Gradient-free Decoder Inversion in Latent Diffusion Models

Seongmin Hong[1] **Suh Yoon Jeon**[1] **Kyeonghyun Lee**[1]

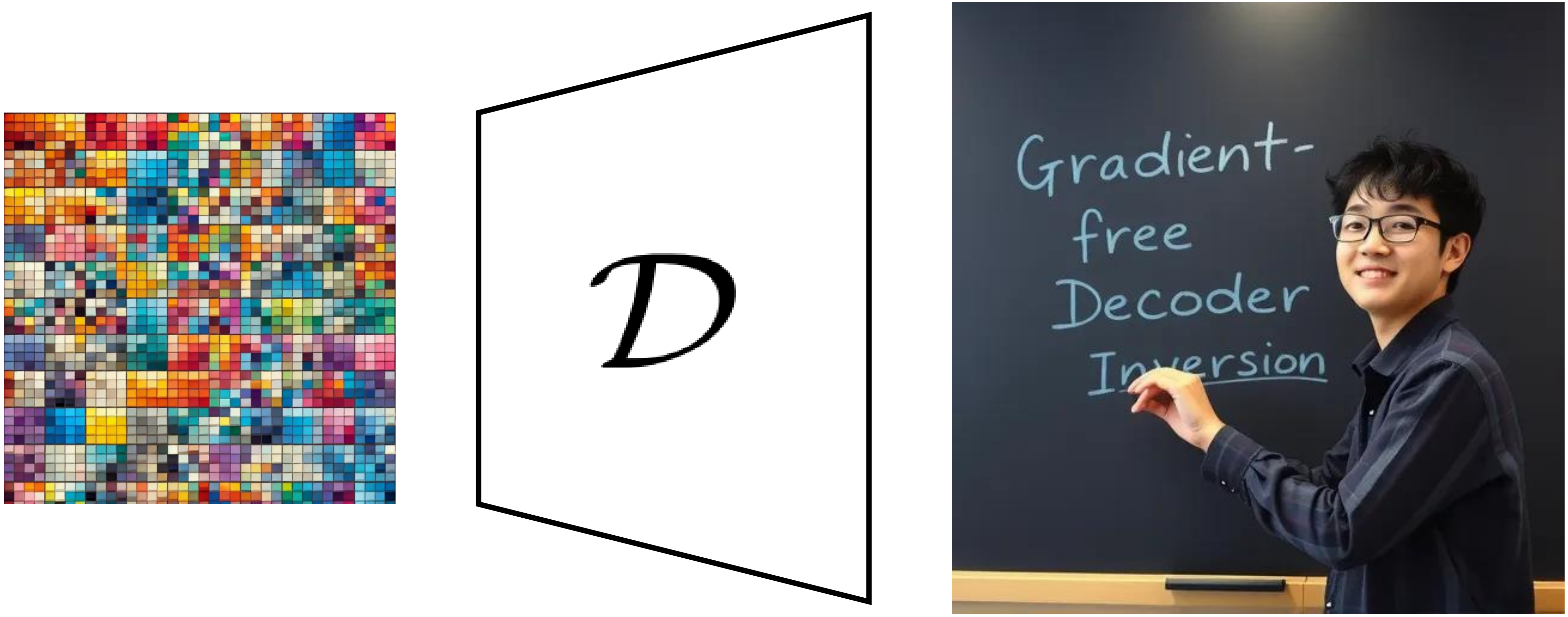
Ernest K. Ryu[2,*] **Se Young Chun**[1,3,*]

[1]Dept. of Electrical and Computer Engineering, [3]INMC & IPAI, Seoul National University
[2]Dept. of Mathematics, University of California, Los Angeles
{smhongok, euniejeon, litiphysics, sychun}@snu.ac.kr, eryu@math.ucla.edu

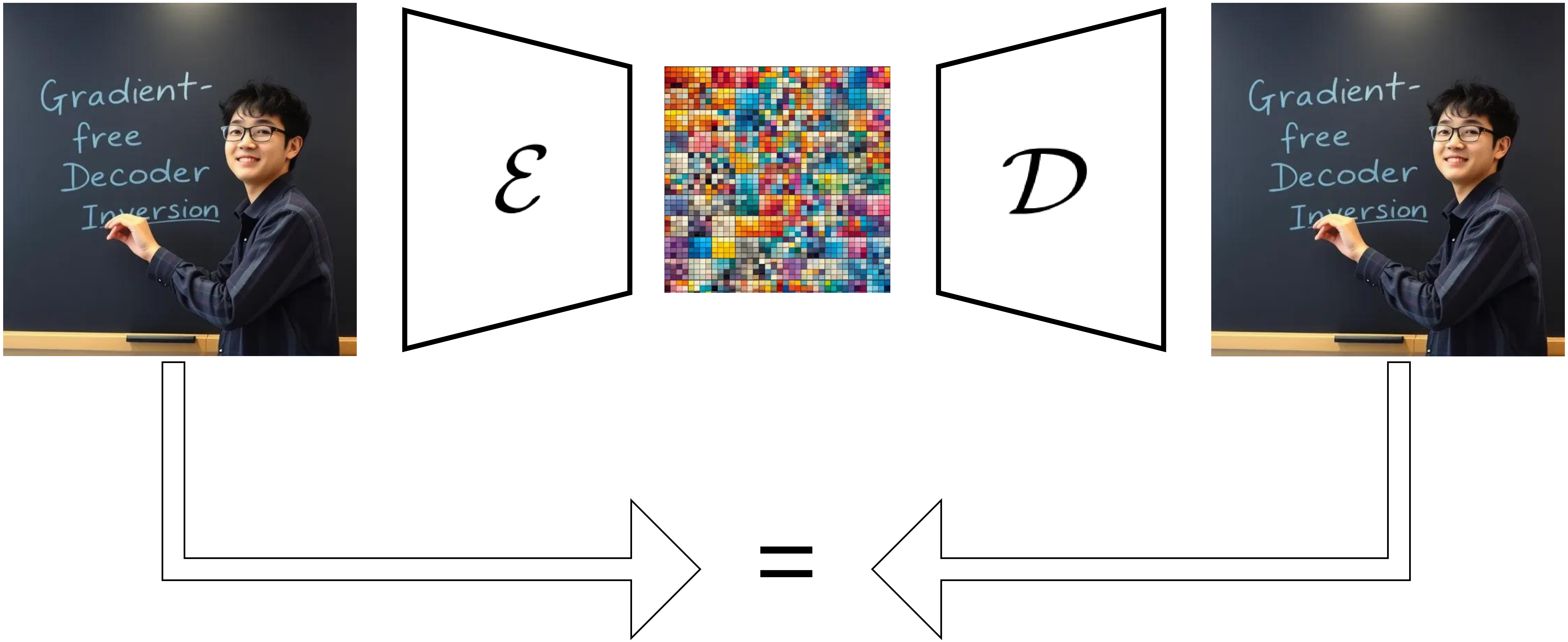
NEURAL INFORMATION PROCESSING SYSTEMS

Intelligent Computational imaging Lab.

Decoder is used in latent diffusion models.

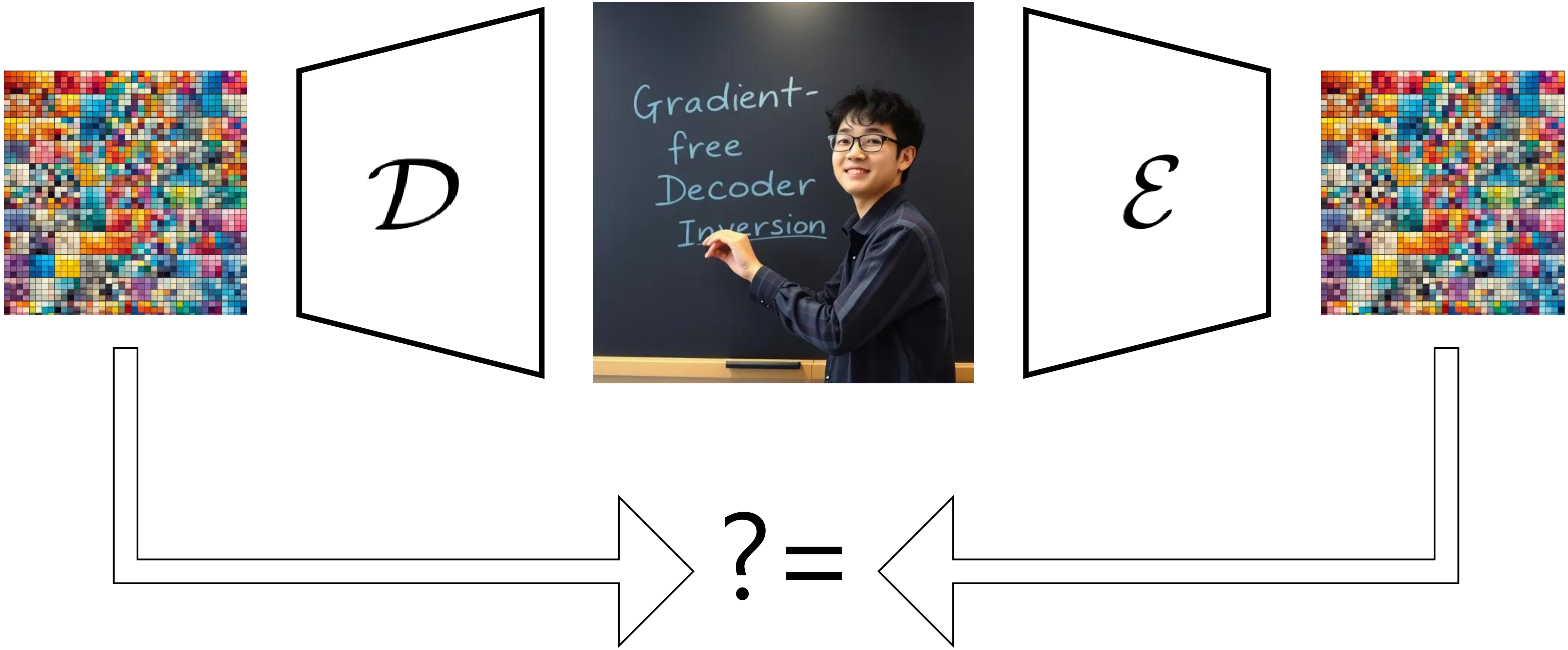


Encoder is the right-inverse of the decoder.



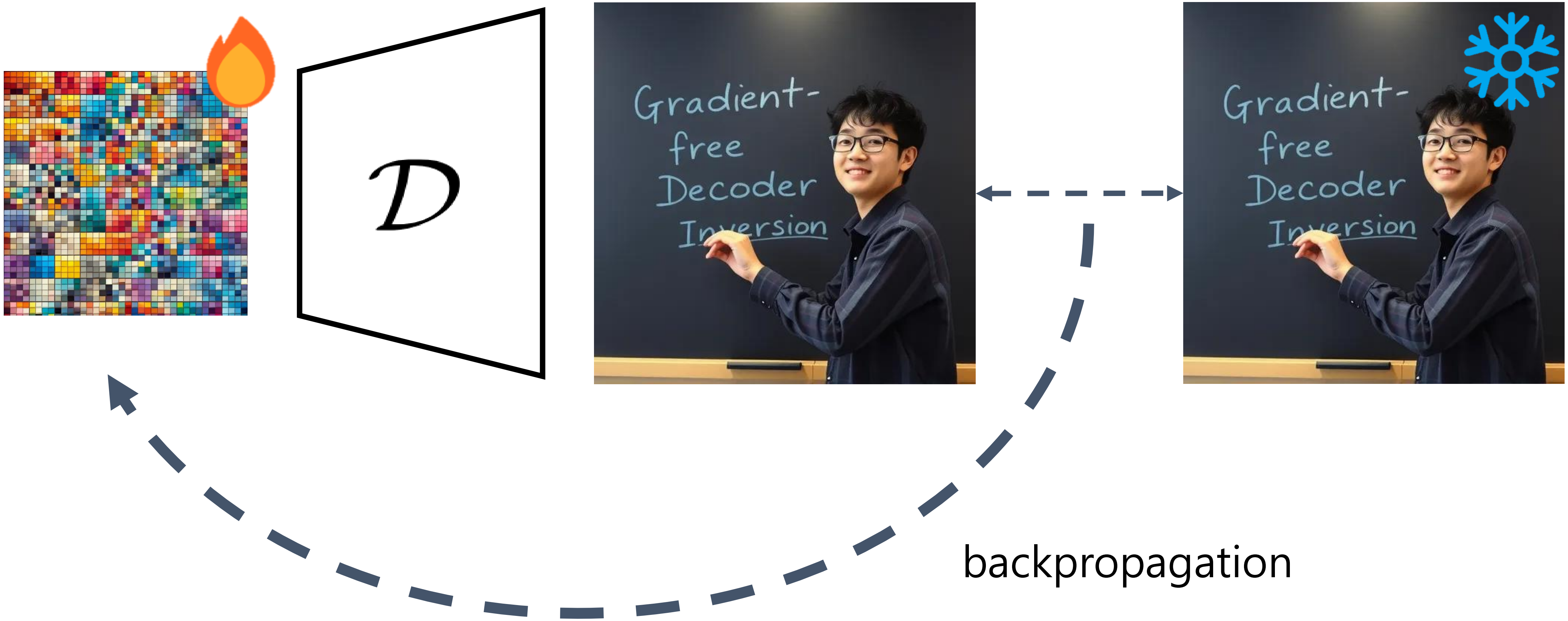
What is the left-inverse of the Decoder?

Encoder? No.



* Equality holds if the encoder and decoder are linear. (Remark 2)

Gradient descent? Heavy.



We propose a gradient-free decoder inversion algorithm!

Motivation of our grad-free method

$$\underset{\boldsymbol{z}\in\mathbb{R}^F}{\text{find}} \quad \boldsymbol{x} = \mathcal{D}(\boldsymbol{z})$$

is difficult.

$$\underset{\boldsymbol{z}\in\mathbb{R}^F}{\text{find}} \quad \mathcal{E}(\boldsymbol{x}) = \mathcal{E}(\mathcal{D}(\boldsymbol{z}))$$

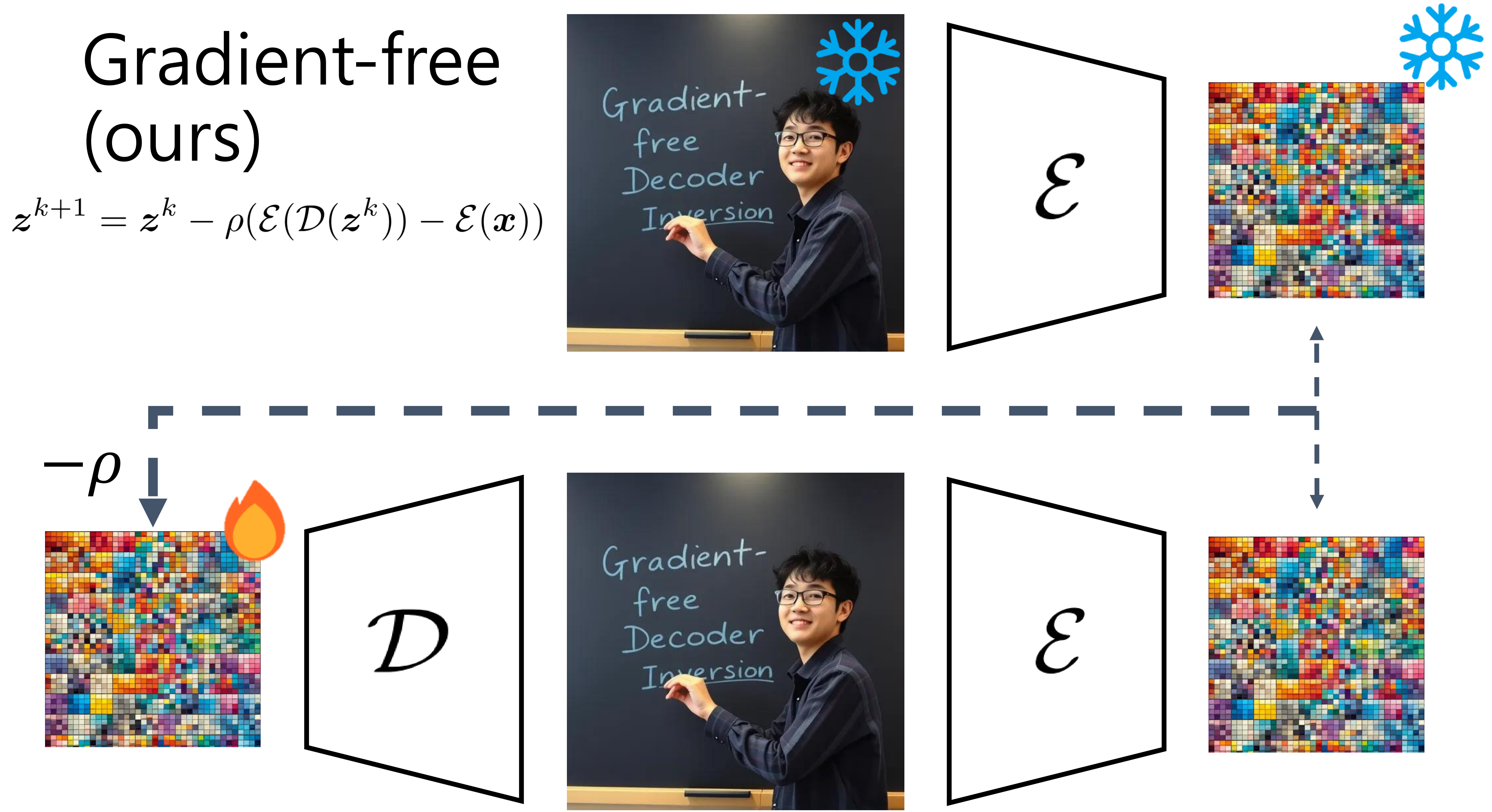
is easier (Remark 1).

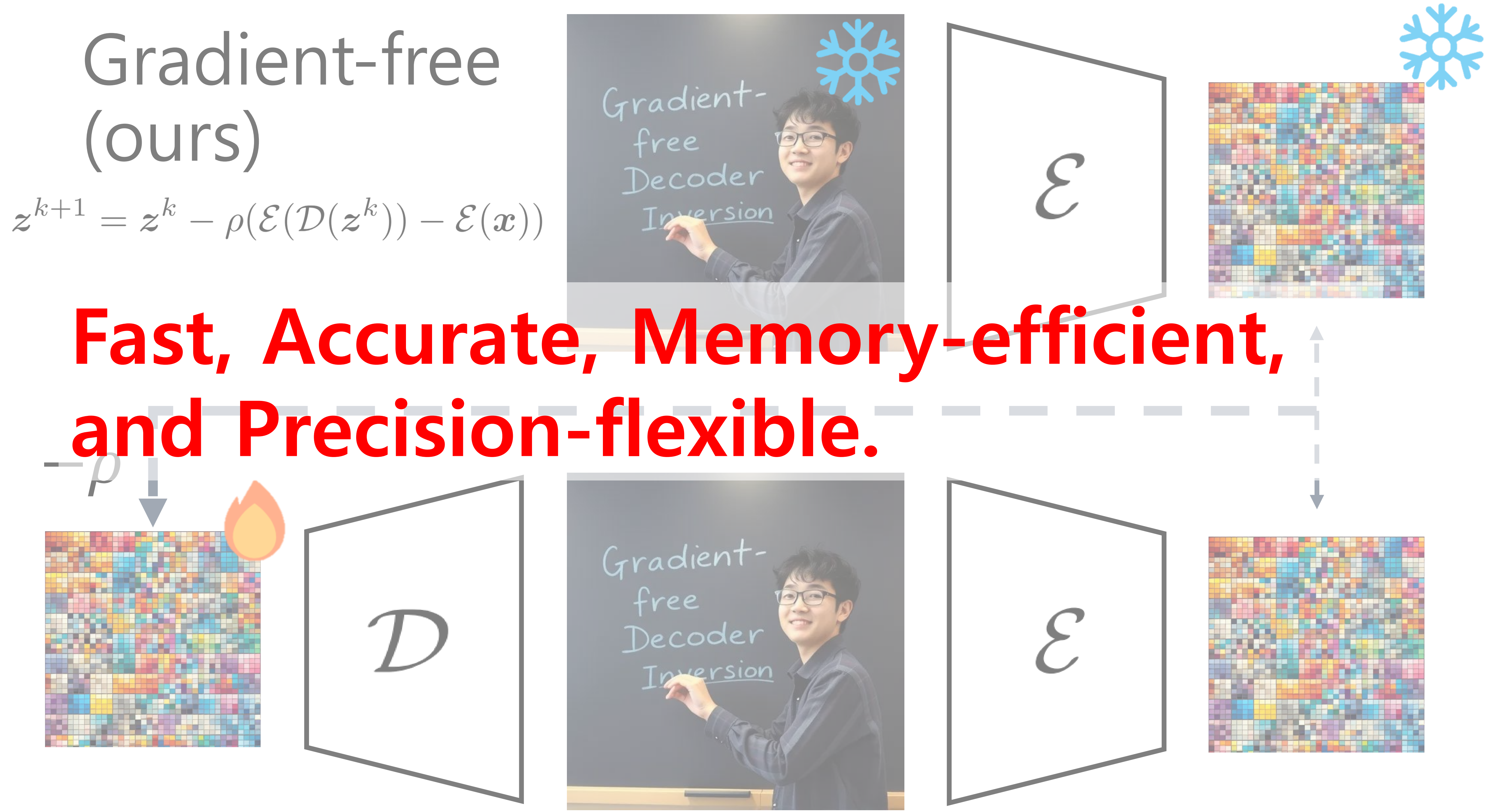
This is equivalent to:

$$\underset{\boldsymbol{z}\in\mathbb{R}^F}{\text{find}} \quad \boldsymbol{z} = \boldsymbol{z} - \rho(\mathcal{E}(\mathcal{D}(\boldsymbol{z})) - \mathcal{E}(\boldsymbol{x})), \quad \forall\rho \in \mathbb{R} \cap \{0\}^C$$

Fixed point iteration:

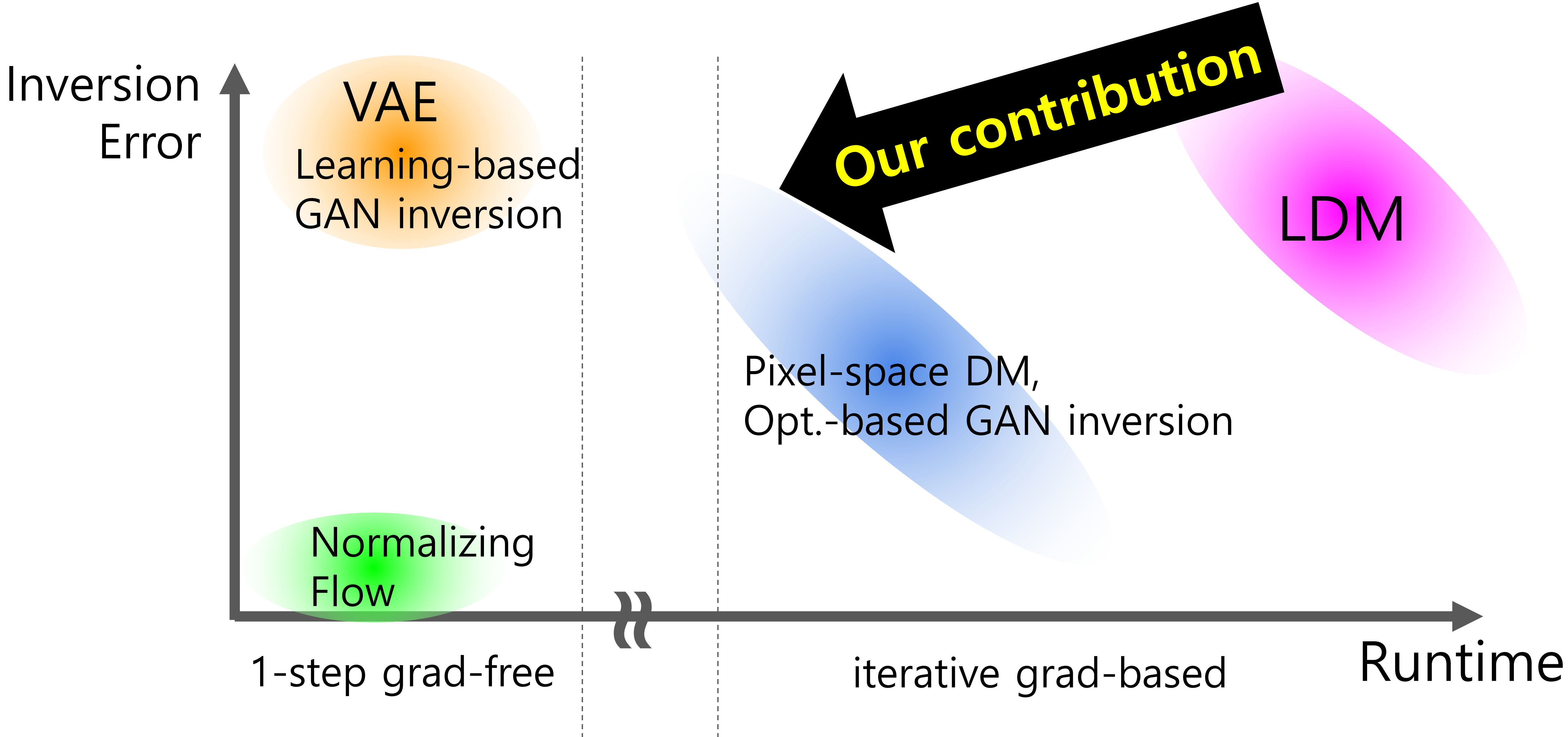
$$\boldsymbol{z}^{k+1} = \boldsymbol{z}^k - \rho(\mathcal{E}(\mathcal{D}(\boldsymbol{z}^k)) - \mathcal{E}(\boldsymbol{x}))$$


Gradient-free
(ours)
$z^{k+1} = z^k - \rho(\mathcal{E}(\mathcal{D}(z^k)) - \mathcal{E}(x))$
Gradient-free Decoder Inversion
$\mathcal{E}$
$-\rho$
$\mathcal{D}$
Gradient-free Decoder Inversion
$\mathcal{E}$



Gradient-free
(ours)
$z^{k+1} = z^k - \rho(\mathcal{E}(\mathcal{D}(z^k)) - \mathcal{E}(x))$
Fast, Accurate, Memory-efficient,
and Precision-flexible.
$-\rho$
$\mathcal{E}$
$\mathcal{D}$
$\mathcal{E}$
Gradient-free Decoder Inversion
Gradient-free Decoder Inversion

We make LDM more inversion-friendly!



Convergence Analysis

Our method provably converges.

Theorem 1 (Convergence of the forward step method)**.** *Let $\beta > 0$, $0 < \rho < 2\beta$, and $\boldsymbol{x} \in \mathbb{R}^N$. Assume $\mathcal{T}(\cdot) = \mathcal{E} \circ \mathcal{D}(\cdot) - \mathcal{E}(\boldsymbol{x})$ is continuous. Consider the iteration*

$$\boldsymbol{z}^{k+1} = \boldsymbol{z}^k - \rho \mathcal{T} \boldsymbol{z}^k \qquad \textit{for} \quad k = 0, 1, \ldots \tag{8}$$

Assume $\boldsymbol{z}^\star$ is a zero of $\mathcal{T}$ (i.e., $\mathcal{T}\boldsymbol{z}^\star = 0$) and

$$\langle \mathcal{T}\boldsymbol{z}^k, \boldsymbol{z}^k - \boldsymbol{z}^\star \rangle \geq \beta \|\mathcal{T}\boldsymbol{z}^k\|_2^2 \quad \textit{for} \quad k = 0, 1, \ldots \tag{9}$$

Then, $\mathcal{T}\boldsymbol{z}^k \to 0$. If, furthermore, $\boldsymbol{z}^k \to \boldsymbol{z}^\infty$, then $\boldsymbol{z}^\infty$ is a zero of $\mathcal{T}$ (i.e., $\mathcal{T}\boldsymbol{z}^\infty = 0$).

Our method provably converges with momentum:

$$\boldsymbol{y}^k = \boldsymbol{z}^k + \alpha(\boldsymbol{z}^k - \boldsymbol{z}^{k-1})$$
$$\boldsymbol{z}^{k+1} = \boldsymbol{y}^k - 2\lambda\beta\mathcal{T}\boldsymbol{y}^k,$$

Theorem 2 (Convergence of the inertial KM iterations)**.** *Let* $0 < \alpha < 1$, $\beta > 0$, $\lambda > 0$ *and* $\boldsymbol{x} \in \mathbb{R}^N$. *Assume* $\mathcal{T}(\cdot) = \mathcal{E} \circ \mathcal{D}(\cdot) - \mathcal{E}(\boldsymbol{x})$ *is continuous. Let* $(\boldsymbol{z}^k, \boldsymbol{y}^k)$ *satisfy (10) and (11). Assume* $\boldsymbol{z}^\star$ *is a zero of* $\mathcal{T}$ *(i.e.,* $\mathcal{T}\boldsymbol{z}^\star = 0$*) and the following holds:*

$$\langle \mathcal{T}\boldsymbol{y}^k, \boldsymbol{y}^k - \boldsymbol{z}^\star \rangle \geq \beta \|\mathcal{T}\boldsymbol{y}^k\|_2^2 \quad \textit{for} \quad k = 0, 1, \ldots \tag{12}$$

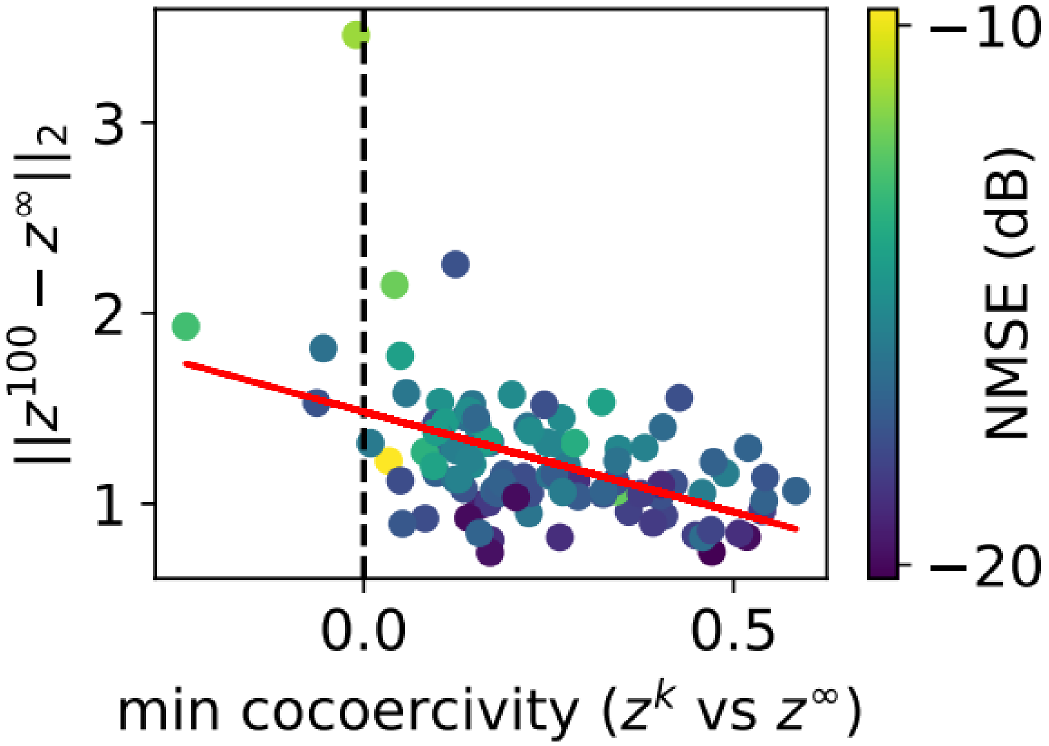
If

$$\lambda(1 - \alpha + 2\alpha^2) < (1 - \alpha)^2, \tag{13}$$

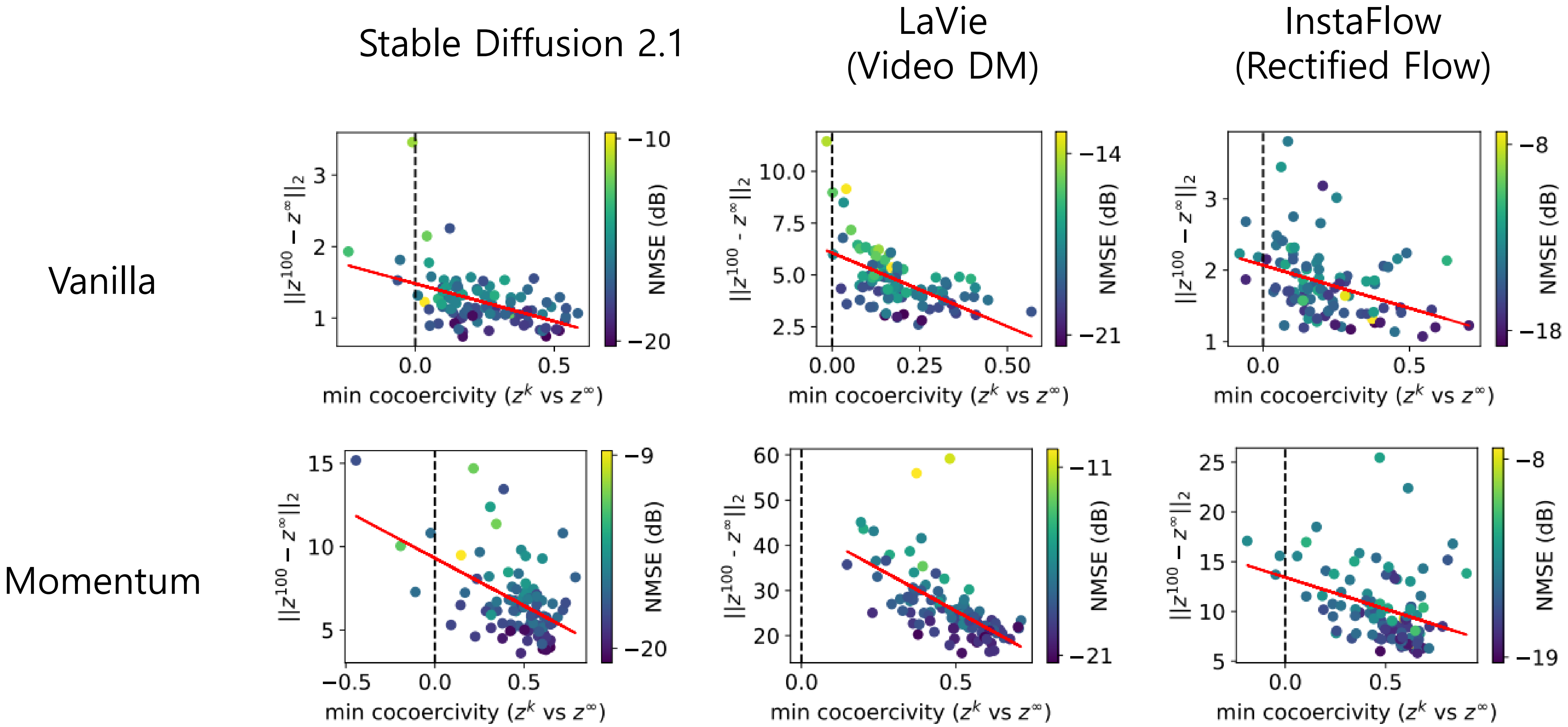
then $(\boldsymbol{y}^k)$ *and* $(\boldsymbol{z}^k)$ *converge to the same limit points.*

Validation on the assumption

- Cocoercivity: $$\frac{\langle \mathcal{E}\mathcal{D}\boldsymbol{z}^{\infty} - \mathcal{E}\mathcal{D}\boldsymbol{z}^{k}, \boldsymbol{z}^{\infty} - \boldsymbol{z}^{k} \rangle}{\|\mathcal{E}\mathcal{D}\boldsymbol{z}^{\infty} - \mathcal{E}\mathcal{D}\boldsymbol{z}^{k}\|_2^2} \geq \beta > 0$$



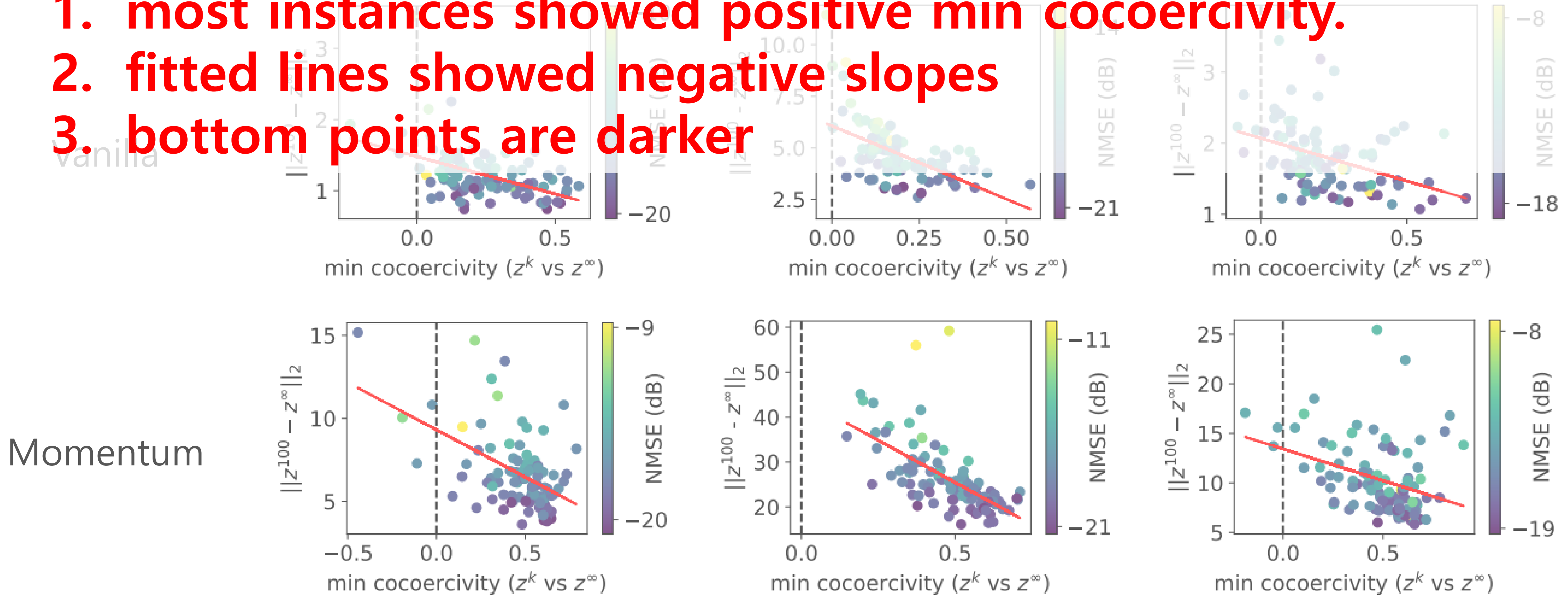
Validation on the assumption in many models



Causality & Evidences

Most instances ⇒ Cocoercivity ⇒ Convergence ⇒ Accuracy

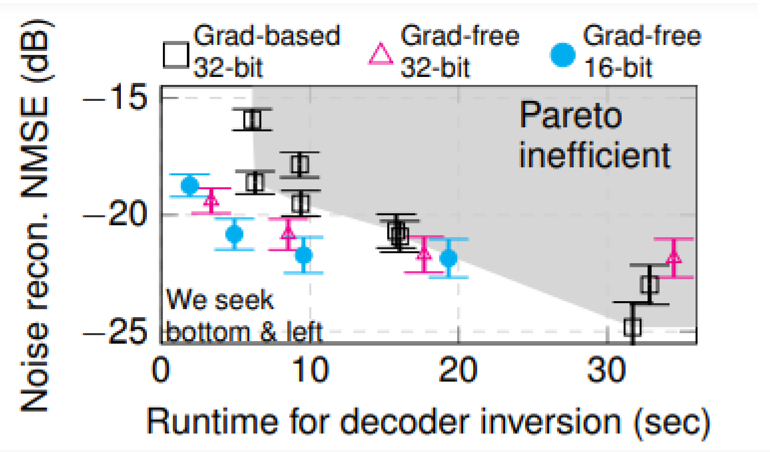
1. **most instances showed positive min cocoercivity.**
2. **fitted lines showed negative slopes**
3. **bottom points are darker**



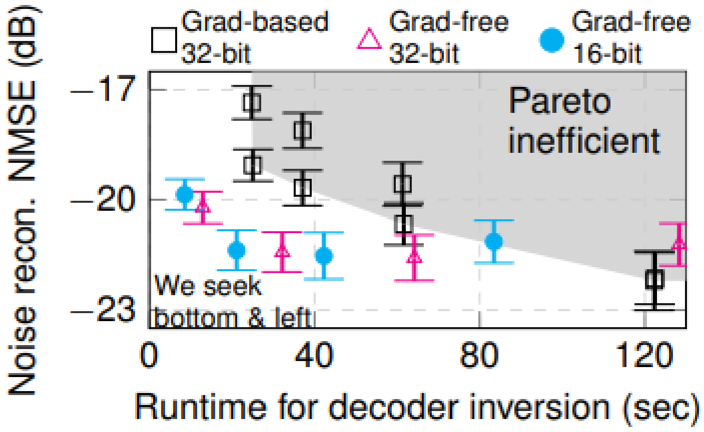
Experiments & Applications

Fast, Accurate, Memory-efficient, Precision-flexible

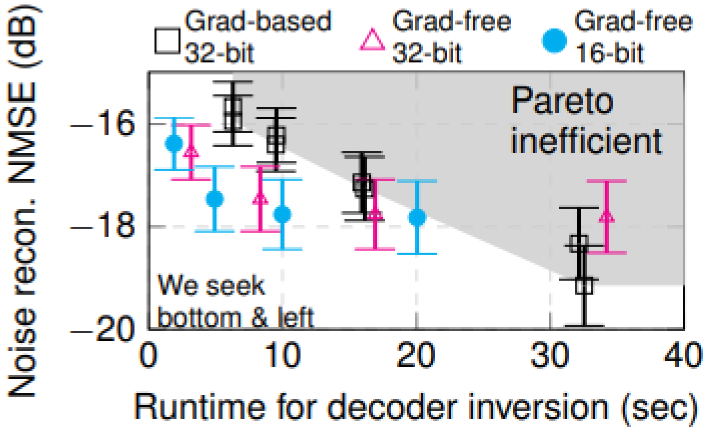
Stable Diffusion 2.1

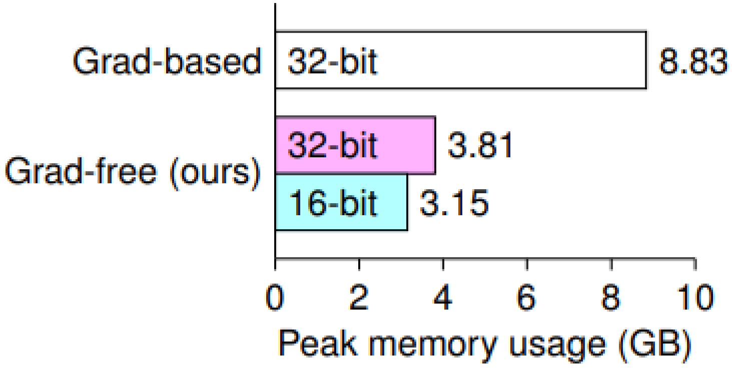


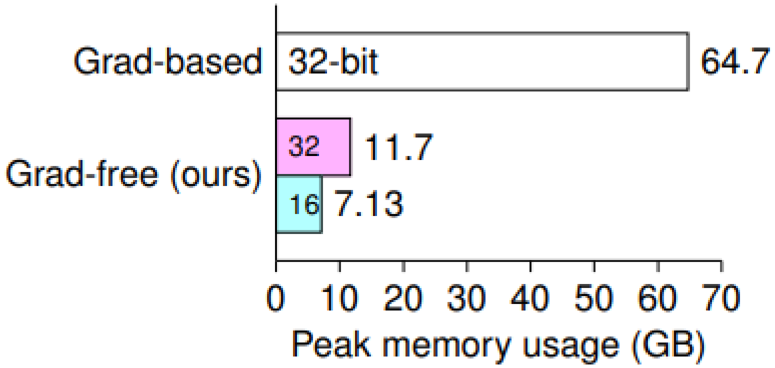
LaVie
(Video DM)

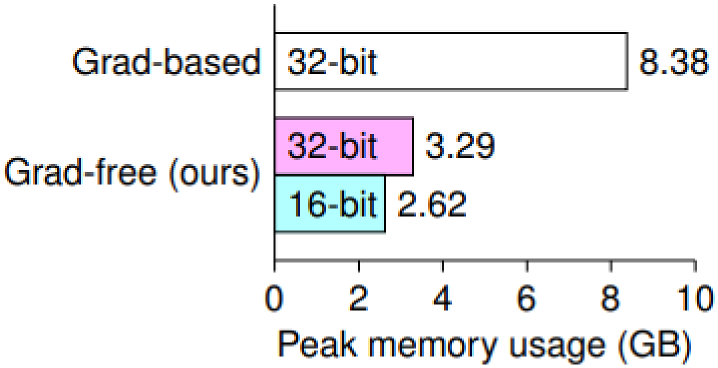


InstaFlow
(Rectified Flow)







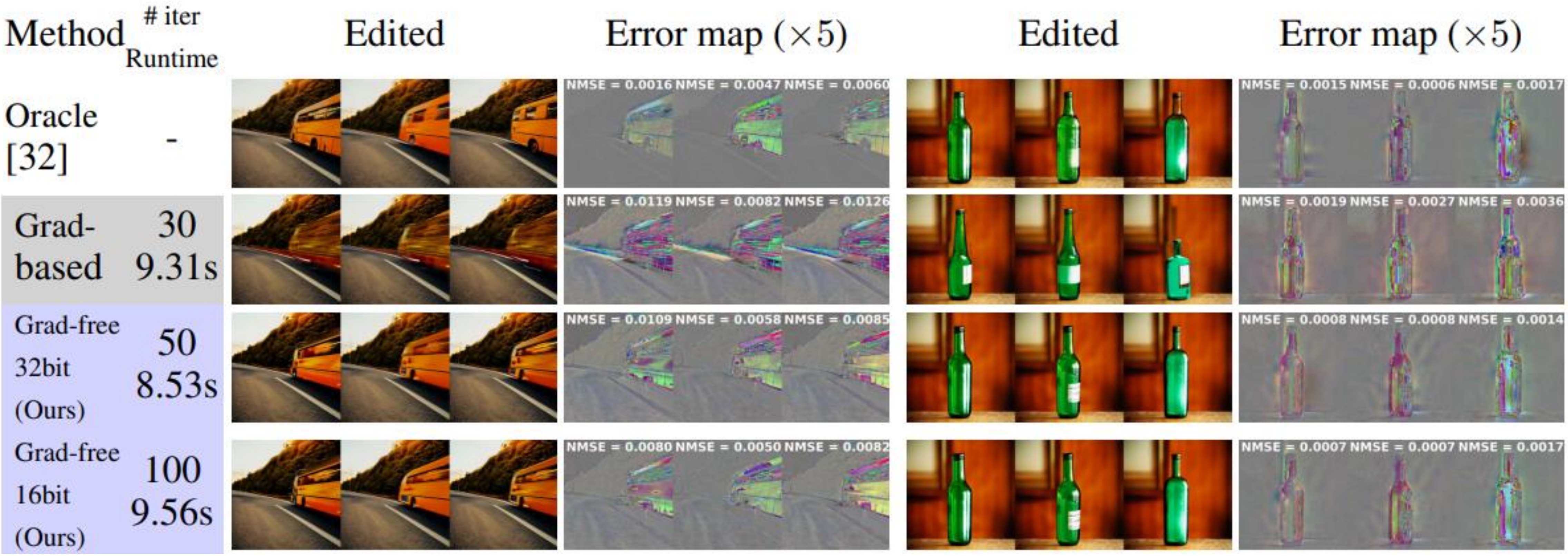


Application: *tree-rings watermarking*

- Invisible, robust watermarking on the initial noise of LDM

LDM		Encoder	Gradient-based [14]	Gradient-free (ours)
SD2.1 [37]	Accuracy	186/300	207/300	202/300
	Peak memory (GB)	5.71	11.4	6.35
	Runtime (s)	5.66	38.0	22.9
InstaFlow [24]	Accuracy	149/300	227/300	227/300
	Peak memory (GB)	2.93	8.84	3.15
	Runtime (s)	3.55	35.9	13.6

Application: Background-preserving editing



Conclusion

- Proposed gradient-free decoder inversion for LDMs with guaranteed convergence, with or without momentum.
- Validated the assumptions and theorems for various LDMs.
- Experimentally showed advantages over gradient-based methods.
  - **Fast**: up to 5× faster
  - **Accurate**: up to 2.3 dB lower in NMSE
  - **Memory-efficient**: up to 89% saved
  - **Precision-flexible**: 16-bit vs 32-bit

See you on: Fri 13 Dec 11 a.m. PST — 2 p.m. PST

Gradient-free Decoder Inversion in Latent Diffusion Models

Project: https://smhongok.github.io/dec-inv.html
arXiv: https://arxiv.org/abs/2409.18442
github: https://github.com/smhongok/dec-inv
mail: smhongok@snu.ac.kr
Lab: https://icl.snu.ac.kr

Project page

NEURAL INFORMATION PROCESSING SYSTEMS

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)] and National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (NRF-2022R1A4A1030579, NRF2022M3C1A309202211). The authors acknowledged the financial support from the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University.