fake}})) \right]$;
    Update generator by descending gradient:;
    $\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(\tilde{x}_{0:T}^{(i),\text{noisy}}; k_{0:T}^{(i),\text{fake}}))$;
```

5 Final Objective

Combining these components, the SF+GAN objective can be written as:

$$\begin{aligned} V(G, D) = \mathbb{E}_{k_{0:T}^{\text{real}}, k_{0:T}^{\text{fake}} \sim p(k)} \Bigg[&\int_{x_{0:T}} p_{\text{data}}(x_{0:T}) \log D\big(F(x_{0:T}, k_{0:T}^{\text{real}}); k_{0:T}^{\text{real}}\big)\, dx_{0:T} \\ &+ \int_{z_{0:T}} p_z(z_{0:T}) \log\big(1 - D(F(G(z_{0:T}), k_{0:T}^{\text{fake}}); k_{0:T}^{\text{fake}})\big)\, dz_{0:T} \Bigg], \end{aligned} \tag{18}$$

where $G(z_{0:T})$ produces clean generated videos, and the forward process F is applied independently to real videos $x_{0:T}$ and generated videos $G(z_{0:T})$ using their respective independently sampled noise schedules $k_{0:T}^{\text{real}}$ and $k_{0:T}^{\text{fake}}$.

6 Deriving the Optimal Discriminator and JSD

To find the optimal discriminator and connect the objective to Jensen–Shannon divergence, we follow the derivation from the original GAN paper, adapted to our conditional setting.

Step 1: Combining Integrals

First, we change variables in the second integral. Let $\tilde{x}_{0:T} = G(z_{0:T})$ be the generated clean video, and after applying the forward process, we have $\tilde{x}_{0:T}^{\text{noisy}} = F(\tilde{x}_{0:T}, k_{0:T}^{\text{fake}})$. Let $p_g(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{fake}})$ denote the distribution of noisy generated videos conditioned on the noise schedule.

For a fixed noise schedule pair $(k_{0:T}^{\text{real}}, k_{0:T}^{\text{fake}})$, the objective becomes:

$$\begin{aligned} V(G, D; k_{0:T}^{\text{real}}, k_{0:T}^{\text{fake}}) = &\int_{x_{0:T}} p_{\text{data}}(x_{0:T}) \log D\big(F(x_{0:T}, k_{0:T}^{\text{real}}); k_{0:T}^{\text{real}}\big)\, dx_{0:T} \\ &+ \int_{z_{0:T}} p_z(z_{0:T}) \log\big(1 - D(F(G(z_{0:T}), k_{0:T}^{\text{fake}}); k_{0:T}^{\text{fake}})\big)\, dz_{0:T}. \end{aligned} \tag{19}$$

Changing variables in the second integral, we can rewrite this as:

$$\begin{aligned} V(G, D; k_{0:T}^{\text{real}}, k_{0:T}^{\text{fake}}) = &\int_{x_{0:T}^{\text{noisy}}} p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{real}}) \log D(x_{0:T}^{\text{noisy}}; k_{0:T}^{\text{real}})\, dx_{0:T}^{\text{noisy}} \\ &+ \int_{x_{0:T}^{\text{noisy}}} p_g(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{fake}}) \log\big(1 - D(x_{0:T}^{\text{noisy}}; k_{0:T}^{\text{fake}})\big)\, dx_{0:T}^{\text{noisy}}, \end{aligned} \tag{20}$$

where $p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{real}})$ is the distribution of noisy real videos after applying F with schedule $k_{0:T}^{\text{real}}$, and similarly for p_g.

Combining both terms under a single integral:

$$V(G, D; k_{0:T}^{\text{real}}, k_{0:T}^{\text{fake}}) = \int_{x_{0:T}^{\text{noisy}}} \Big[p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{real}}) \log D(x_{0:T}^{\text{noisy}}; k_{0:T}^{\text{real}}) + p_g(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{fake}}) \log\big(1 - D(x_{0:T}^{\text{noisy}}; k_{0:T}^{\text{fake}})\big)\Big]\, dx_{0:T}^{\text{noisy}}. \tag{21}$$

Step 2: Finding the Optimal Discriminator

For any $(a, b) \in \mathbb{R}^2 \setminus \{(0, 0)\}$, the function $y \mapsto a \log(y) + b \log(1 - y)$ achieves its maximum in $[0, 1]$ at $y = a/(a + b)$.

Applying this to the integrand for each fixed $x_{0:T}^{\text{noisy}}$ and noise schedules, the optimal discriminator that maximizes $V(G, D; k_{0:T}^{\text{real}}, k_{0:T}^{\text{fake}})$ for a fixed generator G is:

$$D_G^*(x_{0:T}^{\text{noisy}}; k_{0:T}^{\text{real}}, k_{0:T}^{\text{fake}}) = \frac{p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{real}})}{p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{real}}) + p_g(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{fake}})}. \tag{22}$$

Note that for the discriminator to be well-defined, we need to consider it conditioned on the noise schedule. Since noise schedules are sampled independently, we can write the optimal discriminator more generally as:

$$D_G^*(x_{0:T}^{\text{noisy}} | k_{0:T}) = \frac{p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T})}{p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}) + p_g(x_{0:T}^{\text{noisy}} | k_{0:T})}. \tag{23}$$

Step 3: Substituting the Optimal Discriminator

Now we define $C(G) = \max_D V(G, D)$ and substitute the optimal discriminator:

$$\begin{aligned} C(G) = \mathbb{E}_{k_{0:T}^{\text{real}}, k_{0:T}^{\text{fake}} \sim p(k)} \Bigg[&\int_{x_{0:T}^{\text{noisy}}} p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{real}}) \log D_G^*(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{real}})\, dx_{0:T}^{\text{noisy}} \\ &+ \int_{x_{0:T}^{\text{noisy}}} p_g(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{fake}}) \log\big(1 - D_G^*(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{fake}})\big)\, dx_{0:T}^{\text{noisy}} \Bigg]. \end{aligned} \tag{24}$$

Substituting D_G^*:

$$C(G) = \mathbb{E}_{k_{0:T}^{\text{real}}, k_{0:T}^{\text{fake}} \sim p(k)} \left[\int_{x_{0:T}^{\text{noisy}}} p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{real}}) \log \frac{p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{real}})}{p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{real}}) + p_g(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{real}})} \, dx_{0:T}^{\text{noisy}} \right. \\ \left. + \int_{x_{0:T}^{\text{noisy}}} p_g(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{fake}}) \log \frac{p_g(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{fake}})}{p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{fake}}) + p_g(x_{0:T}^{\text{noisy}} | k_{0:T}^{\text{fake}})} \, dx_{0:T}^{\text{noisy}} \right]. \tag{25}$$

Step 4: Expressing as Sum of KL Divergences

To introduce KL divergence terms, we add and subtract $\log(1/2)$ in each integral:

$$\begin{aligned} C(G) = \mathbb{E}_{k_{0:T} \sim p(k)} \Bigg[& \int_{x_{0:T}^{\text{noisy}}} p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}) \log \frac{p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T})}{(p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}) + p_g(x_{0:T}^{\text{noisy}} | k_{0:T}))/2} \, dx_{0:T}^{\text{noisy}} \\ & + \int_{x_{0:T}^{\text{noisy}}} p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}) \log(1/2) \, dx_{0:T}^{\text{noisy}} \\ & + \int_{x_{0:T}^{\text{noisy}}} p_g(x_{0:T}^{\text{noisy}} | k_{0:T}) \log \frac{p_g(x_{0:T}^{\text{noisy}} | k_{0:T})}{(p_{\text{data}}(x_{0:T}^{\text{noisy}} | k_{0:T}) + p_g(x_{0:T}^{\text{noisy}} | k_{0:T}))/2} \, dx_{0:T}^{\text{noisy}} \\ & + \int_{x_{0:T}^{\text{noisy}}} p_g(x_{0:T}^{\text{noisy}} | k_{0:T}) \log(1/2) \, dx_{0:T}^{\text{noisy}} \Bigg], \end{aligned} \tag{26}$$

where we have used the fact that $k_{0:T}^{\text{real}}$ and $k_{0:T}^{\text{fake}}$ are independently sampled from the same distribution $p(k)$.

Recognizing the Kullback–Leibler divergence $\text{KL}(P \| Q) = \int P(x) \log(P(x)/Q(x)) \, dx$, we obtain:

$$\begin{aligned} C(G) = \mathbb{E}_{k_{0:T} \sim p(k)} \Bigg[& \text{KL}\left(p_{\text{data}}(\cdot | k_{0:T}) \,\Big\|\, \frac{p_{\text{data}}(\cdot | k_{0:T}) + p_g(\cdot | k_{0:T})}{2} \right) \\ & + \log(1/2) \\ & + \text{KL}\left(p_g(\cdot | k_{0:T}) \,\Big\|\, \frac{p_{\text{data}}(\cdot | k_{0:T}) + p_g(\cdot | k_{0:T})}{2} \right) \\ & + \log(1/2) \Bigg]. \end{aligned} \tag{27}$$

Step 5: Arriving at Jensen–Shannon Divergence

Combining the terms:

$$\begin{aligned} C(G) = -2\log 2 + \mathbb{E}_{k_{0:T} \sim p(k)} \Bigg[& \text{KL}\left(p_{\text{data}}(\cdot | k_{0:T}) \,\Big\|\, \frac{p_{\text{data}}(\cdot | k_{0:T}) + p_g(\cdot | k_{0:T})}{2} \right) \\ & + \text{KL}\left(p_g(\cdot | k_{0:T}) \,\Big\|\, \frac{p_{\text{data}}(\cdot | k_{0:T}) + p_g(\cdot | k_{0:T})}{2} \right) \Bigg]. \end{aligned} \tag{28}$$

The Jensen–Shannon divergence is defined as:

$$\text{JSD}(P \| Q) = \frac{1}{2} \text{KL}\left(P \,\Big\|\, \frac{P+Q}{2} \right) + \frac{1}{2} \text{KL}\left(Q \,\Big\|\, \frac{P+Q}{2} \right), \tag{29}$$

so the sum of the two KL terms equals $2 \cdot \mathrm{JSD}(p_{\mathrm{data}}(\cdot|k_{0:T}) \| p_g(\cdot|k_{0:T}))$.

Therefore, we have:

$$C(G) = -2\log 2 + 2\,\mathbb{E}_{k_{0:T}\sim p(k)}\Big[\mathrm{JSD}\left(p_{\mathrm{data}}(\cdot|k_{0:T}) \,\|\, p_g(\cdot|k_{0:T})\right)\Big]. \tag{30}$$

At optimal discriminator $D^*(x|k)$, we recover that

$$V(G, D^*) = -2\log 2 + 2\,\mathbb{E}_{k_{0:T}\sim p(k)}\Big[\mathrm{JSD}\left(p_{\mathrm{data}}(x|t) \,\|\, p_g(x|t)\right)\Big]. \tag{31}$$

Since $-2\log 2$ is a constant, we have

$$V(G, D^*) \propto \mathbb{E}_{k_{0:T}\sim p(k)}\Big[\mathrm{JSD}\left(p_{\mathrm{data}}(x|t) \,\|\, p_g(x|t)\right)\Big], \tag{32}$$

so minimizing $V(G, D^*)$ is equivalent to minimizing

$$\mathbb{E}_{k_{0:T}\sim p(k)}\Big[\mathrm{JSD}\left(p_{\mathrm{data}}(x|t) \,\|\, p_g(x|t)\right)\Big]. \tag{33}$$

This shows that the SF+GAN training minimizes the expected Jensen–Shannon divergence between real and generated conditional distributions.

7 Conclusion

Thus, by extending the original GAN framework to operate over time-conditioned video distributions and noise levels $k_{0:T}$, we see that the SF+GAN formulation is equivalent to minimizing an *expected conditional Jensen–Shannon divergence* similar to the GAN objective derived in the Diffusion-GAN paper.