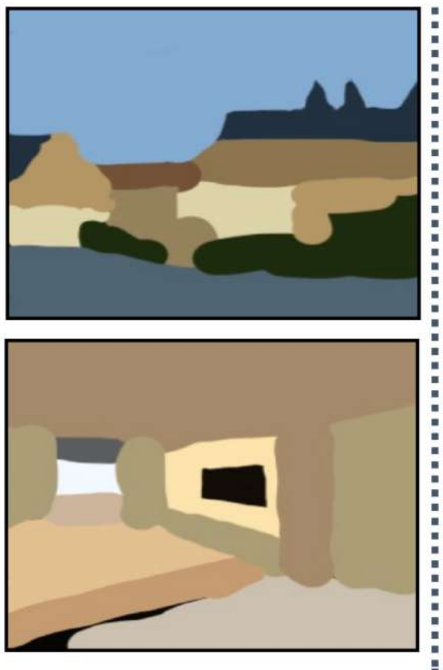
SDEdit
Guided image synthesis and editing with stochastic differential equations

C Meng, Y He, Y Song, J Song, J Wu, JY Zhu, S Ermon
Presented by: PD Alfano

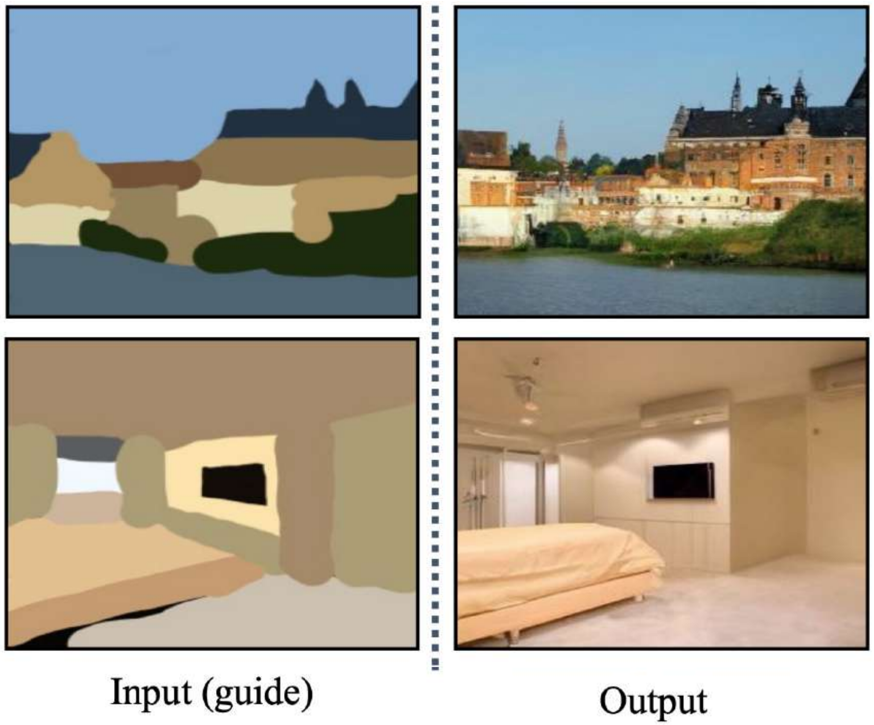
Do you believe in magic?

Stroke Painting to Image


Input (guide)

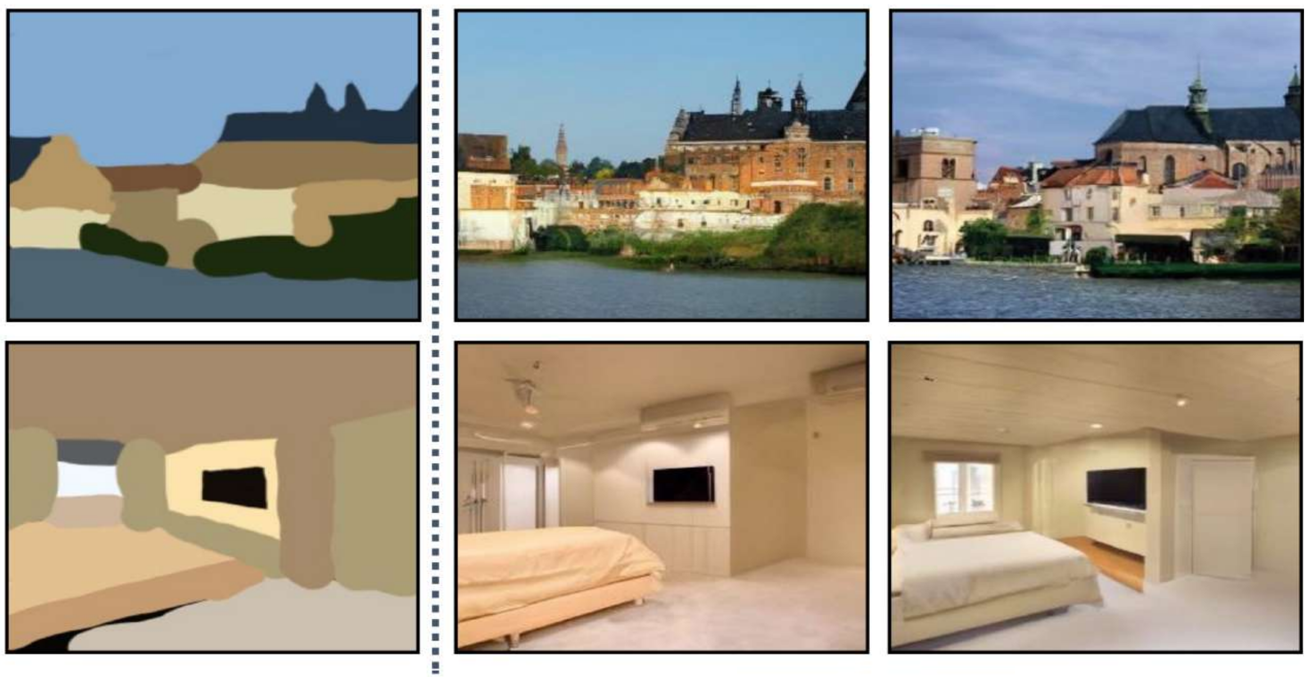
Do you believe in magic?

Stroke Painting to Image



Do you believe in magic?

Stroke Painting to Image



Input (guide) Output

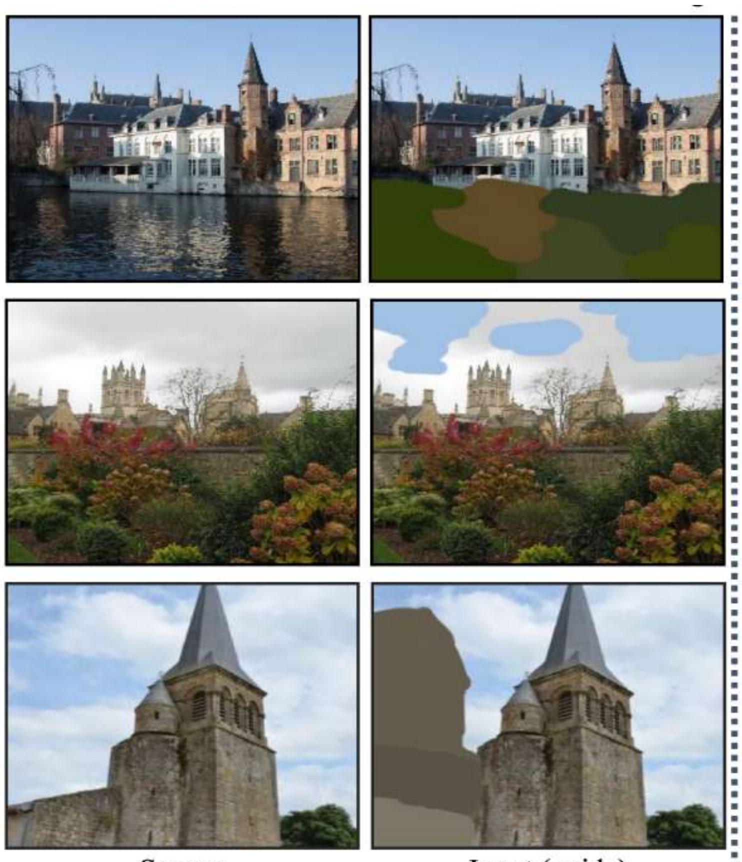
Do you believe in magic?

Stroke-based Editing

Source

Do you believe in magic?

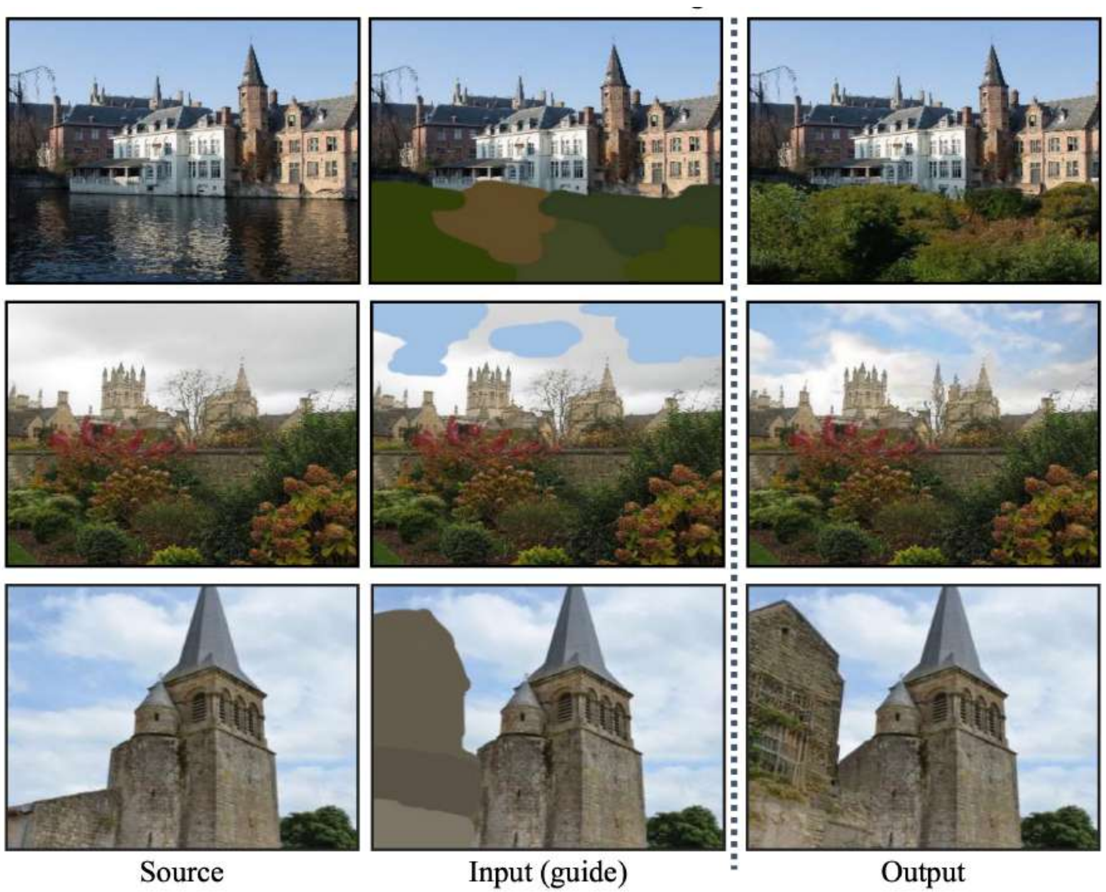
Stroke-based Editing



Source Input (guide)

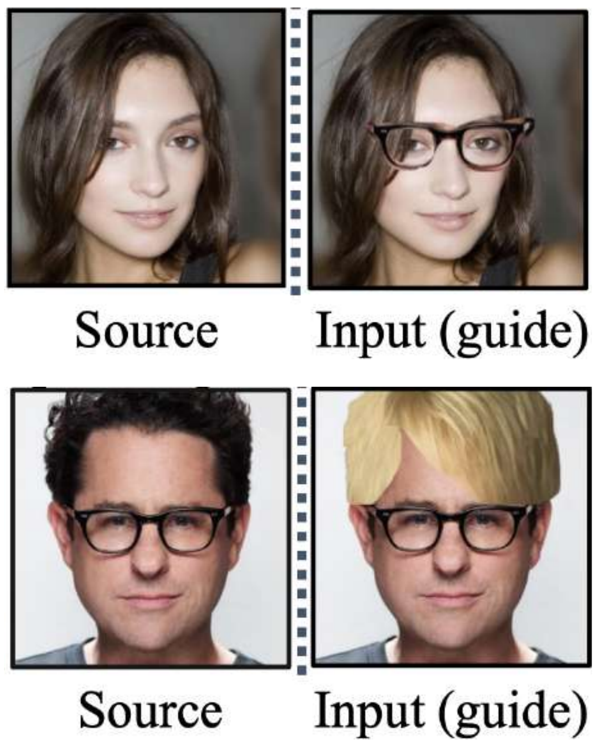
Do you believe in magic?

Stroke-based Editing



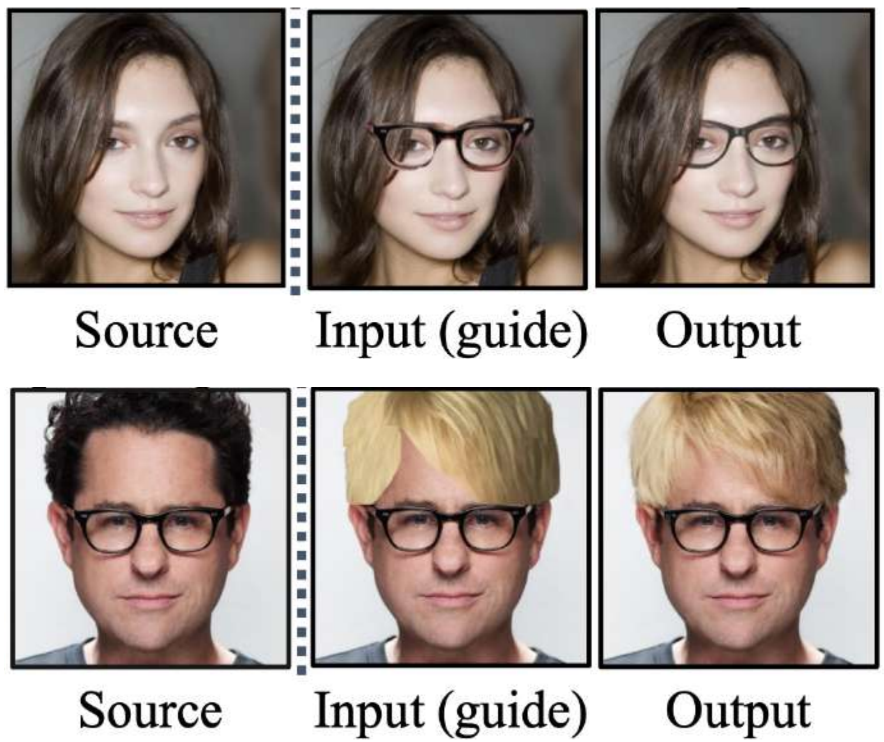
Do you believe in magic?

Image Compositing



Do you believe in magic?

Image Compositing



UniGe | MaLGa

Naïve problems

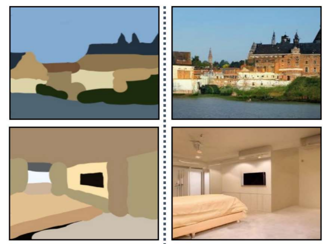
1. Input size?

    3 tasks:

    1 training

    Answer: next slide

6 chs



Naïve problems

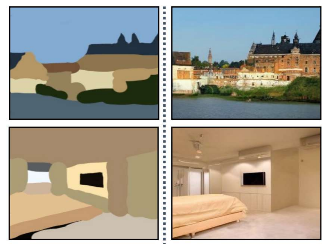
1. Input size?

    3 tasks:

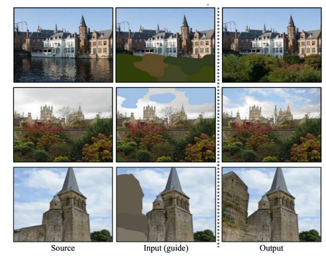
    1 training

    Answer: next slide

6 chs



9 chs



Naïve problems

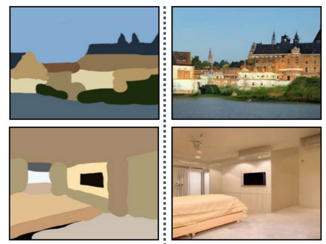
1. Input size?

    3 tasks:

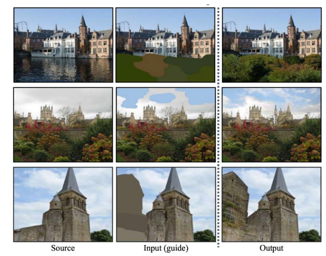
    1 training

    Answer: next slide

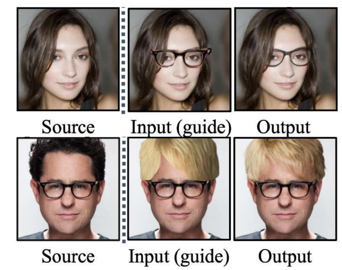
6 chs



9 chs

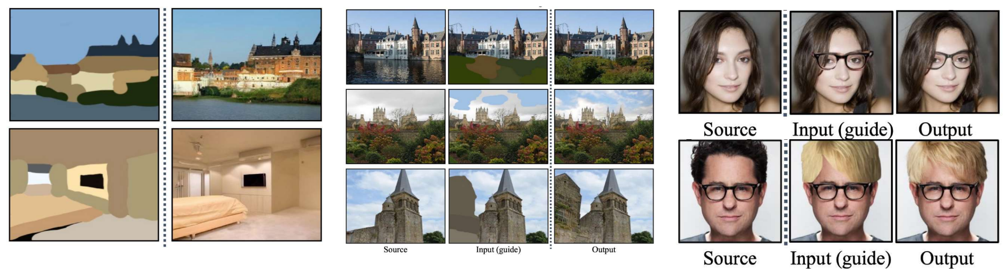


10 chs



Naïve problems

6 chs | 9 chs | 10 chs

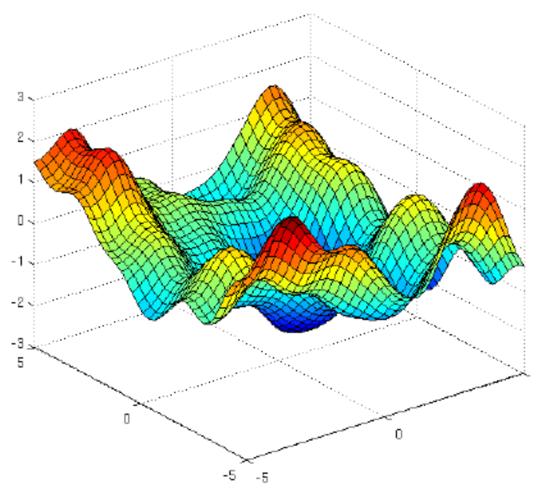


1. Input size?

   3 tasks:

   1 training

   Answer: next slide

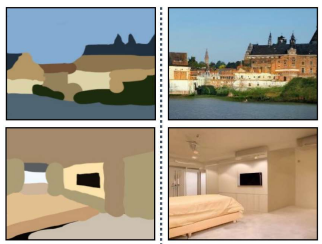


2. Robustness?

   Answer: next sect.

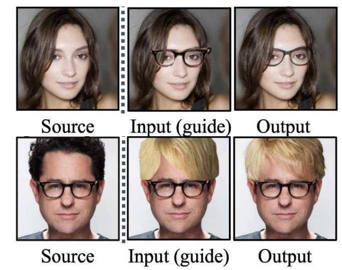
Input size

6 chs



9 chs

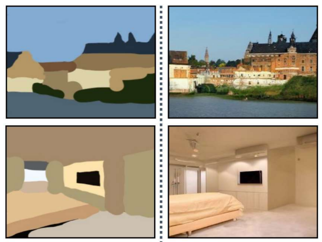
10 chs



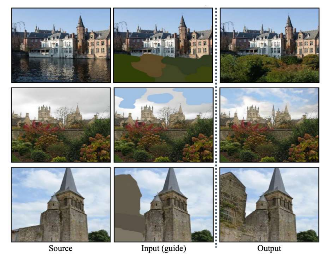
Super-imposition

Input size

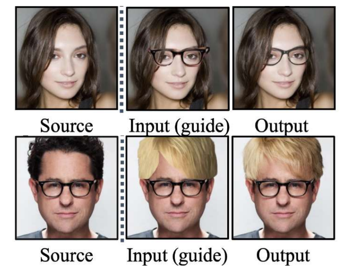
6 chs



9 chs



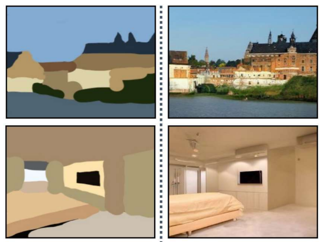
10 chs



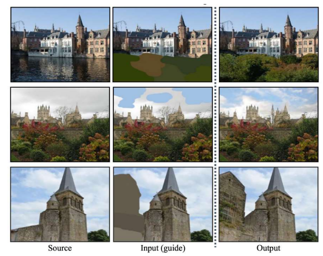
Super-imposition

Input size

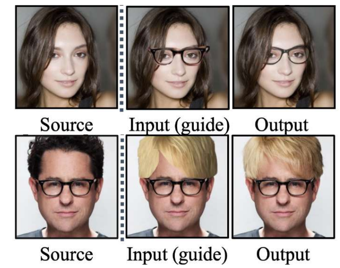
6 chs



9 chs



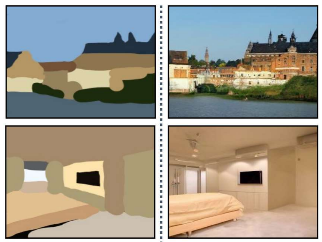
10 chs



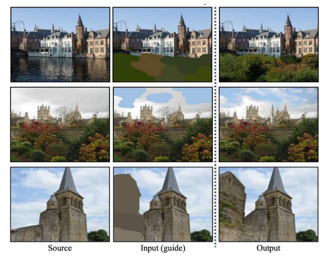
Super-imposition

Input size

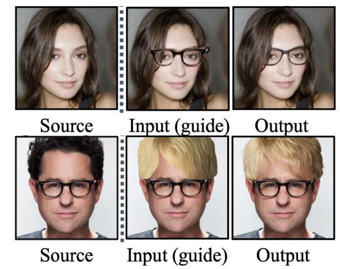
6 chs



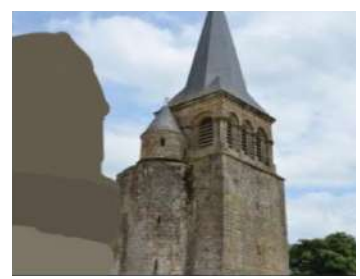
9 chs



10 chs

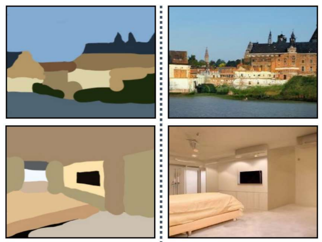


Super-imposition

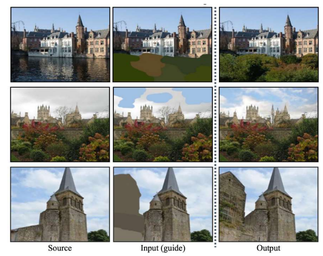


Input size

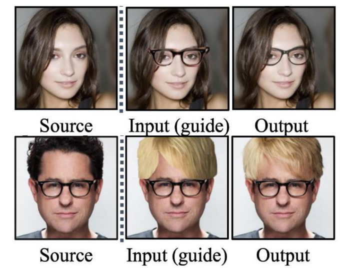
6 chs



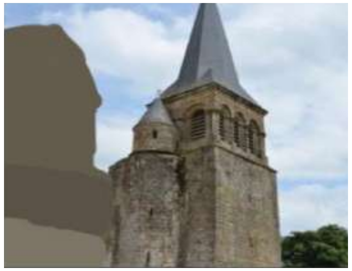
9 chs



10 chs

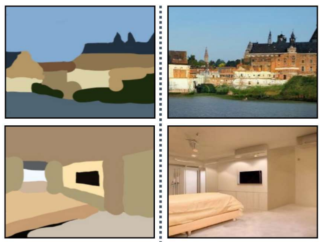


Super-imposition

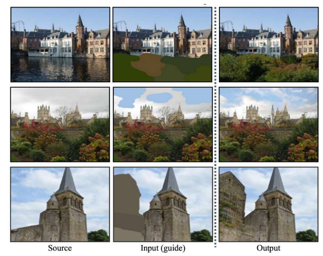


Input size

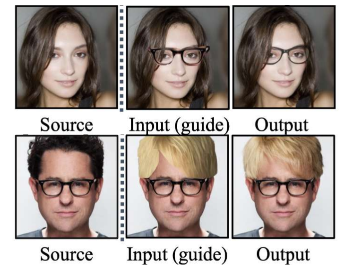
6 chs



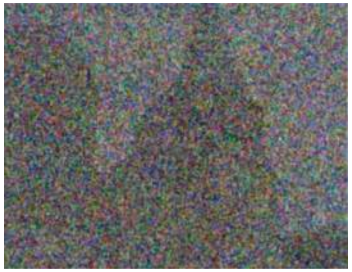
9 chs



10 chs



Super-imposition

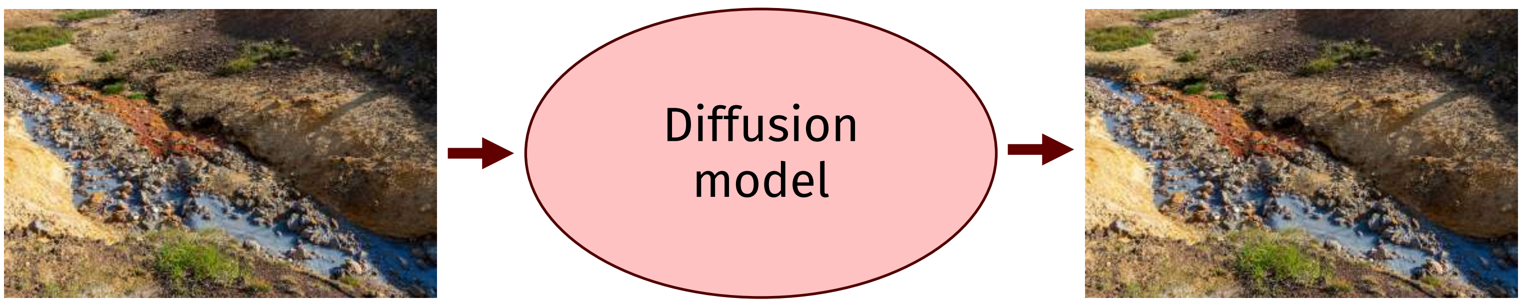


Intuition

Just the usual diffusion model!

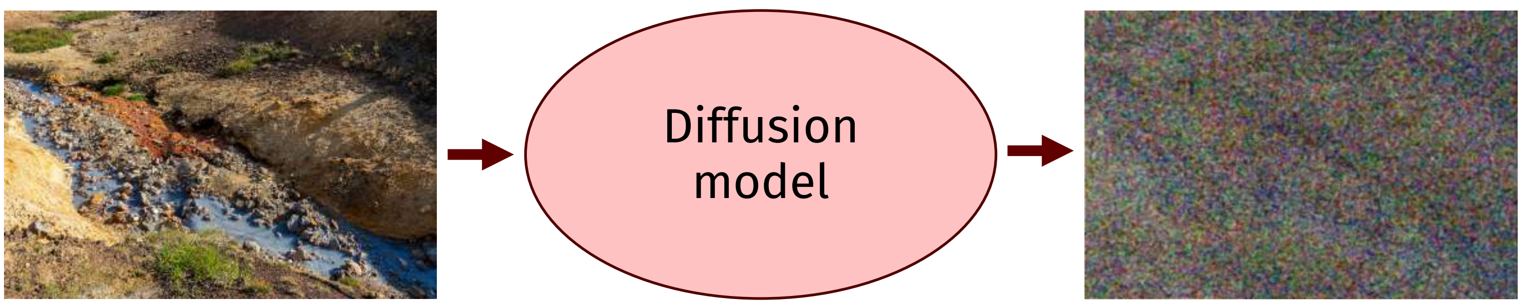
Intuition

Just the usual diffusion model!



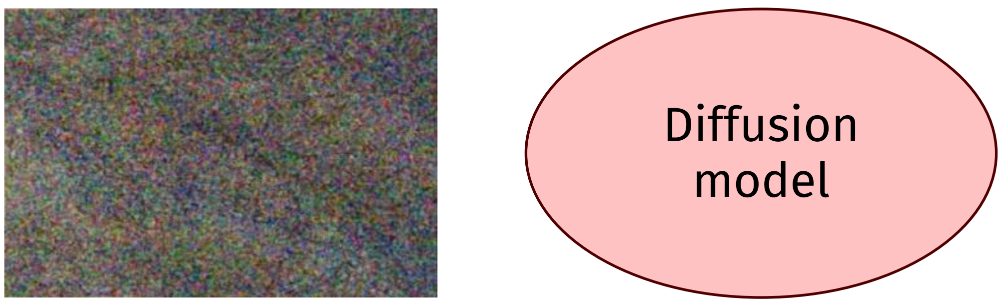
Intuition

Just the usual diffusion model!



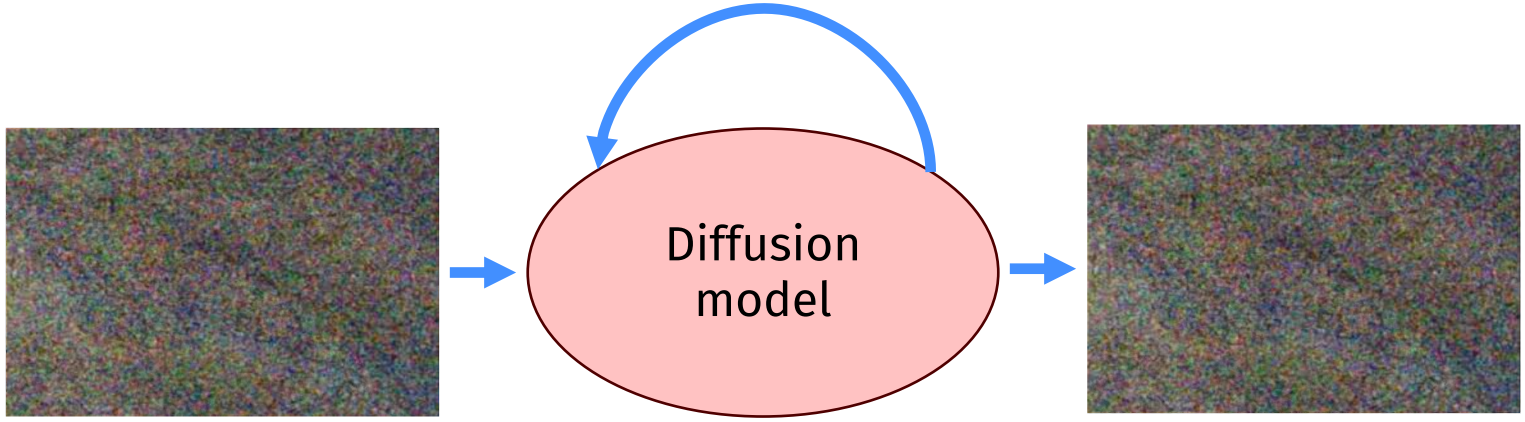
Intuition

Just the usual diffusion model!



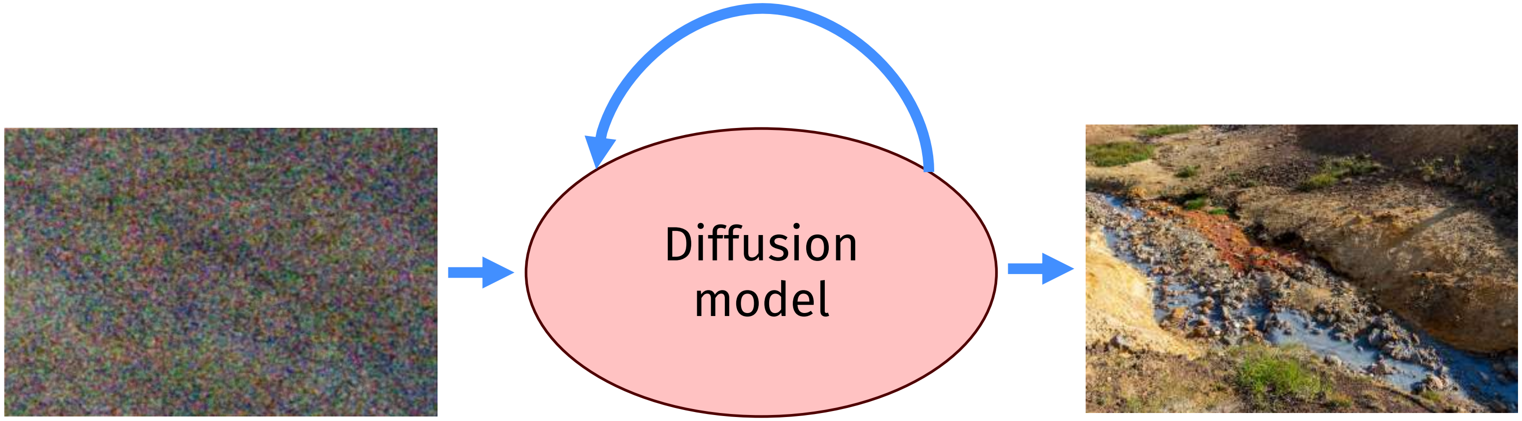
Intuition

Just the usual diffusion model!

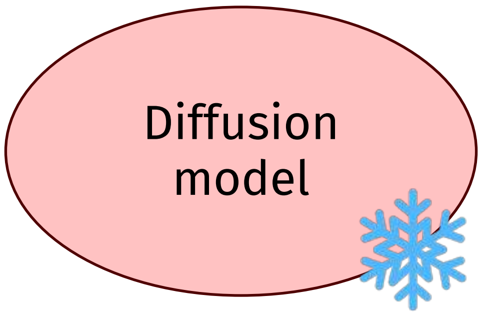


Intuition

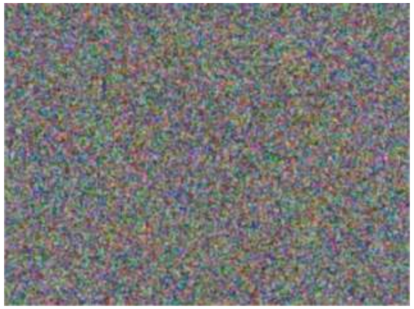
Just the usual diffusion model!

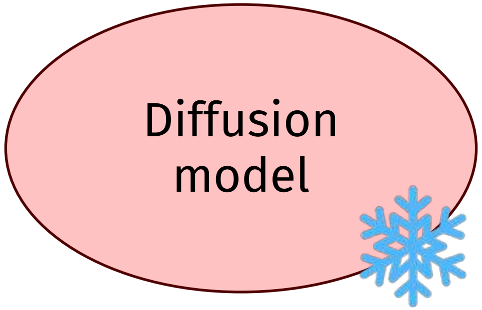


Hijack your diffusion model

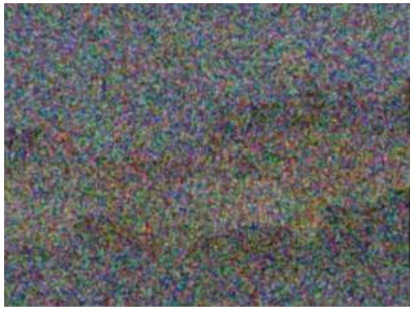


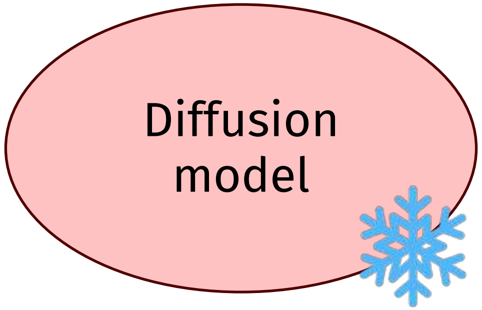
Hijack your diffusion model



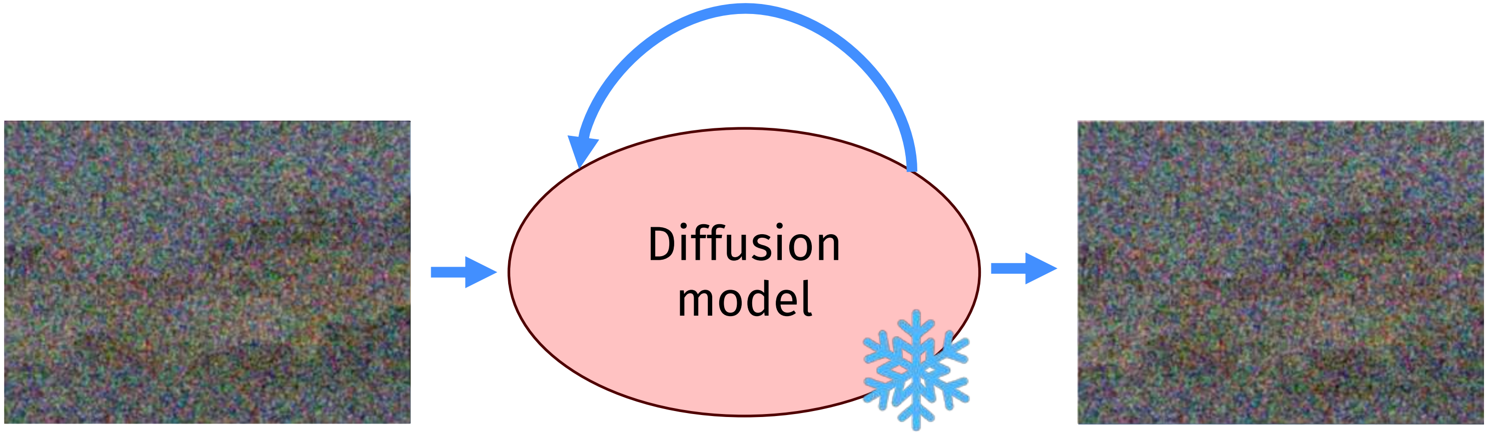


Hijack your diffusion model

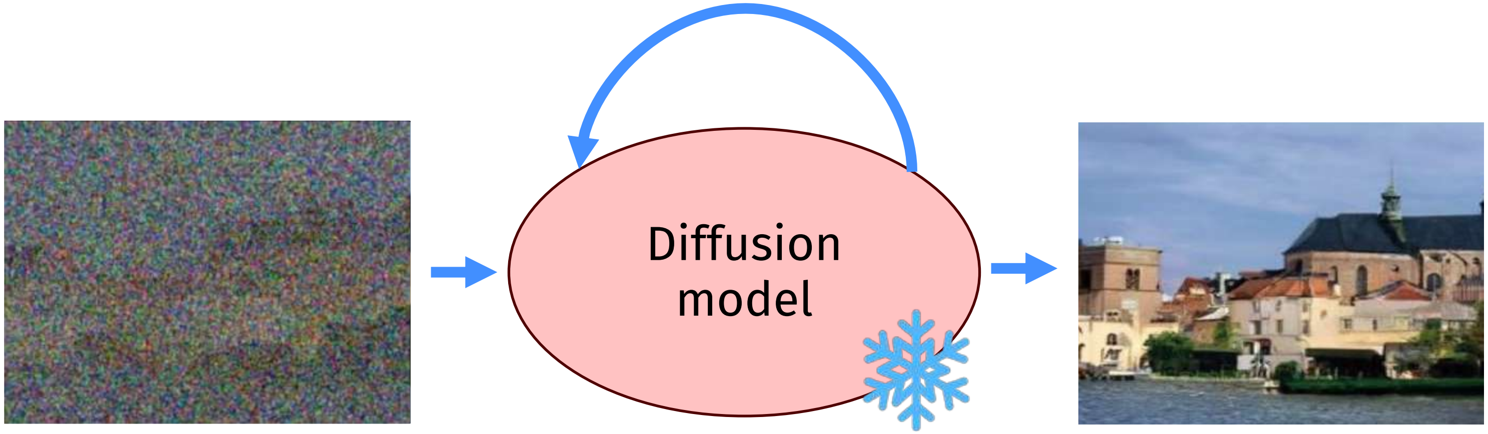




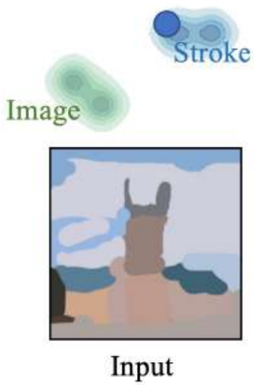
Hijack your diffusion model



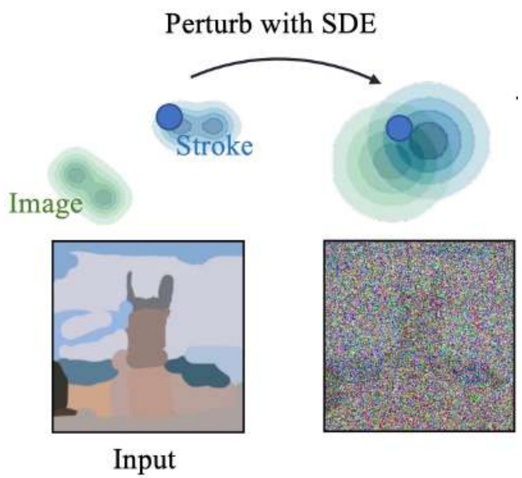
Hijack your diffusion model



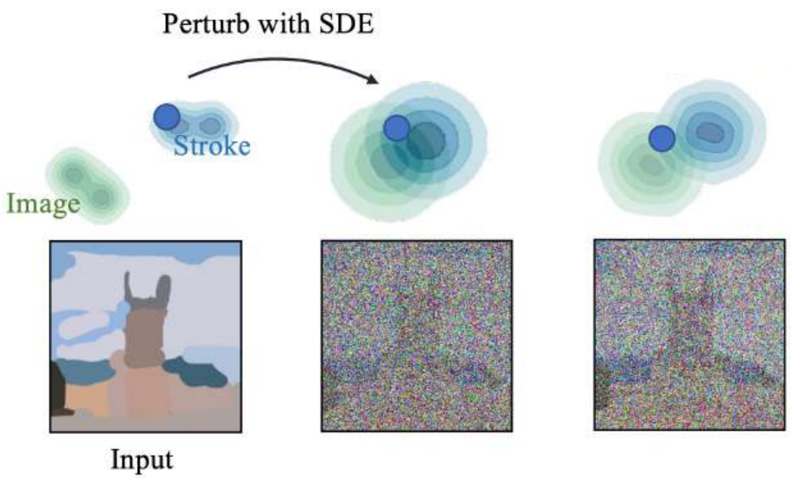
Perturbed distributions



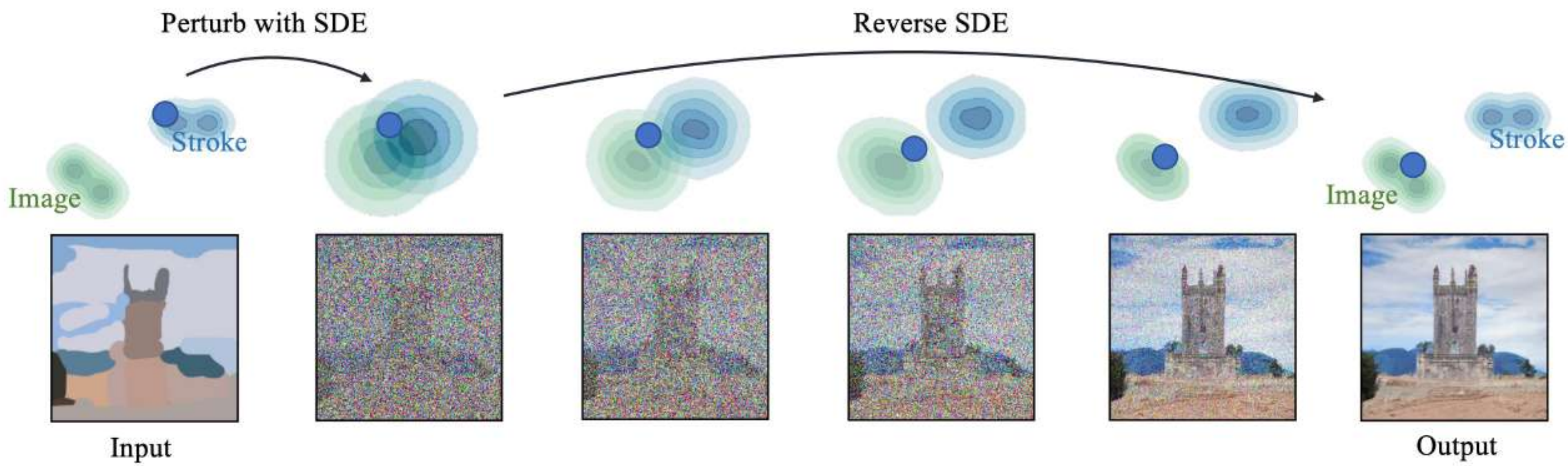
Perturbed distributions



Perturbed distributions

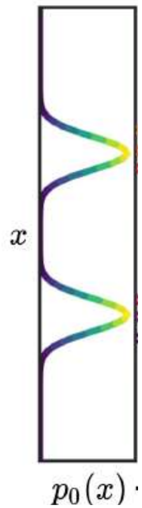


Perturbed distributions



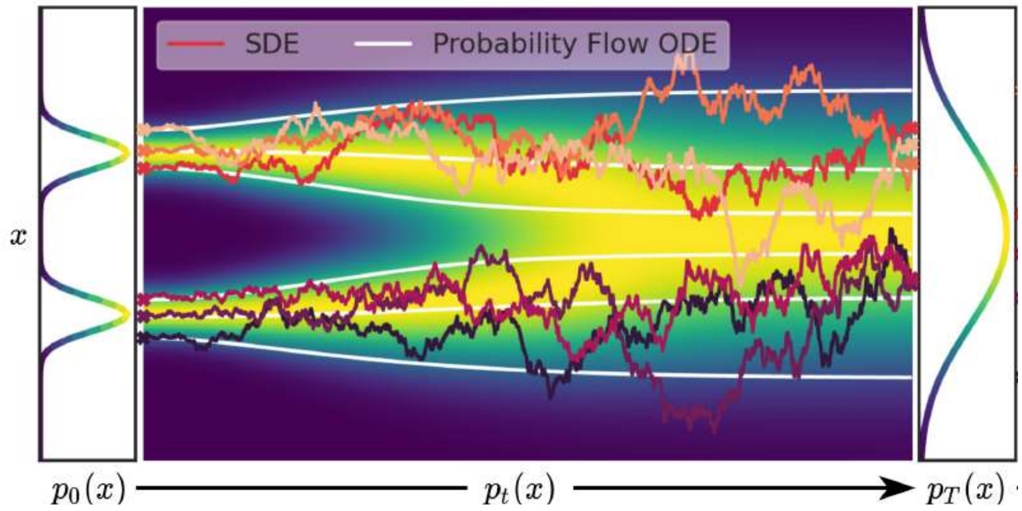
Back in the days..

(Actually is 2021)



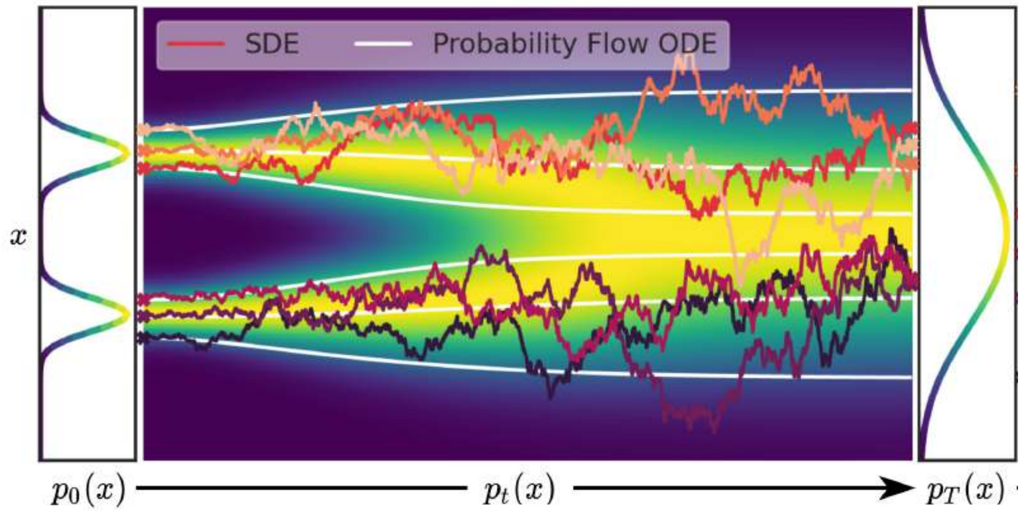
Back in the days..

(Actually is 2021)



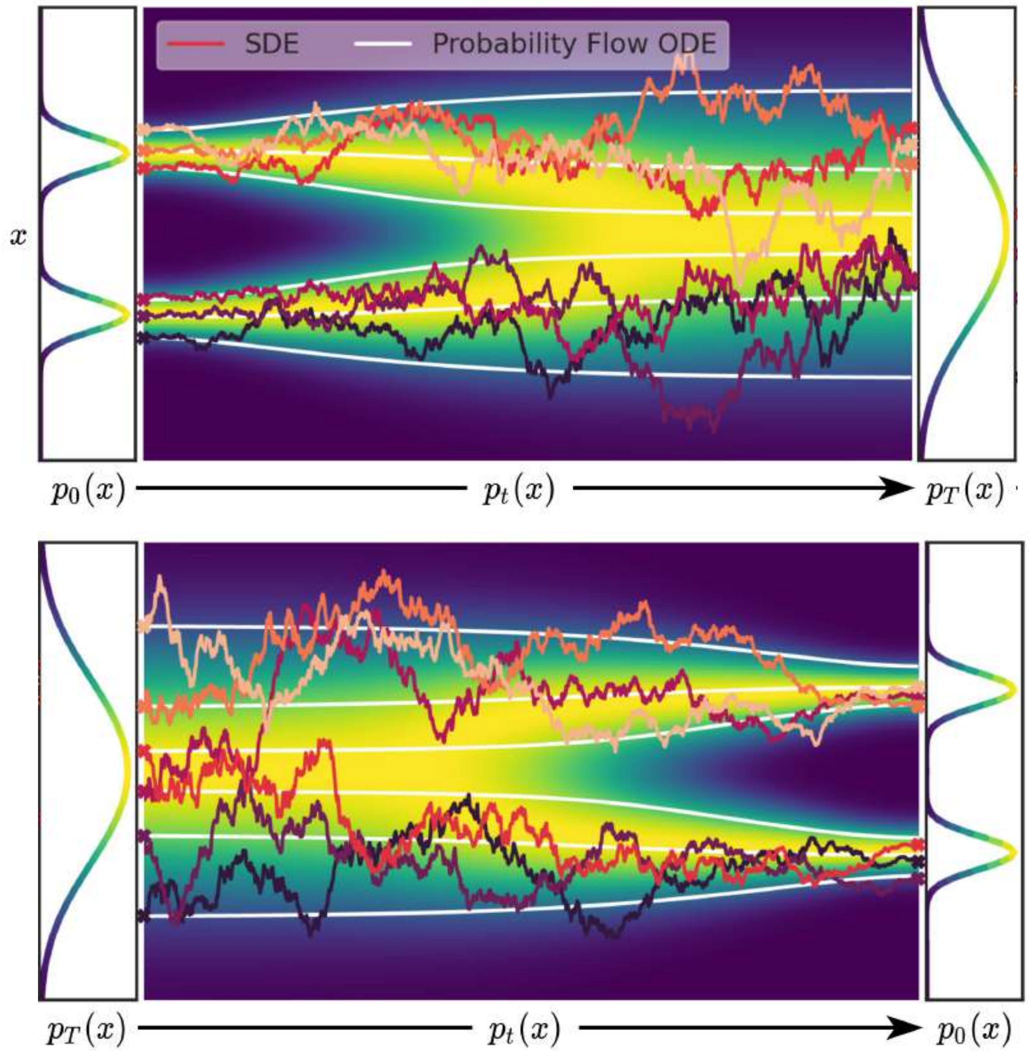
Back in the days..

(Actually is 2021)



Back in the days..

(Actually is 2021)



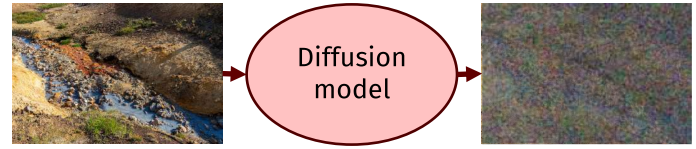
Problems & Intuition

Training

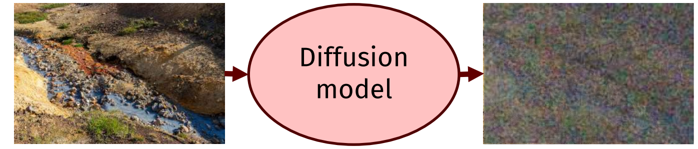
Experiments

Conclusions

Forward process



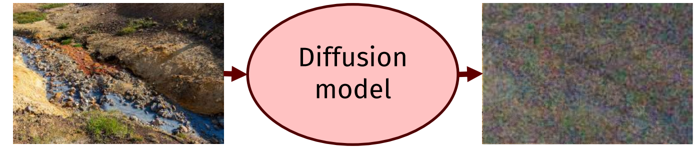
Forward process



Similar to original diffusion. Given:

$\mathbf{x}(t) \in \mathbb{R}^d \qquad t \in [0, 1] \qquad \mathbf{x}(0) \sim p_0 = p_{\text{data}}$

Forward process



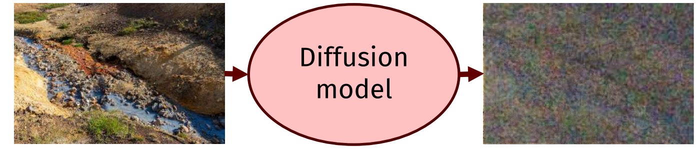
Similar to original diffusion. Given:

$\mathbf{x}(t) \in \mathbb{R}^d$ $\quad t \in [0, 1]$ $\quad \mathbf{x}(0) \sim p_0 = p_{\text{data}}$

$$\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \sigma(t)\mathbf{z},$$

where: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}),$

Forward process



Similar to original diffusion. Given:

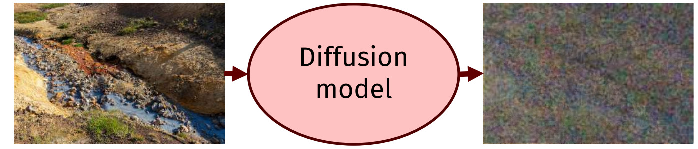
$\mathbf{x}(t) \in \mathbb{R}^d$ $\qquad t \in [0, 1]$ $\qquad \mathbf{x}(0) \sim p_0 = p_{\text{data}}$

$$\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \sigma(t)\mathbf{z},$$

Weighted data

where: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}),$

Forward process



Similar to original diffusion. Given:

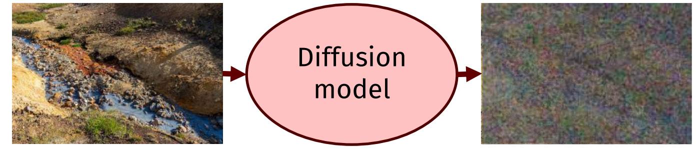
$\mathbf{x}(t) \in \mathbb{R}^d$ $\quad t \in [0, 1]$ $\quad \mathbf{x}(0) \sim p_0 = p_{\text{data}}$

$$\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \sigma(t)\mathbf{z},$$

Weighted data Weighted noise

where: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}),$

Forward process



Similar to original diffusion. Given:

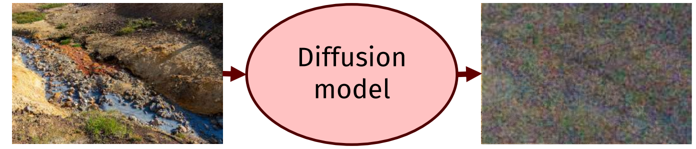
$\mathbf{x}(t) \in \mathbb{R}^d \qquad t \in [0,1] \qquad \mathbf{x}(0) \sim p_0 = p_{\text{data}}$

$$\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \sigma(t)\mathbf{z},$$

Weighted data Weighted noise

where: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \qquad \alpha(t) : [0,1] \to [0,1] \qquad \sigma(t) : [0,1] \to [0,\infty)$

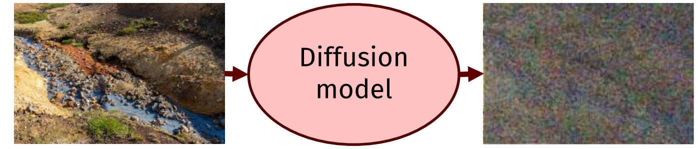
Weight set



$$\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \sigma(t)\mathbf{z},$$

$\alpha(t) : [0,1] \rightarrow [0,1]$ $\sigma(t) : [0,1] \rightarrow [0,\infty)$

Weight set



$$\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \sigma(t)\mathbf{z},$$

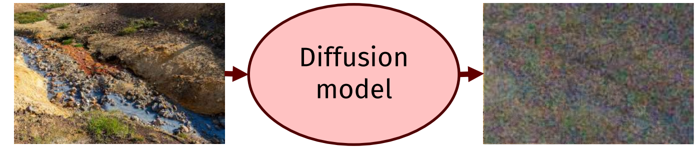
$$\alpha(t) : [0,1] \to [0,1] \qquad \sigma(t) : [0,1] \to [0,\infty)$$

Two usual approaches:

- Variance Exploding SDE (VE-SDE)

For all t, $\alpha(t) = 1$

Weight set



$$\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \sigma(t)\mathbf{z},$$

$\alpha(t) : [0,1] \to [0,1]$ $\quad$ $\sigma(t) : [0,1] \to [0,\infty)$

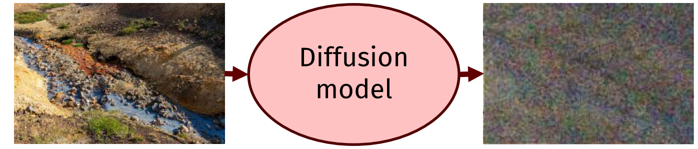
Two usual approaches:

- Variance Exploding SDE (VE-SDE)

For all t, $\alpha(t) = 1$

Large $\sigma(1)$ so $\mathcal{N}(\mathbf{0}, \sigma^{\mathbf{2}}(\mathbf{1})\mathbf{I})$ gaussian noise

Weight set



$$\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \sigma(t)\mathbf{z},$$

$\alpha(t) : [0, 1] \to [0, 1]$ $\sigma(t) : [0, 1] \to [0, \infty)$

Two usual approaches:

- Variance Exploding SDE (VE-SDE)

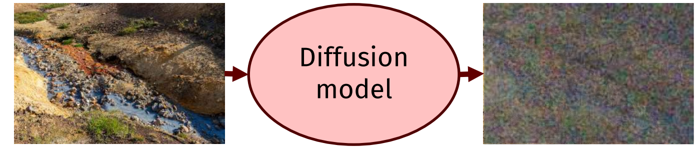
  For all t, $\alpha(t) = 1$

  Large $\sigma(1)$ so $\mathcal{N}(\mathbf{0}, \sigma^{\mathbf{2}}(\mathbf{1})\mathbf{I})$ gaussian noise

- Variance Preserving SDE (VP-SDE)

  $$\alpha^2(t) + \sigma^2(t) = 1$$

Weight set



$$\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \sigma(t)\mathbf{z},$$

$$\alpha(t) : [0,1] \to [0,1] \qquad \sigma(t) : [0,1] \to [0,\infty)$$

Two usual approaches:

- Variance Exploding SDE (VE-SDE)

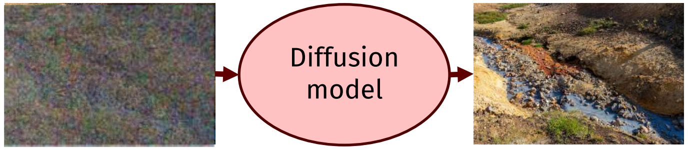
For all t, $\alpha(t) = 1$

Large $\sigma(1)$ so $\mathcal{N}(\mathbf{0}, \sigma^{\mathbf{2}}(\mathbf{1})\mathbf{I})$ gaussian noise

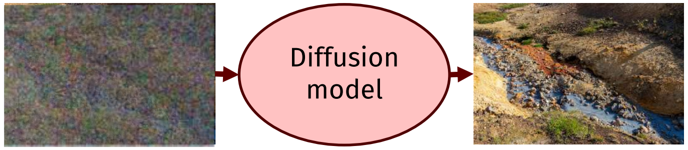
- Variance Preserving SDE (VP-SDE)

$\alpha^2(t) + \sigma^2(t) = 1$ $\alpha(t) \to 0$ as $t \to 1$

Backward process



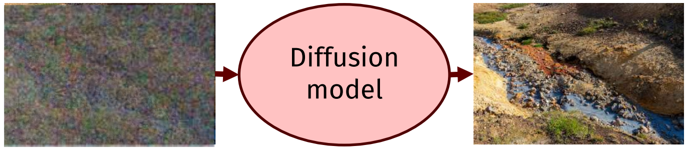
Backward process



Disclaimer:

Score, a noise generalization

Backward process

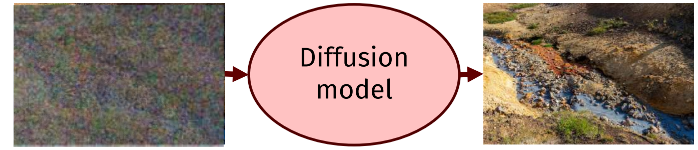


Disclaimer:

Score, a noise generalization

Score, can be reduced to noise

Backward process



Disclaimer:

Score, a noise generalization

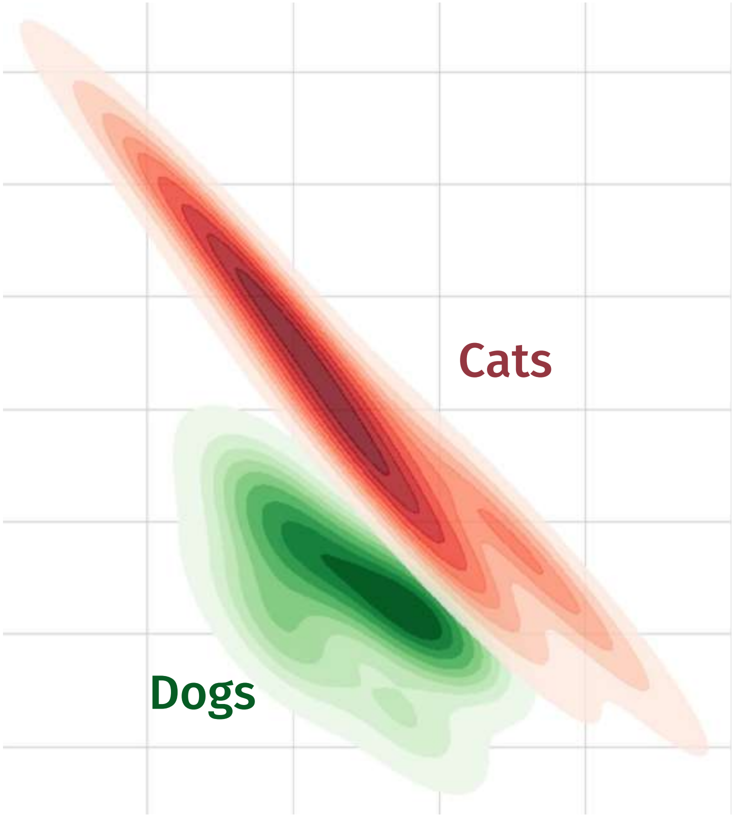
Score, can be reduced to noise

Do not predict noise, predict score!

Data distribution

Data distribution:

$$\log p_t(\mathbf{x})$$



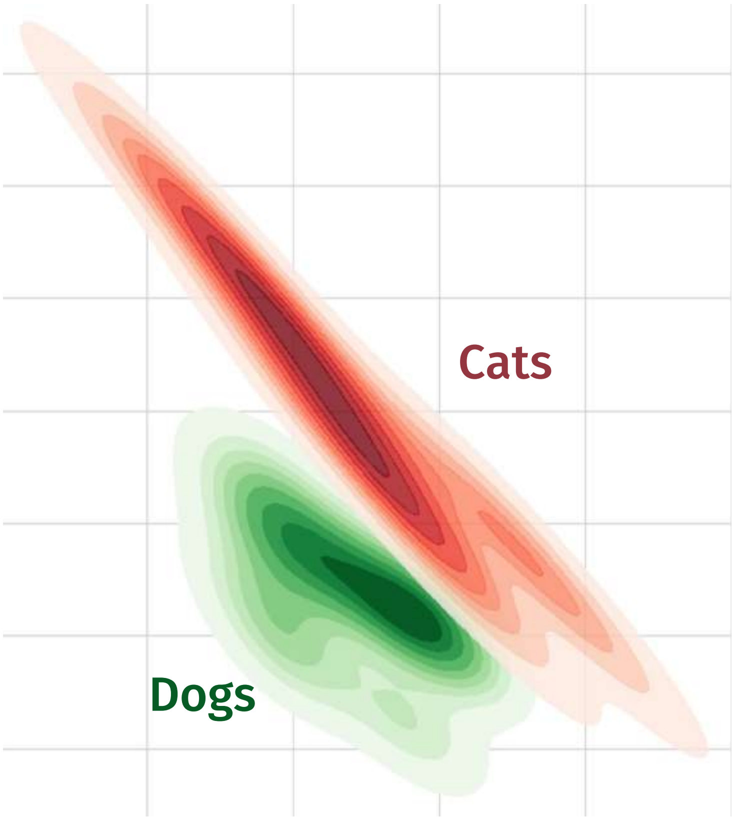
Data distribution

Data distribution:

$$\log p_t(\mathbf{x})$$

Gradient:

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$$



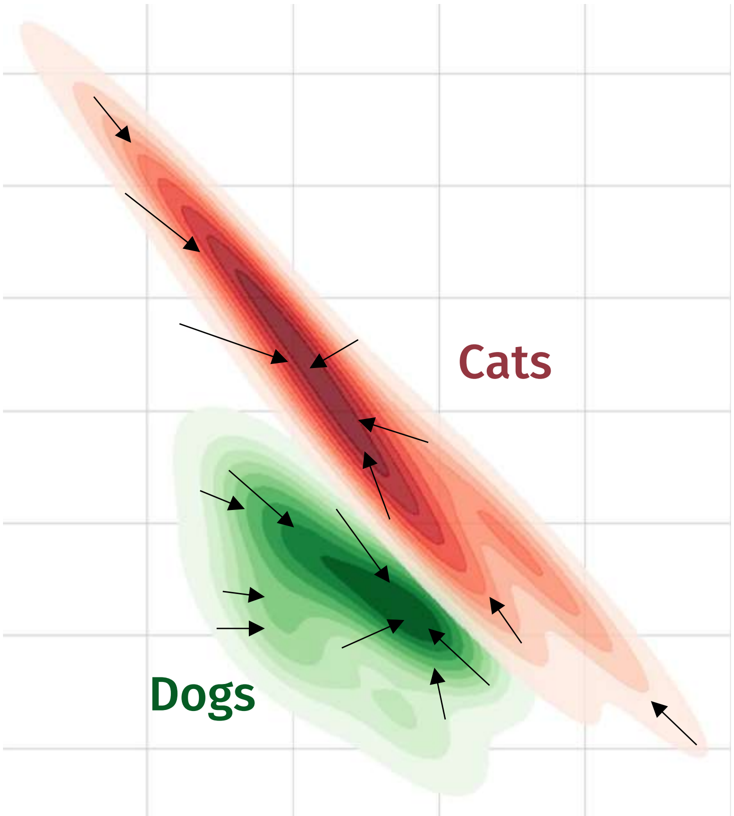
Data distribution

Data distribution:

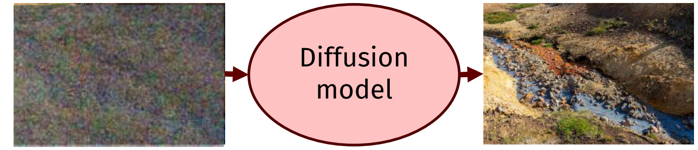
$$\log p_t(\mathbf{x})$$

Gradient:

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$$



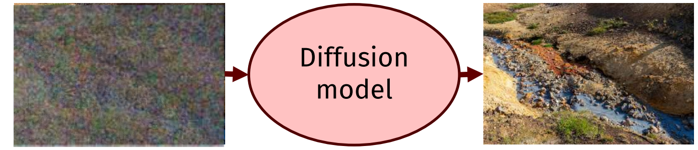
Backward process



Approximate gradient $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ with score:

$$\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)$$

Backward process



Approximate gradient $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ with score:

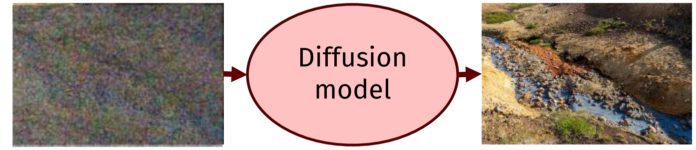
$$\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)$$

Loss:

$$L_t = \mathbb{E}_{\mathbf{x}(0)\sim p_{\text{data}}, \mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[||\sigma_t \boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t) - \mathbf{z}||_2^2]$$

score modeled on the noise..

Backward process

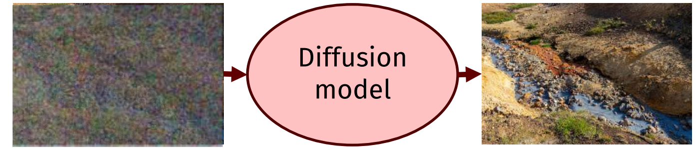


Given gradient approximation

SDE solution via Euler-Maruyama method:

$$x(t) = x(t+\Delta t) + (\sigma^2(t) - \sigma^2(t+\Delta t))\mathbf{s}_\theta(x(t), t) + \sqrt{\sigma^2(t) - \sigma^2(t+\Delta t)}\mathbf{z}'$$

Backward process

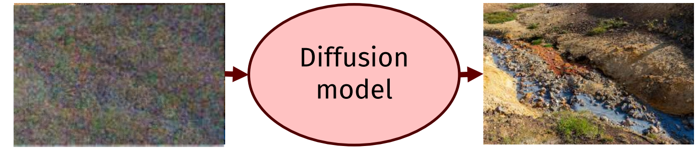


Given gradient approximation

SDE solution via Euler-Maruyama method:

$$x(t) = x(t + \Delta t) + (\sigma^2(t) - \sigma^2(t + \Delta t))\mathbf{s}_\theta(x(t), t) + \sqrt{\sigma^2(t) - \sigma^2(t + \Delta t)}\mathbf{z}'$$

Backward process



Given gradient approximation

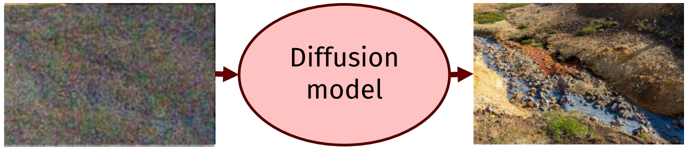
SDE solution via Euler-Maruyama method:

$$x(t) = x(t + \Delta t) + (\sigma^2(t) - \sigma^2(t + \Delta t))\mathbf{s}_\theta(x(t), t) + \sqrt{\sigma^2(t) - \sigma^2(t + \Delta t)}\mathbf{z}'$$

$$\varepsilon = \sqrt{\sigma^2(t) - \sigma^2(t + \Delta t)}$$

Disclaimer! Always known quantity!

Backward process



Given gradient approximation

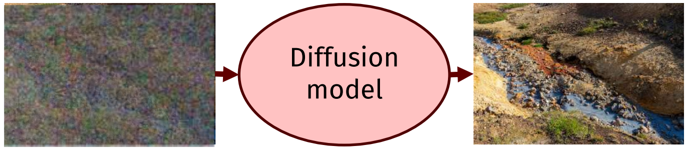
SDE solution via Euler-Maruyama method:

$$x(t) = x(t+\Delta t) + (\sigma^2(t) - \sigma^2(t+\Delta t))\mathbf{s}_\theta(x(t), t) + \sqrt{\sigma^2(t) - \sigma^2(t+\Delta t)}\mathbf{z}'$$

$$\varepsilon = \sqrt{\sigma^2(t) - \sigma^2(t+\Delta t)}$$

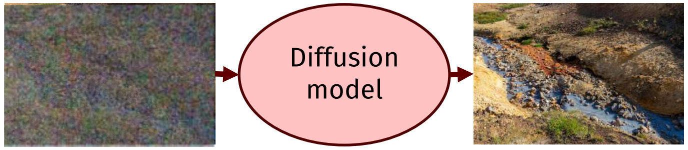
$$x(t) = x(t+\Delta t) + \varepsilon^2 \mathbf{s}_\theta(x(t), t) + \varepsilon \mathbf{z}'$$

Backward process



$$x(t) = x(t + \Delta t) + \varepsilon^2 \mathbf{s}_\theta(x(t), t) + \varepsilon \mathbf{z}'$$

Backward process

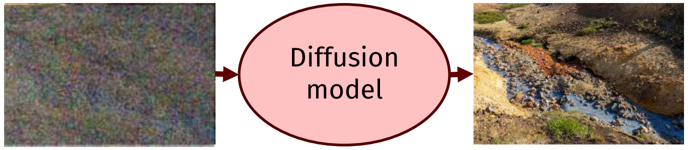


$$x(t) = x(t + \Delta t) + \varepsilon^2 \mathbf{s}_\theta(x(t), t) + \varepsilon \mathbf{z}'$$

Intuitively

Previous location

Backward process



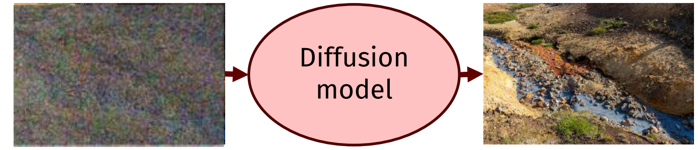
$$x(t) = x(t + \Delta t) + \varepsilon^2 \mathbf{s}_\theta(x(t), t) + \varepsilon \mathbf{z}'$$

Intuitively

Previous location

Score/gradient/noise direction

Backward process



$$x(t) = x(t + \Delta t) + \varepsilon^2 \mathbf{s}_\theta(x(t), t) + \varepsilon \mathbf{z}'$$

Intuitively

Previous location

Score/gradient/noise direction

Stochastic noise

Realism, faithfulness

Output:

-) Realistic

-) Faithfull to guide

Realism, faithfulness

Output:

-) Realistic

-) Faithfull to guide

Fun fact: not positively correlated

Realism, faithfulness

Output:

-) Realistic

-) Faithfull to guide

Fun fact: not positively correlated

	Not Faithful	Faithful
Not Realistic		
Realistic		

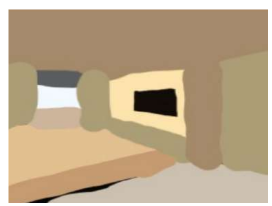
Realism, faithfulness

Output:

-) Realistic

-) Faithfull to guide

Fun fact: not positively correlated

	Not Faithful	Faithful
Not Realistic		
Realistic		

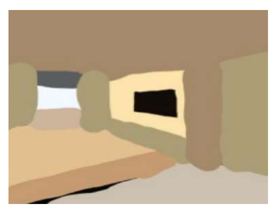
Realism, faithfulness

Output:

-) Realistic

-) Faithfull to guide

Fun fact: not positively correlated

	Not Faithful	Faithful
Not Realistic		
Realistic		

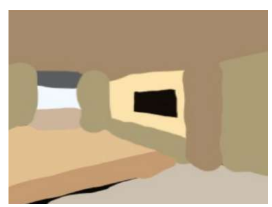
Realism, faithfulness

Output:

-) Realistic

-) Faithfull to guide

Fun fact: not positively correlated

	Not Faithful	Faithful
Not Realistic		
Realistic		

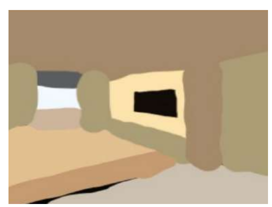
Realism, faithfulness

Output:

-) Realistic

-) Faithfull to guide

Fun fact: not positively correlated

	Not Faithful	Faithful
Not Realistic		
Realistic		

KID and L_2

Realism: Kernel Inception Distance (KID)

- Subsample images from same class
- Compute similarity

KID (

,

)

KID and L_2

Realism: Kernel Inception Distance (KID)

- Subsample images from same class
- Compute similarity

KID ( , )

Faithfulness: L_2

- Between guide and output

L_2 ( , )

t influence

$t \in [0, 1]$

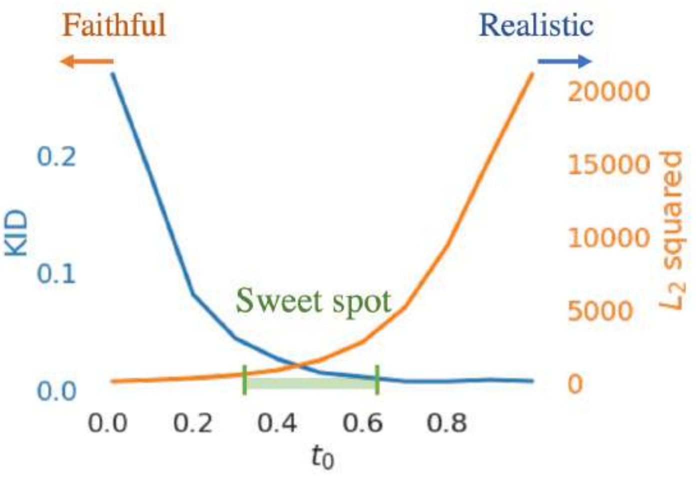
Set a proper t_0

t influence

$t \in [0, 1]$

Set a proper t_0

Compromise realism/faithfulness

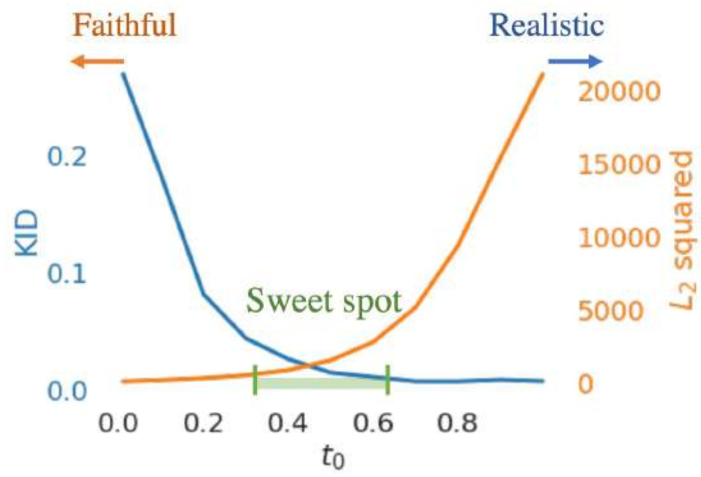


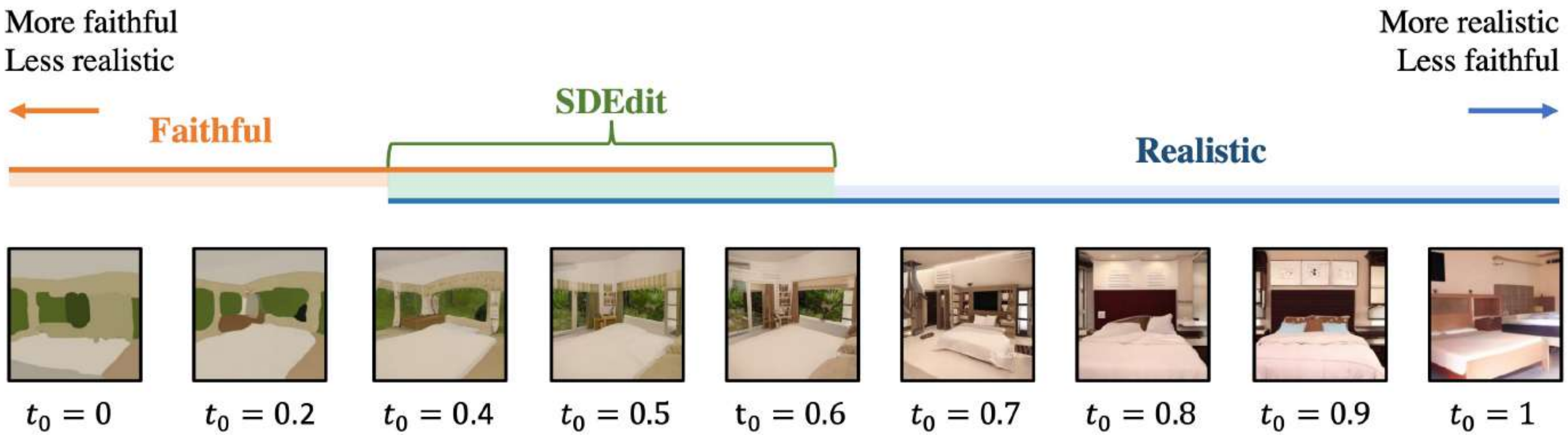
t influence

$t \in [0, 1]$

Set a proper t_0

Compromise realism/faithfulness





Algorithm

Algorithm 1 Guided image synthesis and editing with SDEdit (VE-SDE)

Require: $\mathbf{x}^{(g)}$ (guide), t_0 (SDE hyper-parameter), N (total denoising steps)

Algorithm

Algorithm 1 Guided image synthesis and editing with SDEdit (VE-SDE)

Require: $\mathbf{x}^{(g)}$ (guide), t_0 (SDE hyper-parameter), N (total denoising steps)

$\Delta t \leftarrow \frac{t_0}{N}$

$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

$\mathbf{x} \leftarrow \mathbf{x} + \sigma(t_0)\mathbf{z}$

Algorithm

Algorithm 1 Guided image synthesis and editing with SDEdit (VE-SDE)

Require: $\mathbf{x}^{(g)}$ (guide), t_0 (SDE hyper-parameter), N (total denoising steps)

$\Delta t \leftarrow \frac{t_0}{N}$

$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

$\mathbf{x} \leftarrow \mathbf{x} + \sigma(t_0)\mathbf{z}$ | **Forward init**

Algorithm

Algorithm 1 Guided image synthesis and editing with SDEdit (VE-SDE)

Require: $\mathbf{x}^{(g)}$ (guide), t_0 (SDE hyper-parameter), N (total denoising steps)

$\Delta t \leftarrow \frac{t_0}{N}$

$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

$\mathbf{x} \leftarrow \mathbf{x} + \sigma(t_0)\mathbf{z}$

for $n \leftarrow N$ **to** 1 **do**

$\quad t \leftarrow t_0 \frac{n}{N}$

$\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

$\quad \epsilon \leftarrow \sqrt{\sigma^2(t) - \sigma^2(t - \Delta t)}$

$\quad \mathbf{x} \leftarrow \mathbf{x} + \epsilon^2 \boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}, t) + \epsilon \mathbf{z}$

end for

Return $\mathbf{x}$

Forward init

Backward loop

UniGe | MaLGa

Metrics & quantitative

Hard to compare generative methods

Metrics & quantitative

Hard to compare generative methods

Faithfulness (pure number)

Baselines	Faithfulness score (L_2) ↓
In-domain GAN-1	101.18
In-domain GAN-2	57.11
StyleGAN2-ADA	68.12
e4e	53.76
SDEdit	**32.55**

Metrics & quantitative

Hard to compare generative methods

Faithfulness (pure number)

More realistic (comparison)

Baselines	Faithfulness score (L_2) ↓	SDEdit is more realistic (MTurk) ↑
In-domain GAN-1	101.18	94.96%
In-domain GAN-2	57.11	97.87%
StyleGAN2-ADA	68.12	98.09%
e4e	53.76	80.34%
SDEdit	**32.55**	–

Metrics & quantitative

Hard to compare generative methods

Faithfulness (pure number)

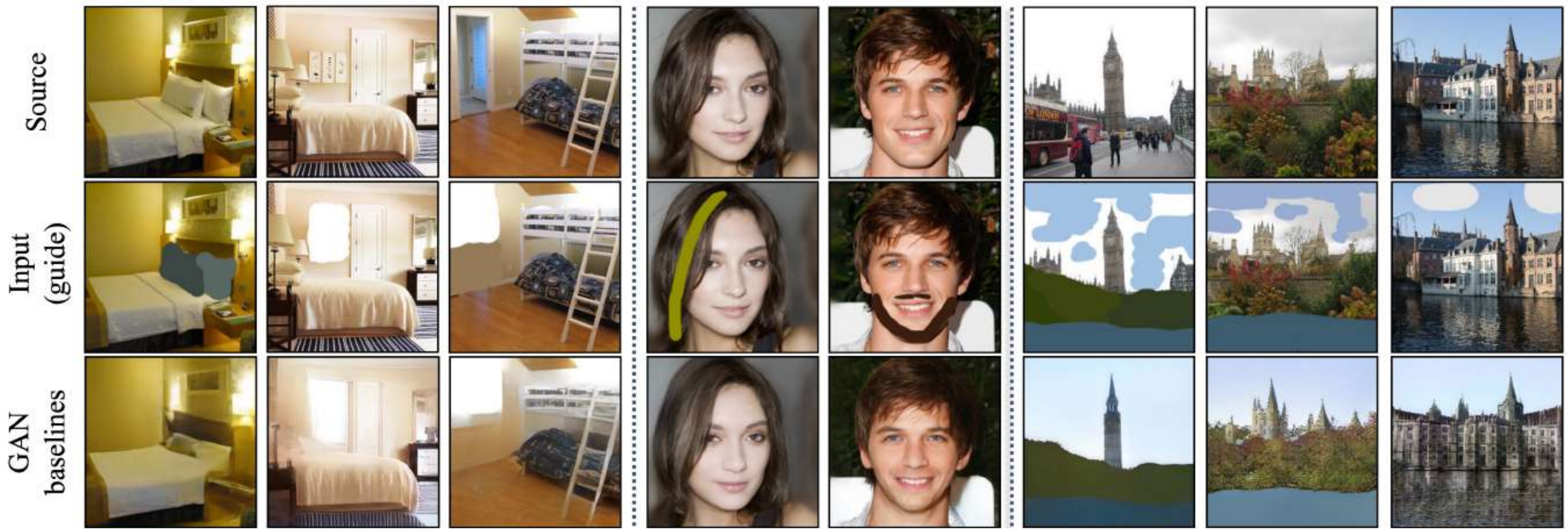
More realistic (comparison)

More satisfactory (comparison)

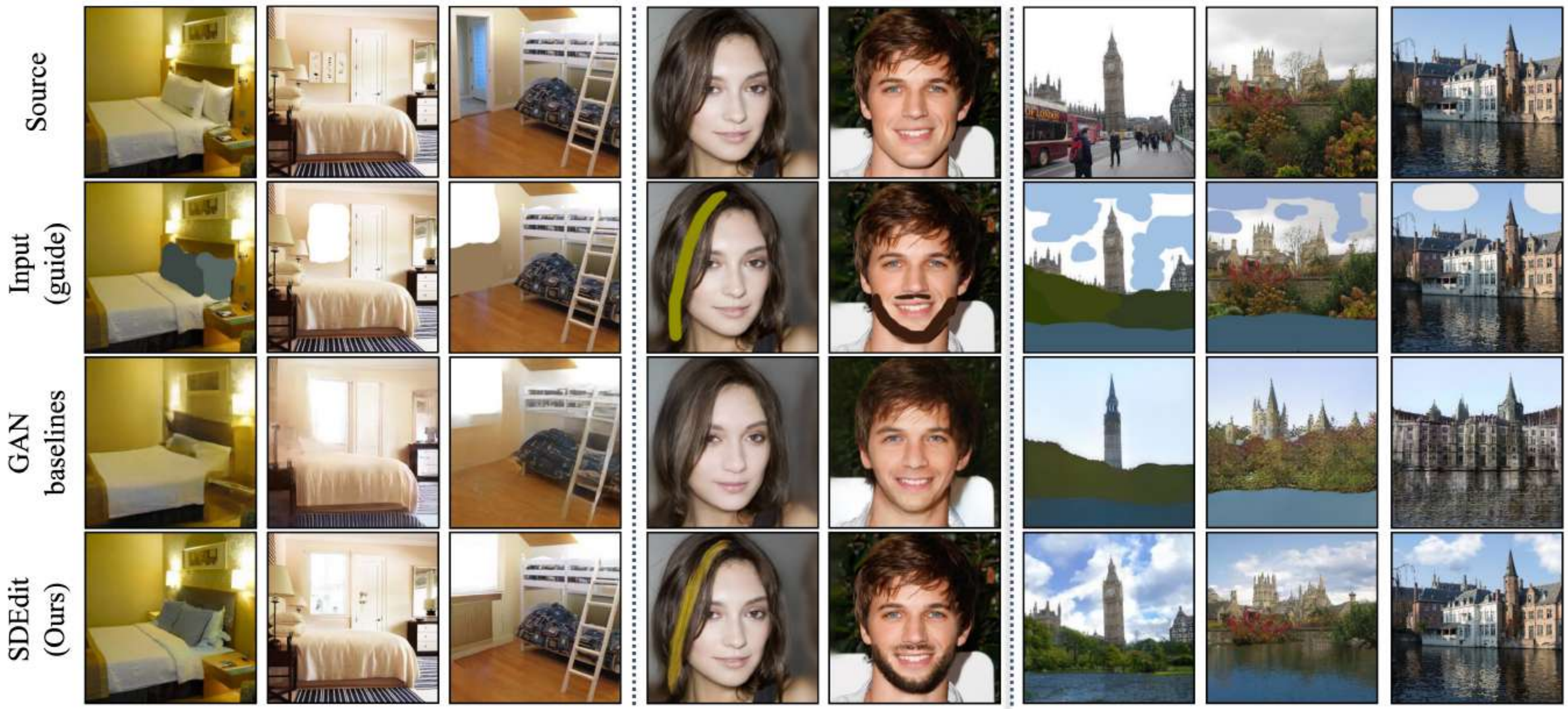
Baselines	Faithfulness score (L_2) ↓	SDEdit is more realistic (MTurk) ↑	SDEdit is more satisfactory (Mturk) ↑
In-domain GAN-1	101.18	94.96%	89.48%
In-domain GAN-2	57.11	97.87%	89.51%
StyleGAN2-ADA	68.12	98.09%	91.72%
e4e	53.76	80.34%	75.43%
SDEdit	**32.55**	–	–

Qualitative

Qualitative



Qualitative



UniGe | MaLGa

Conclusions

- Effective generative method

Conclusions

- Effective generative method
- Robust

Conclusions

- Effective generative method
- Robust
- Pretty slow

UniGe

MaLGa