

h-Edit: Effective and Flexible Diffusion-Based Editing via Doob's *h*-Transform

Toan Nguyen*, Kien Do*, Duc Kieu, Thin Nguyen
A2I2, Deakin University, Australia. *Equal contribution

CVPR 2025



DEAKIN
UNIVERSITY

DEAKIN
APPLIED ARTIFICIAL
INTELLIGENCE INITIATIVE

Training-Free Image Editing with Diffusion Models (1)

Source: a woman with [black] hair and red lipstick holding a flower
Edit: a woman with [silver] hair and red lipstick holding a flower



Source

NMG + P2P

StyleD + P2P

NP + P2P

NT + P2P

PnP Inv + P2P

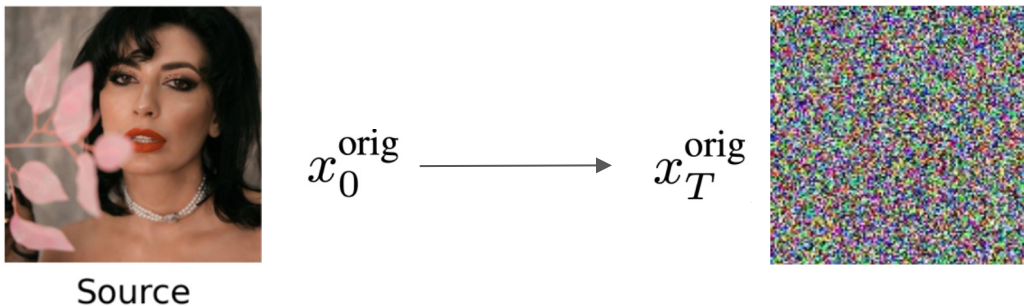
h-edit-D + P2P

Goal: Generate images that align with editing prompt (editing **accuracy**) while being **faithful** to the original image.

→ How to achieve and balance both targets? (**Problem 1**)

How diffusion models perform editing?

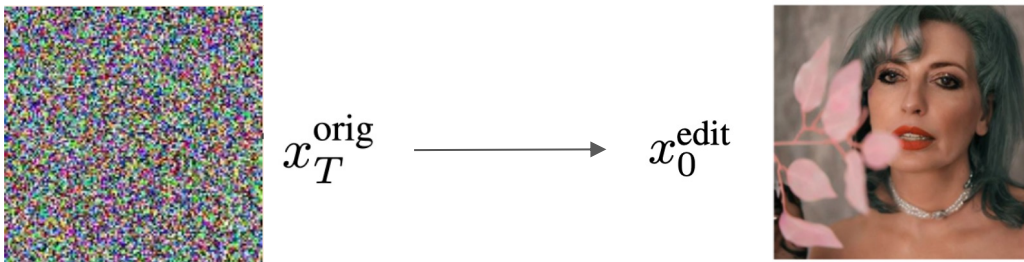
Inversion:



- Two popular types are **DDIM Inversion** and **DDPM Inversion** (slightly better).

Editing: Perform **step-by-step sampling** with editing conditions.

- Often requires further **attention control** for faithfulness.



Training-Free Image Editing with Diffusion Models (2)

Problem 2:

How to **combine** multiple types of editing effectively?

For example: Text-guided and Style Editing.

Style						
Source						
EF w/ P2P	 336.2	 567.3	 459.2	 397.1	 409.2	 498.4
h-Edit-R w/ P2P	 277.6	 443.2	 348.5	 276.8	 247.3	 456.6
	- 'flower'	- 'square' + 'round'	- 'dog' + 'monkey'	- 'cat' + 'bear'	- 'dog' + 'wolf'	- 'husky dog'

Training-Free Image Editing with Diffusion Models (3)

Problem 3:

- Current literature do not pay attention on the editing process of diffusion models, lacking of **theoretical foundation**.
- Struggle with the first two problems!!
 - Most of current text-guided methods focus on sampling from $p(x_{t-1} \mid x_t, c^{\text{edit}})$
- **No guarantee** to fall into the **target** distribution $p(x_0) p(c^{\text{edit}} \mid x_0)$
- By this approach, there is **no control for the trade-off** between editing & faithfulness.
- How to **combine** with other editing conditions?

Motivation of h -Edit (1)

- We know the abstract target distribution is $p(x_0) p_Y(y | x_0)$
- We know the starting distribution for editing is $p(x_T)$
- Why don't we **build a bridge** to connect them?

Can be
anything:
text, graph,
reference
images, audio.

→ **Guarantee** to fall into the **target** distribution.

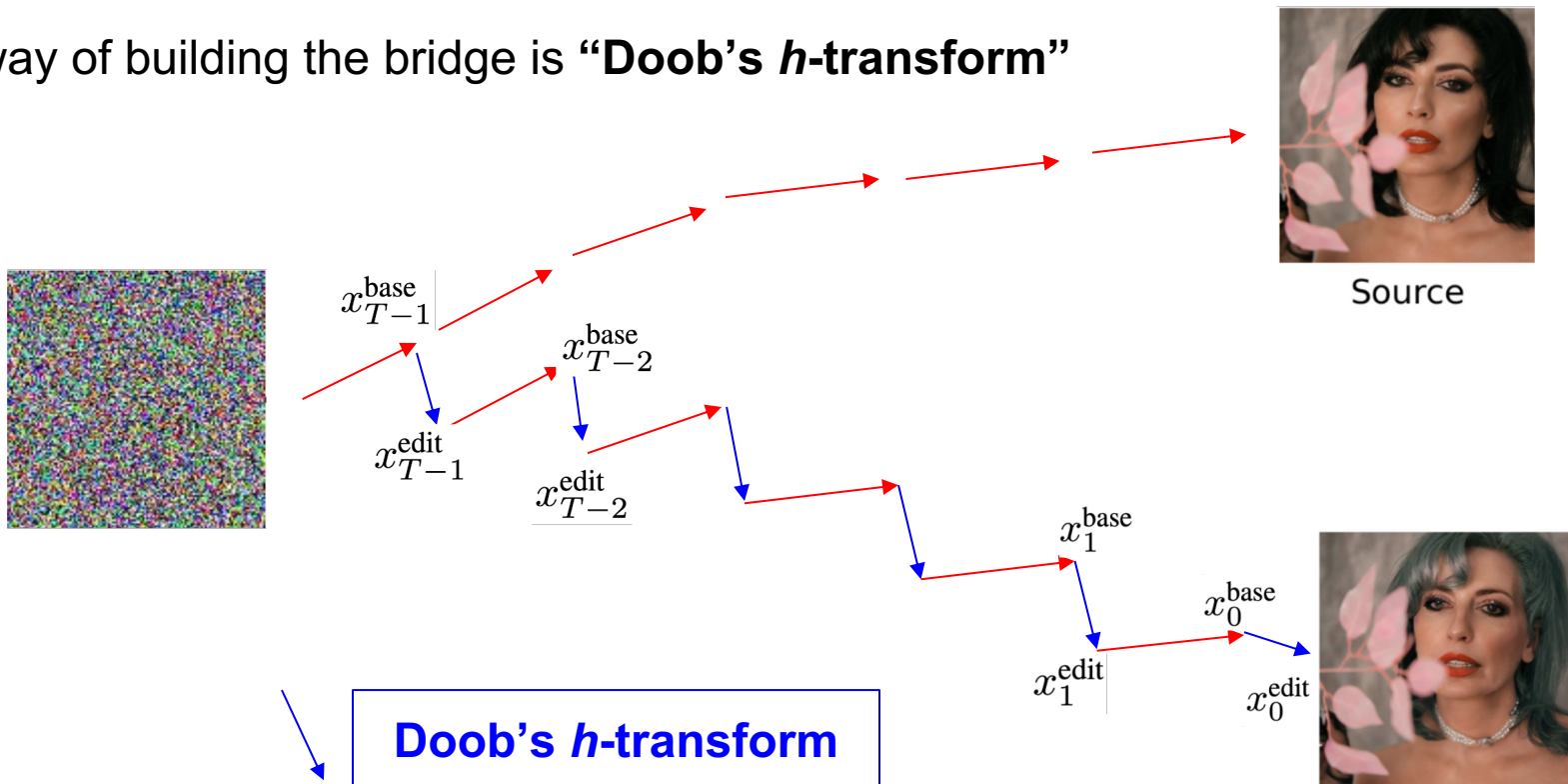
→ For **combining**, we adjust the target distribution to

$$p(x_0) p_Y(y_1 | x_0) p(x_0) p_Y(y_2 | x_0) \dots p(x_0) p_Y(y_n | x_0)$$

→ Naturally decompose the update into “**reconstruction**” term and “**editing**” term.

How *h*-Edit works?

A way of building the bridge is “**Doob’s *h*-transform**”



How h -Edit works? h -transform + LMC sampling!

Implicit form:
$$p_{\theta}^h(x_{t-1}|x_t) = p_{\theta}(x_{t-1}|x_t) \frac{h(x_{t-1}, t-1)}{h(x_t, t)} \longrightarrow p_Y(y | x_{t-1})$$

Explicit form:
$$p^h(x_t) = \frac{p(x_t)h(x_t, t)}{\mathbb{E}_{p(x_0)}[h(x_0, 0)]} \longrightarrow p_Y(y | x_t)$$

To sample x_{t-1}^{edit} , we perform Langevin Monte Carlo sampling to sample from $p_{\theta}^h(x_{t-1}|x_t)$ or with the score of $p^h(x_t)$

How h -Edit works? h -transform + LMC sampling! (2)

Explicit form:

$$\begin{aligned}x_{t-1} &\approx x_t + \eta \nabla_{x_t} \log (p(x_t) h(x_t, t)) + \sqrt{2\eta} z \\&= \left(x_t + \eta \nabla_{x_t} \log p(x_t) + \sqrt{2\eta} z \right) \\&\quad + \eta \nabla_{x_t} \log h(x_t, t) \\&= \underbrace{x_{t-1}^{\text{base}}}_{\text{rec.}} + \underbrace{\eta \nabla_{x_t} \log h(x_t, t)}_{\text{editing}}\end{aligned}$$

- Naturally decompose the update into “**reconstruction**” term and “**editing**” term.
- We can have **multiple** “editing” terms with any form.

How h -Edit works? h -transform + LMC sampling! (3)

Implicit form:

$$\begin{aligned}x_{t-1} &\approx x_{t-1}^{\text{init}} + \gamma \nabla_{x_{t-1}} \log p^h(x_{t-1}|x_t) + \sqrt{2\gamma}z \\&= \left(x_{t-1}^{\text{init}} + \gamma \nabla_{x_{t-1}} \log p(x_{t-1}|x_t) + \sqrt{2\gamma}z \right) \\&\quad + \gamma \nabla_{x_{t-1}} \log h(x_{t-1}, t-1) \\&\approx \underbrace{x_{t-1}^{\text{base}}}_{\text{rec.}} + \underbrace{\gamma \nabla_{x_{t-1}} \log h(x_{t-1}^{\text{base}}, t-1)}_{\text{editing}}\end{aligned}$$

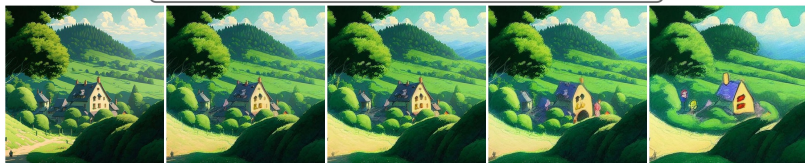
- What's special about implicit form? → **Optimization** over the space of x_{t-1} as the editing term acts on x_{t-1}

$$\begin{aligned}x_{t-1}^{(0)} &= x_{t-1}^{\text{base}} \\x_{t-1}^{(k+1)} &= x_{t-1}^{(k)} + \gamma \nabla_{x_{t-1}} \log h(x_{t-1}^{(k)}, t-1)\end{aligned}$$

**Multiple
optimization steps
for hard editing
cases**

Results of Implicit h-Edit using multiple optimization steps

Source: an anime painting of a house on a hill
Edit: [kids crayon drawing of] an anime painting of a house on a hill



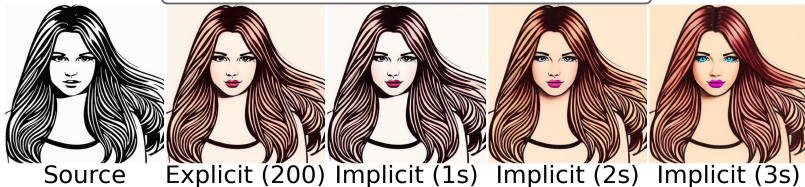
Source: a painting of a waterfall in the mountains
Edit: a painting of a waterfall [and angels] in the mountains



Source: a [square] wooden crate filled with radishes and greens
Edit: a [round] wooden crate filled with radishes and greens



Source: a [black and white] drawing of a woman with long hair
Edit: a [colorful and detailed] drawing of a woman with long hair



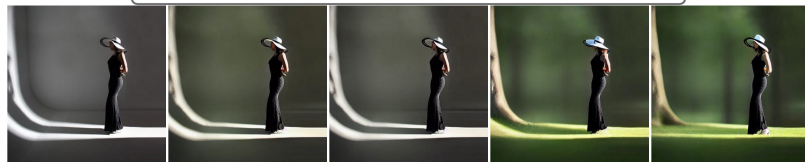
Source: the palace of the emperor in vienna [in a sunny day]
Edit: the palace of the emperor in vienna [in a cloudy day]



Source: a [dry] tree in the wild
Edit: a [blooming] tree in the wild



Source: a woman in a black dress and hat stands in front of [a large wall]
Edit: a woman in a black dress and hat stands in front of [a large park]



Source: a squirrel sitting on top of a wooden platform
Edit: a squirrel sitting on top of a wooden platform [reading a book]



How h -Edit works? Design h -function effectively!

- Text-guided editing

We introduce \hat{w}^{orig} at editing time
(different to w^{orig} at inversion time)

$$\begin{aligned}\nabla \log h(x_{t-1}, t-1) \\ &= \nabla \log p(y|x_{t-1}) \\ &= \nabla \log p(x_{t-1}|y) - \nabla \log p(x_{t-1})\end{aligned}$$

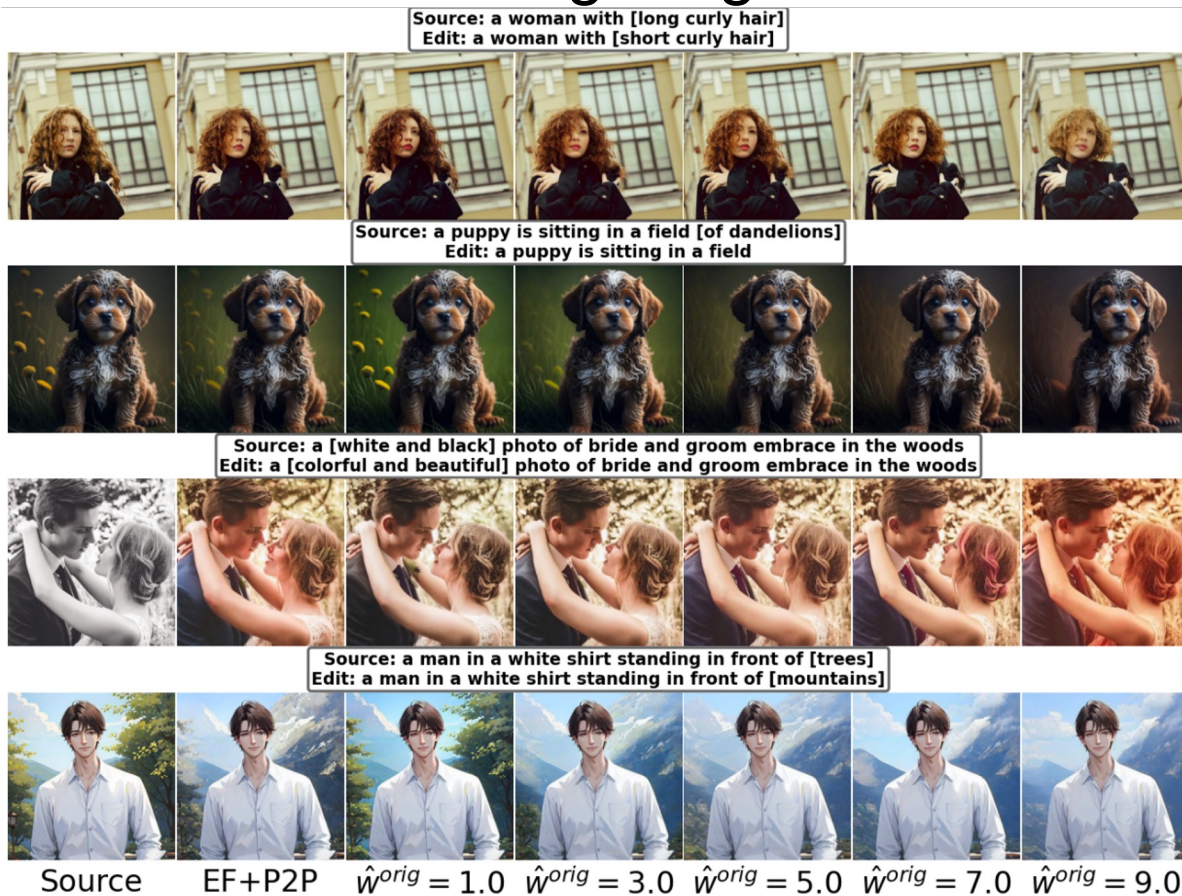
$$\frac{-\tilde{\epsilon}_{\theta}(x_{t-1}, t-1, c^{\text{edit}})}{\sigma_{t-1}}$$

$$\frac{-\tilde{\epsilon}_{\theta}(x_{t-1}, t-1, c^{\text{orig}})}{\sigma_{t-1}}$$

$$(1 - w^{\text{edit}}) \epsilon_{\theta}(x_{t-1}, t-1, \emptyset) + w^{\text{edit}} \epsilon_{\theta}(x_{t-1}, t-1, c^{\text{edit}})$$

$$(1 - \hat{w}^{\text{orig}}) \epsilon_{\theta}(x_{t-1}, t-1, \emptyset) + \hat{w}^{\text{orig}} \epsilon_{\theta}(x_{t-1}, t-1, c^{\text{orig}})$$

\hat{w}^{orig} plays the role of removing “negative” information!



Quantitative Results on PIE-Bench

Inv.	Attn.	Method	CLIP Sim. \uparrow	Local CLIP \uparrow	DINO Dist. $\times 10^2\downarrow$	LPIPS $\times 10^2\downarrow$	SSIM $\times 10\uparrow$	PSNR \uparrow
Deter.	P2P	NP	0.246	0.140	1.62	6.90	8.34	26.21
		NT	0.248	0.130	1.34	6.07	8.41	27.03
		StyleD	0.248	0.085	1.17	6.61	8.34	26.05
		NMG	0.249	0.087	1.32	5.59	8.47	27.05
		PnP Inv	0.250	0.095	1.17	5.46	8.48	27.22
		<i>h</i> -Edit-D	0.253	0.147	1.17	4.85	8.54	27.87
Random	None	EF	0.254	0.122	1.29	6.09	8.37	25.87
		LEDITS++	0.254	0.113	2.34	8.88	8.11	23.36
		<i>h</i> -Edit-R	0.255	0.148	1.28	5.55	8.46	26.43
	P2P	EF	0.255	0.126	1.51	5.70	8.40	26.30
		<i>h</i> -Edit-R	0.256	0.159	1.45	5.08	8.50	26.97

Better editing fidelity!

Better faithfulness!

Comparison with baselines: Deterministic Inversion



Source NMG + P2P StyleD + P2P NP + P2P NT + P2P PnP Inv + P2P **h-edit-D + P2P**



Source NMG + P2P StyleD + P2P NP + P2P NT + P2P PnP Inv + P2P **h-edit-D + P2P**

Comparison with baselines: Random Inversion



Source

LEDITS++

EF + P2P

h-edit-R + P2P



Source

LEDITS++

EF + P2P

h-edit-R + P2P

Editing with external reward models

$$\nabla \log h(x_{t-1}, t-1) = \nabla \log \left[\sum_{x_0} p(x_0 | x_{t-1}) h(x_0, 0) \right]$$

$$= \nabla \log [\mathbb{E}_{p(x_0|x_{t-1})} [h(x_0, 0)]]$$

$$\approx \nabla \log [h(x_{0|t-1}, 0)]$$








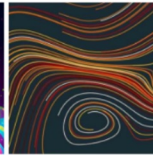











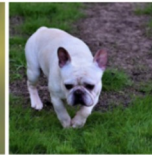
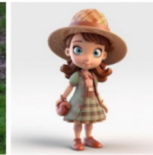






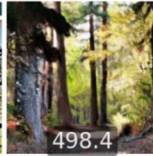



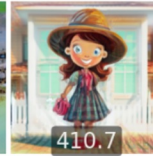

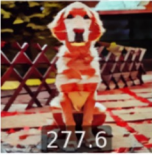


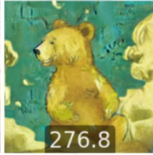

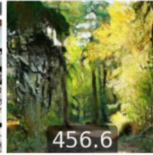
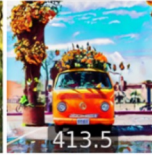
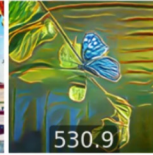

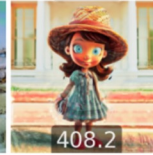

$x_{0|t-1}$ can be
computed with
Tweedie's formula

We can perform editing with gradients from external reward models trained on x_0

Editing with reward models: Face Swapping with ArcFace



Editing with conditional score and reward models: Text-guided and Style Editing with CLIP

Style											
Source											
EF w/ P2P	 336.2	 567.3	 459.2	 397.1	 409.2	 498.4	 397.8	 787.6	 427.4	 410.7	 721.3
h-Edit-R w/ P2P	 277.6	 443.2	 348.5	 276.8	 247.3	 456.6	 413.5	 530.9	 299.7	 408.2	 562.0
	-'flower'	-'square' +'round'	-'dog' +'monkey'	-'cat' +'bear'	-'dog' +'wolf'	-'husky dog'	-'surfboards' +'flowers'	-'bird' +'butterfly'	-'bulldog' +'rat'	+'house'	+'chintzy doll'

*THANK YOU FOR
LISTENING*

Our paper: <https://arxiv.org/abs/2503.02187>

Source code: <https://github.com/nktoan/h-edit>