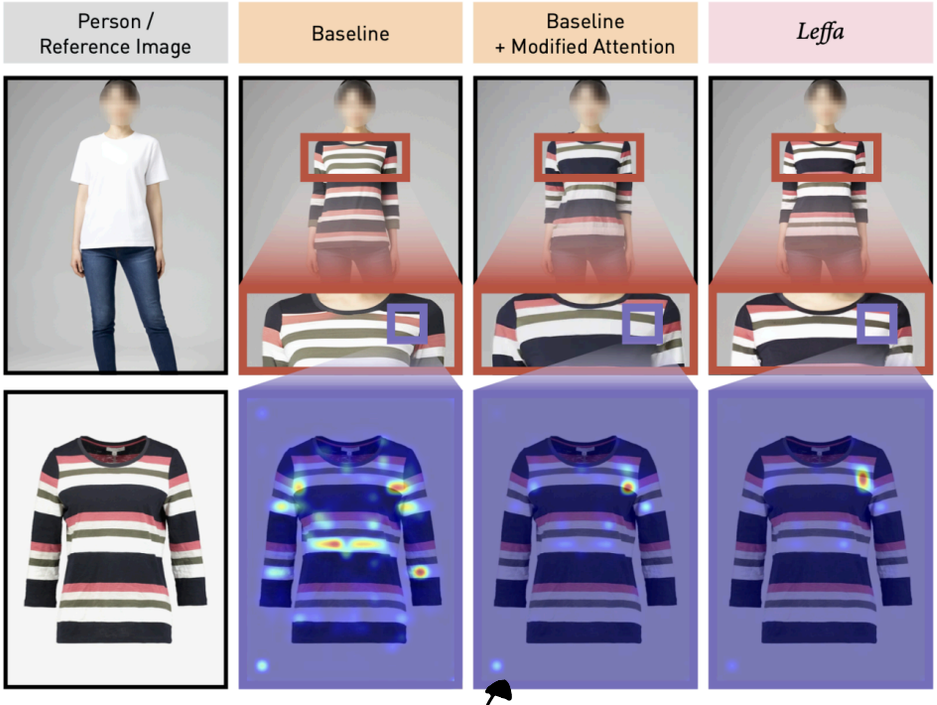
Leffa → generate a person image conditioned on reference images

↳ learning flow fields in attention

↳ attention mechanism that tells the model which part of the "reference" image should map the target



↳ it was manually changed

↳ specifically a loss function

Leffa is a new training method

↳ plugged into any diffusion-based approach

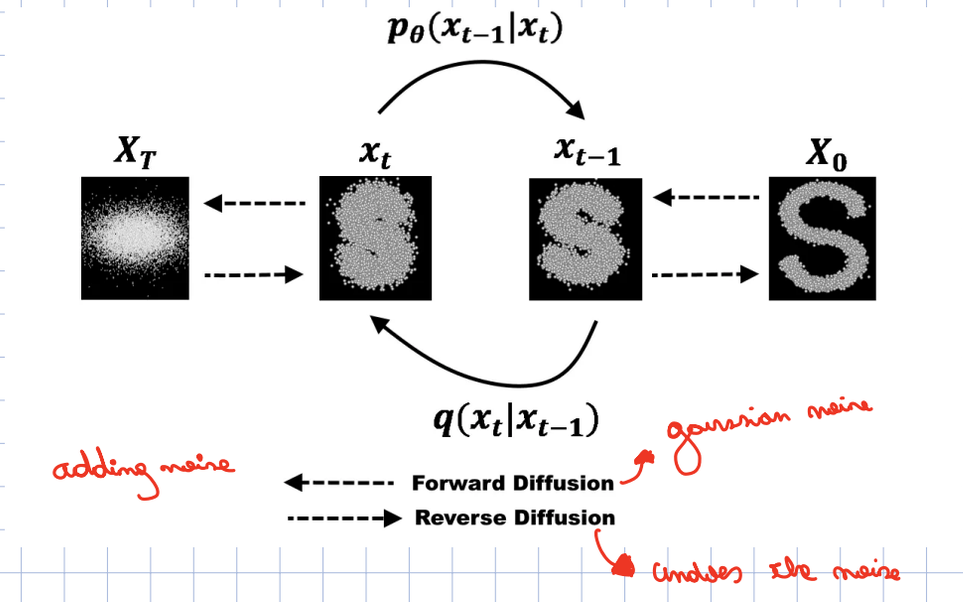
Steps: → VAE (Variational autoencoder) to compress

reduce computational cost

(I) → Encoder $\mathcal{E}$ → pixel space into latent space (I_0)

II pick a timestep $t \in \{1, \dots, T\}$

Gaussian noise $\varepsilon \sim \mathcal{N}(0,1)$ with a noise scheduler



- diffusion models learn to reverse the process of noising an image
  → the model practices "denoising"
- this process is like molecules spreading out in a gas
  ↳ in physics, diffusion spreads particles from high density to low-density

forward process

- a Markov chain → each step only depends on the immediately prev step
  → apply gaussian noise in each step

$$q(x_{0:T}|x_0) = q(x_0) \prod_{t=1}^{T} q(x_t|x_{t-1})$$

- Langevin Dynamics → sampling from a distribution by using the grad
  ↳ "tiny random walk"

Reconstruction

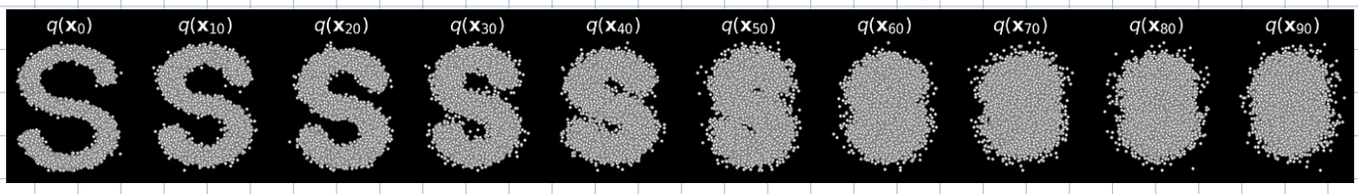
↳ we know how to go from $x_{t-1} \xrightarrow{to} x_t$

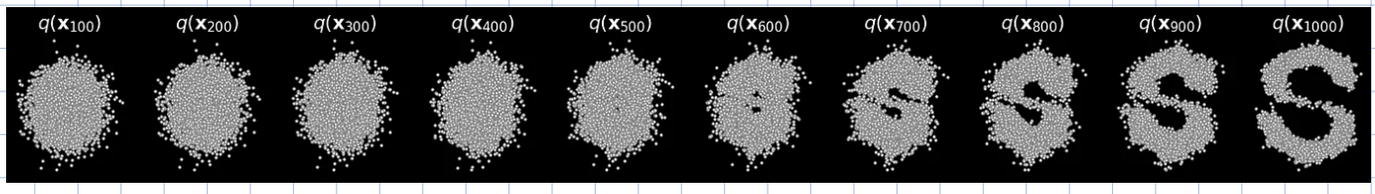
↳ $x_t \xrightarrow{to} x_{t-1}$ (train a Neural Net) U-Net that predicts how to remove noise from x_t

The neural network answers "Given the noisy image x_t, which pixels (and how) should we remove to get a cleaner version x_{t-1}"

↳ outputs the mean and variance

Diffusion noise





↑ reverse process

III

- Virtual try-on process
  - Garment masks → binary mask indicating which pixels belong to the garment
  - eliminate the garment
  - Add noise
- Pose transfer → "these are the person's appearance details (face, cloth) to preserve"

↳ DensePose image to capture body language

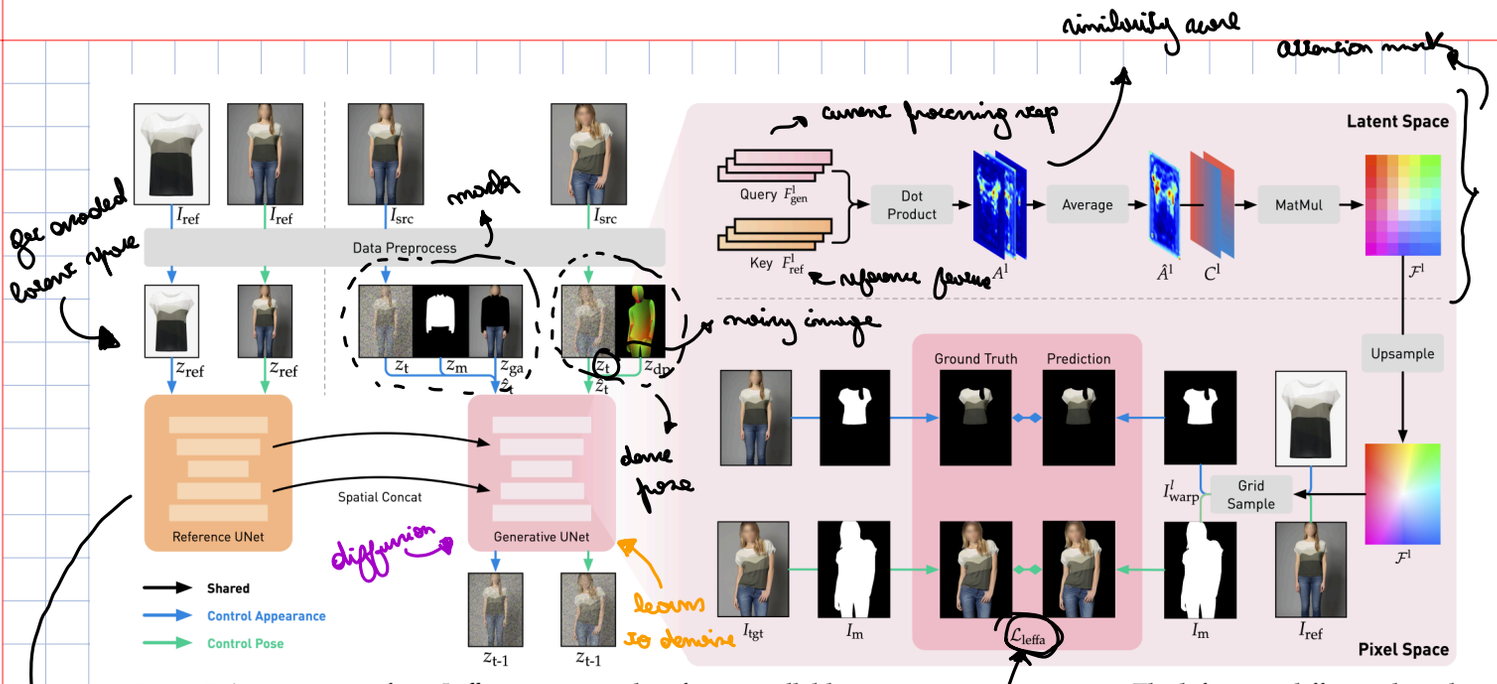


Figure 3 An overview of our *Leffa* training pipeline for controllable person image generation. The left is our diffusion-based baseline; the right is our *Leffa* loss. Note that I_{src} and I_{tgt} are the same image during training.

- $Q \in \mathbb{R}^{m \times m}$ (the current feature map)
- $K \in \mathbb{R}^{n \times m}$ (reference map)
- $(QK)^t$ → similarity score for "which reference spots are similar to my current location"
- Softmax (probs)

z_t

1. **You have a noisy latent** z_t that you feed into the Generative UNet.
2. **Within** the UNet, each layer outputs a feature map—call the l-th layer's output $\mathbf{F}^l_{gen}$.
3. **Meanwhile**, the Reference UNet produces its own feature map $\mathbf{F}^l_{ref}$ describing the garment/person in the reference image.
4. **Cross-attention** at layer l takes:
   - $\mathbf{F}^l_{gen}$ as "Query,"
   - $\mathbf{F}^l_{ref}$ as "Key/Value."
   - It aligns them to inject the correct garment/pose details into the Generative UNet's feature map.

At the **end** of the Generative UNet, you get a predicted **denoised latent** $zt-1$. *That* $zt-1$ is still in the VAE's latent space; it's just less noisy than z_t. When you finally reach z_0 (fully denoised), you **decode** it back to an RGB image.

a less flexible representation

Leffa loss → alleviates detail distortion through learning flow fields in attention

divergence arises when the target query in attn layer fails to attend to the corresponding region

target → $\mathcal{L}_{affa}$ loss guides the target query to spatially focus on the correct region

↳ based on the attention map

(!!) The problem was that the attention layer may be looking at the wrong place

flow field :D maps an output pixel to a reference pixel

↳ compare to ground truth → learn not to diverge

$\mathcal{L}_{final} = \mathcal{L}_{diffusion} + \lambda_{affa} \mathcal{L}_{affa}$ ← how weight of $\mathcal{L}_{affa}$

Self-Correction for Human Parsing

image pixel from human body ⟶ semantic category

target → improve the model performance and generalization by progressively refining the noisy labels during training

SCHP → purification strategy → label refinement procedure

↳ self correction mechanism

↳ correct the noisy training label via a model and label renewable promoting process.

Methodology

CE2P → parsing branch, edge branch, fusion branch → boundary maps → combined to produce a refined human parsing feed

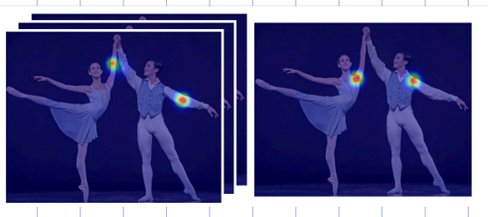
OpenPose (Method)

1. 2D confidence map (S) ∧ 2D Vector fields L (PAFs)

where ...

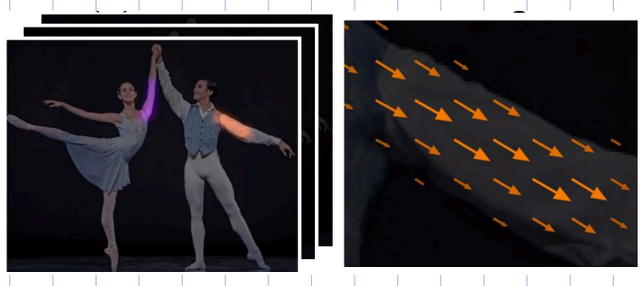
$S = (S_1, S_2, \dots, S_J)$ — $S_J \in \mathbb{R}^{w \times h}$ (width × height) — body part

J → confidence map



$L = (L_1, L_2, \dots, L_C)$ — C → one for vector field

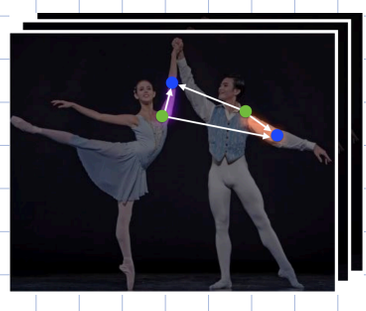
$L_C \in \mathbb{R}^{w \times h \times 2}$ → encodes a 2D vector



finally

Joins by greedy inference

$S_J \cap L_C$ → output



Network Architecture

- The iterative prediction architecture refines predictions over stages $t \in \{1, \ldots, T\}$ with intermediate supervision at each stage

Simultaneous Detection and Association

- features maps of $F \xrightarrow{\text{produces}} L^1 = \phi^1(F)$ (PAFs), Stage 1

so at each stage $L^t = \phi^t(F, L^{t-1}), \forall (2 \le t \le T_P)$ — total TAPs

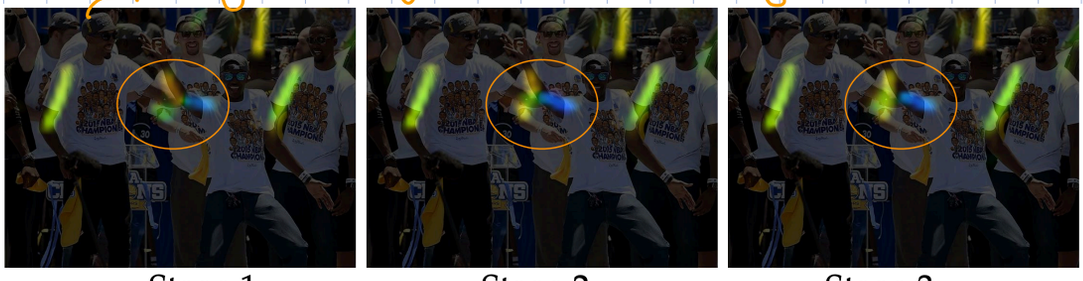
↳ output previous L^{t-1}

Same for confidence maps

$S^t = \rho^t(F, L^{T_P}, S^{t-1}), \quad \forall T_P < t \le T_P + T_C$

refers to the CNN for inference at stage t

refinement of attention over stages



Stage 1 Stage 2 Stage 3

Confidence maps for Part detection

↳ generate individual confidence maps $S^*_{j,k}$ for each person k.

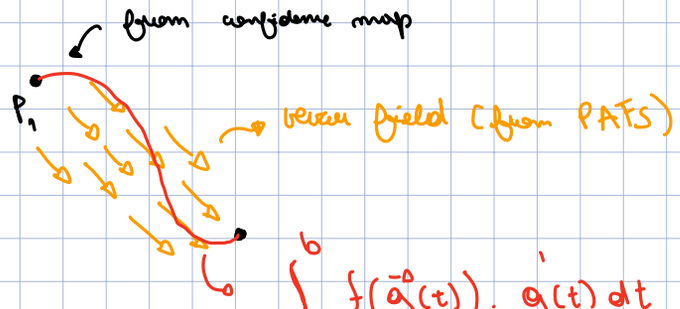
!! Decide which pairs belong together to form a person's pose

PAFs → 2D vector field for each limb

↳ preserve both location and orientation information across the region of support of the limb

ground truth

Part association and Pose assembly



$$\int_a^b f(\vec{g}(t)) \cdot g'(t)\, dt$$

↳ sum up how well PAFs vectors align with the direction

DensePose → mapping all human pixels of RGB to the 3D-Surface

↳ COCO Dataset ground-truth labeled Img 2 Surface

Diffusion model:

↳ slowly destroy structure in a data distribution

↳ then

↳ learn a reverse diffusion process that restores structure in data

forward diffusion process

↳ each step of the forward diffusion process is defined as

$$q(x_t / x_{t-1}) = \sim N\left(x_t, \sqrt{1-\beta_t}\, x_{t-1}; \beta_t \mathbb{I}\right)$$

x_t ↳ output of forward process

$\sqrt{1-\beta_t}\, x_{t-1}$ ↳ μ

$\beta_t \mathbb{I}$ ↳ σ

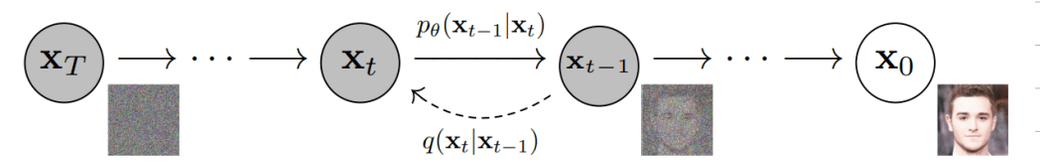
Schedule → range from 0 to 1

↳ OpenAI in 2021 proposed cosine schedule

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}\, x_0; (1-\bar{\alpha}_t) I) = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\,\varepsilon$$

Reverse diffusion process

↳ Convert pure noise into an image



- Diffusion model predicts the entire noise to be removed in a given timestep

- diffusion model —predicts→ noise present in the current image

then ...

scheduler (reverse process) subtracts the predicted noise from the image

Architecture

↳ U-Net with embeddings

→ each step in the process adds embeddings with information about current timestep and prompt

Resnet - blocks (residual networks)

- allow deeper networks to learn residual functions rather than direct mapping
- learn the difference between input and desired output

↓

- If best mapping were simply to keep input changed then residual $F(x) = 0$
- diffusion models: used to process and refine feature representations at various resolutions.

Downsample block → reduces spatial resolution (height × width) of feature maps

- ↑ receptive field ⇒ integrate context over larger area of img
- Diffusion models: reduce compute cost and refine feature representations

Self-attention blocks → allow position in the feature map

↗ use of positional encoding