

CONDITIONAL DIFFUSION

- learns to generate edited images based on two inputs:

- original image & style label

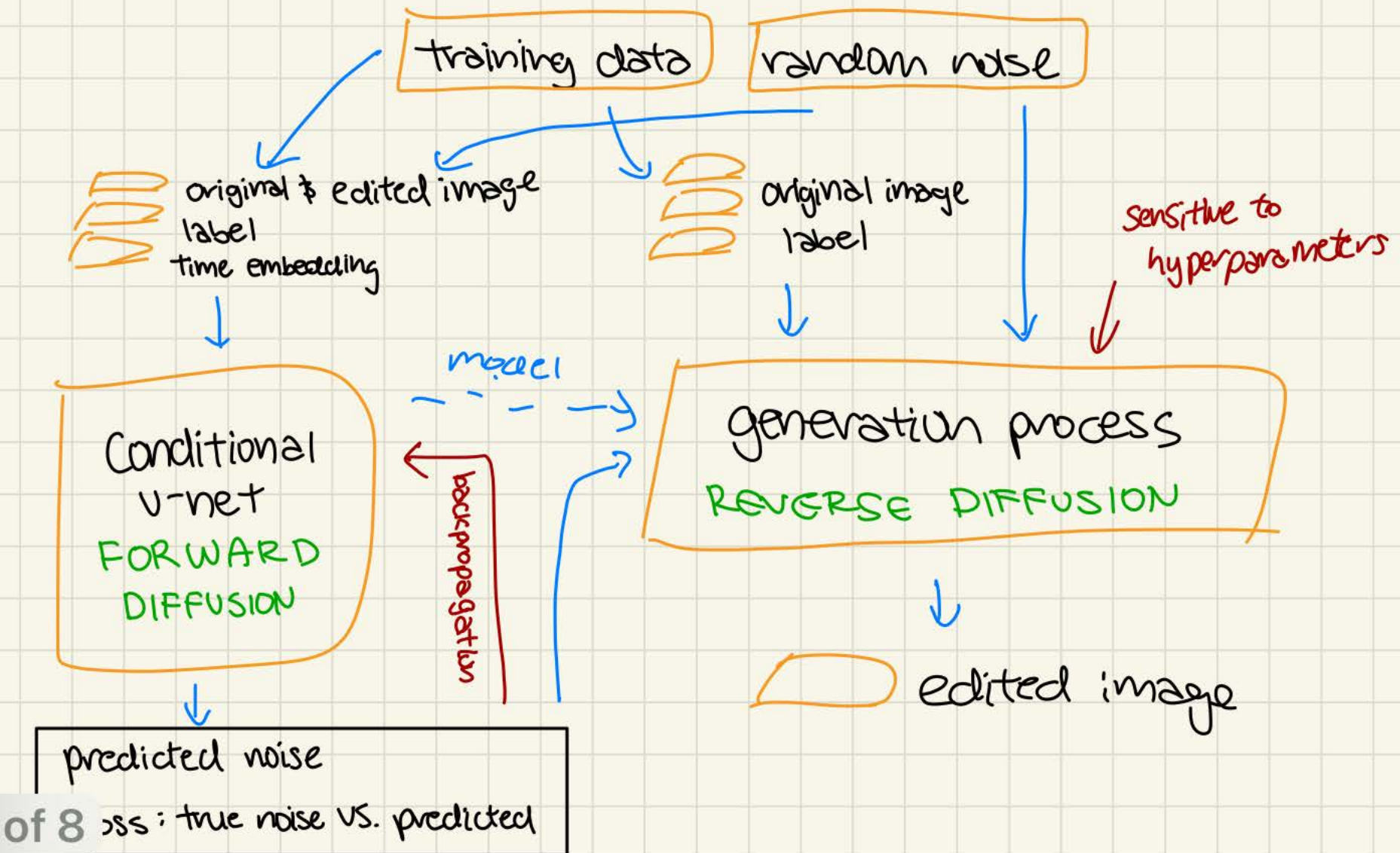
Forward Process - gradually adding noise to the edited image until it becomes completely random noise

Reverse Diffusion - gradually removing the noise, conditioned on the original image and the desired style label, until the edited image is reconstructed

Conditional U-Net: (used during training)

- inputs: noisy-edited images, original images, style labels, & diffusion time
- predicts the noise component present at each step
- outputs: model's predicted noise, estimate of the denoised edited image at current timestep
- **RANDOM NOISE IS ADDED TO EDITED IMAGE FIRST**

- generating images from pure noise (random noise)
- iteratively denoises using model's predictions
- conditioned on original image and style



CONDITIONAL WGAN

CRITIC

Critic output: SCALAR

- does not output a probability like a discriminator
- scalar - how "real" the input is according to the critic's learned function
- high score = sample more real low score = more fake

- scalar used to compute Wasserstein loss
- average critic scores for generated (fake) images & real images
- generator tries to minimize Wasserstein distance

inputs: unedited image, label, edited (real or fake)

- both real and fake are shown, but generator is only updated based on the critic's evaluation of its own fake outputs

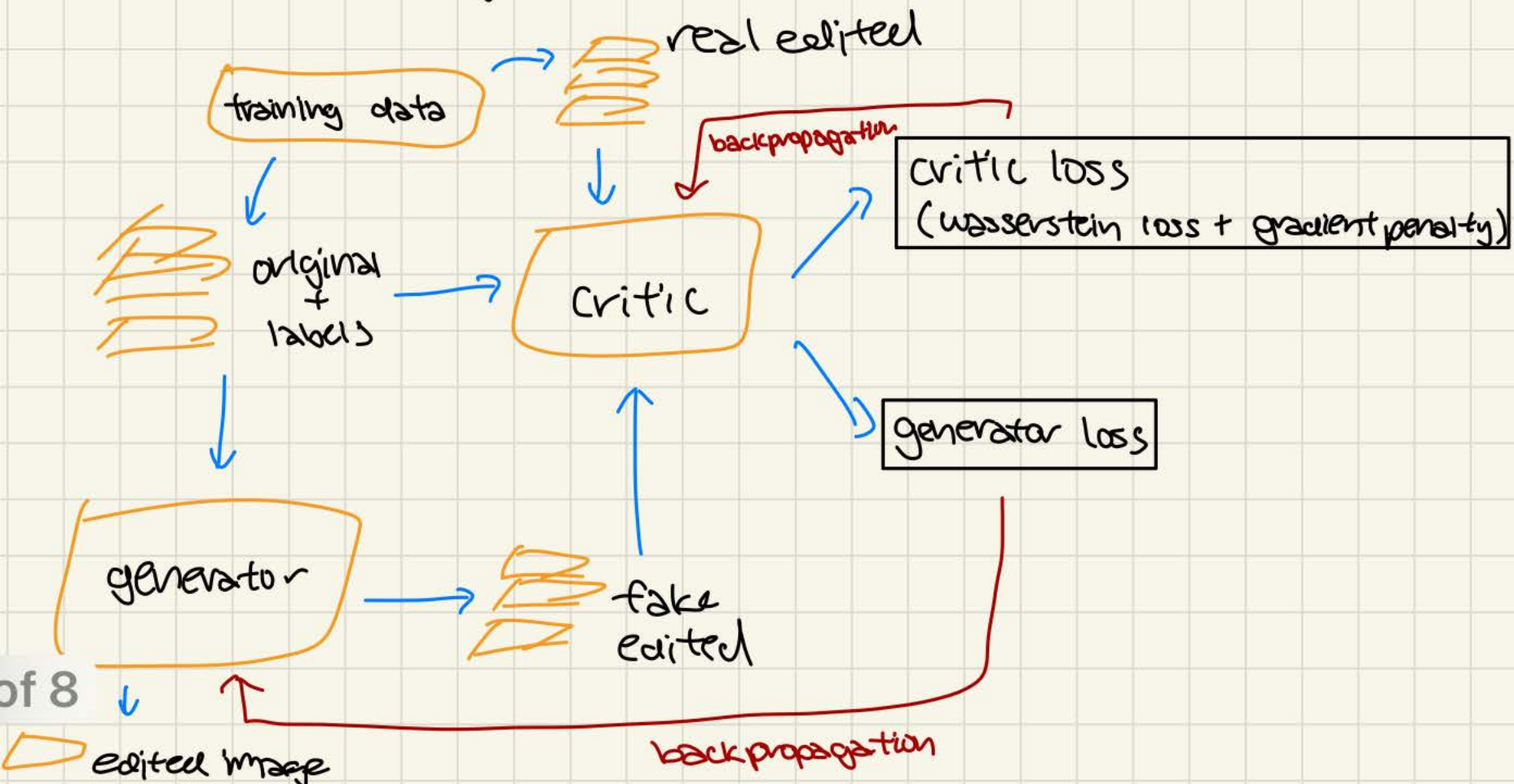
GENERATOR

difference from the usual generator:

- uses a U-net architecture instead of taking noise and upsampling from a latent space

U-net: encoder (downsamples), decoder (generates edited image)

tanh activation - $[-1, 1]$ range



DUMP/SEMI-ORGANIZATION

Before the actual color grading:

- trying 2 different models, see which one is "better"

"how will we analyze the results to conclude which one is better?"

Conditional Wasserstein GAN and Conditional Diffusion Model

- 9 different edits

"VISUAL ANALYSIS"

- NEED TO FIGURE OUT HOW TO ANALYZE GAN & DIFFUSION IN THE INITIAL COLOR CORRECTIONS

Image evaluation:

Peak Signal-to-Noise Ratio (PSNR) - similarity between AI & human

Structural Similarity Index (SSIM) - overall edge structure

- how will we get different color graded images?

- on the doc we said "professional color-graded images", probably will not be applicable if a python library is used

- worst case just stick with the color corrected images