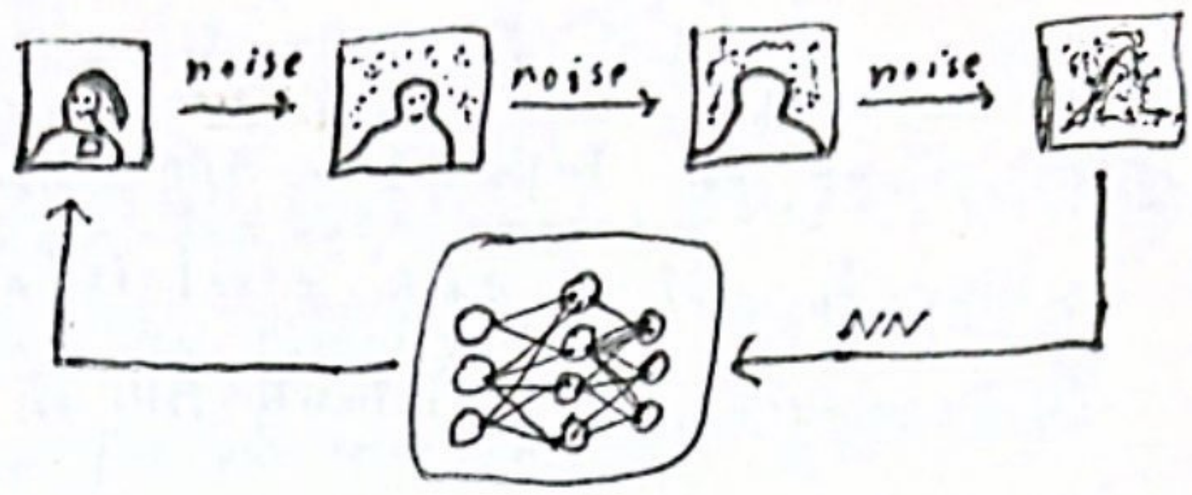
Diffusion Models & DDPM (Jonathan notes)

- in 2020 a paper called Denoising Diffusion Probabilistic models (DDPM) the paper showed for the first time it was possible to generate high quality imgs using a diffusion process where pure noise is transformed step by step into realistic imgs

- the core idea is to take training imgs, add noise step by step until the img is completely destroyed, then train a NN to reverse this process one step



- Algorithm: from DDPM a approach of Diffusion

Total added noise (model learns to predict) this

Algorithm 1 Training

1: repeat
2: $x_0 \sim q(x_0)$
3: $t \sim \text{uniform}(\{1, \dots, T\})$
4: $\epsilon \sim \mathcal{N}(0, I)$
5: take gradient step on
$\nabla_\theta \| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) \|^2$
(model output)
6: until converged

Algorithm 2 Sampling

1: $x_T \sim \mathcal{N}(0, I)$
2: for $t = T, \dots, 1$ do $z = 0$
3: $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $z = 0$
4: $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t z$
(x_{t-1}: output img; scaled model output; $\sigma_t z$: Random noise img)
5: end for
6: return x_0

Clean img

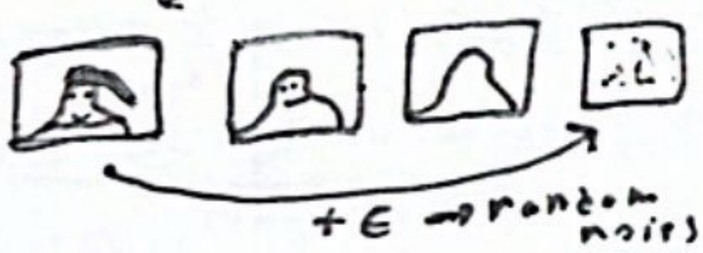
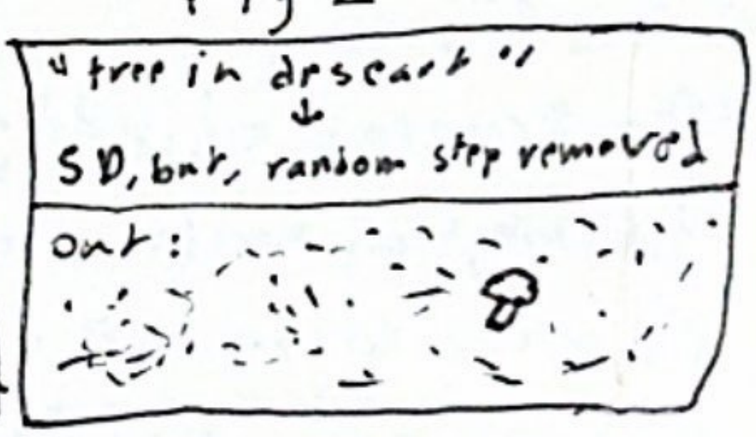


Fig 1

Fig 2



NOTE: Stable Diffusion is a popular Diffusion model and uses DDPM

- a Diffusion model Dose not work well if we have the models NN predict the prev img and go step by step rather predict the total noise added, to get back to original img - Also random noise is added Not just in training but in img generation as well. Algo 2 tells us when generating new imgs at each step after NN predicts a less noisy img we need to add random noise to this img before passing it back into our model for new output. This added noise matters for ex Fig 1 is stable diffusion (normal) and Fig 2 is stable diffusion but we dont add any random noise when generating new img. So why dont we go from noise to slightly less noisy img and why do we add noise in generating img? a simple mathematically equivalent way of understanding why DDPM works