GPSToken: Gaussian Parameterized Spatially-adaptive Tokenization for Image Representation and Generation

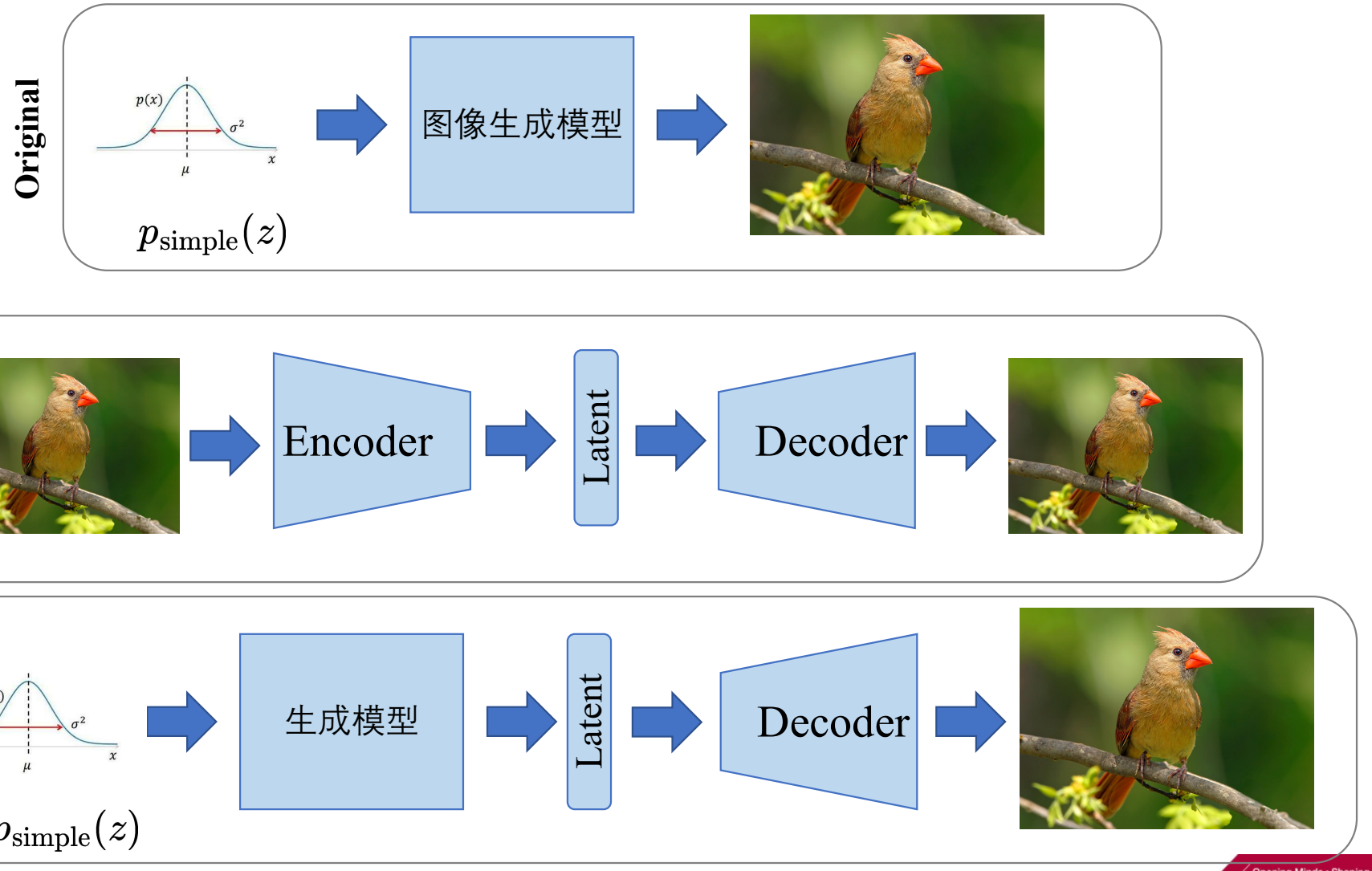
github: https://github.com/xtudbxk/GPSToken
arxiv: https://arxiv.org/abs/2509.01109
nips25: https://neurips.cc/virtual/2025/poster/119331

ZHANG Zhengqiang

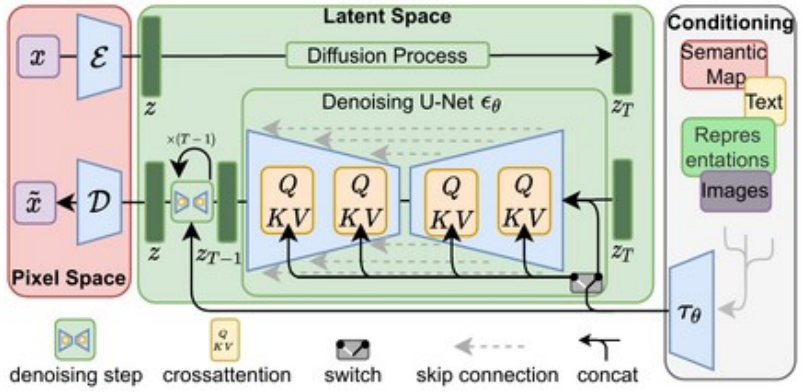
25/09/2025

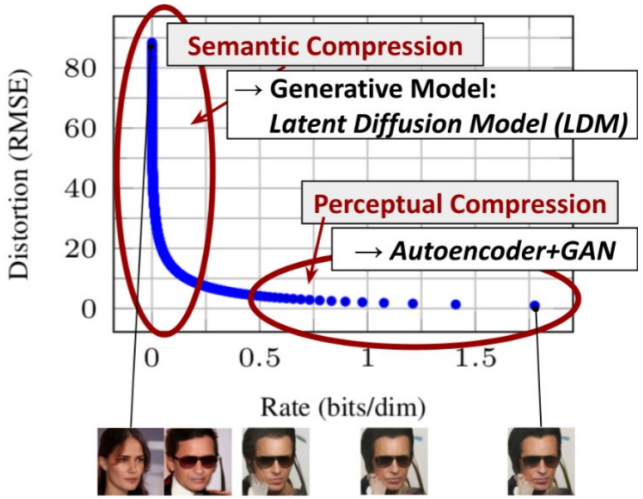
1. Introducation
   1.1 基于Latent的生成模型
   1.2 现有tokenizer的问题
2. Methods
3. Experiments
4. Future Works

– 基于Latent的生成模型



Introducation – 基于Latent的生成模型

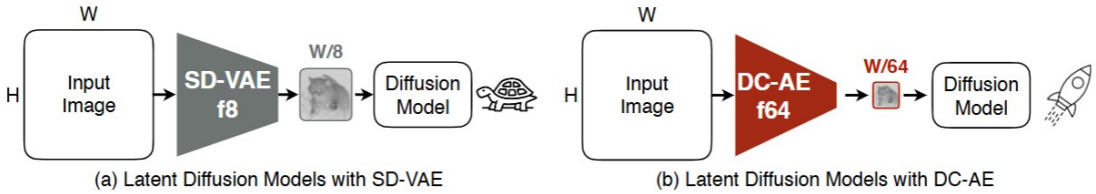




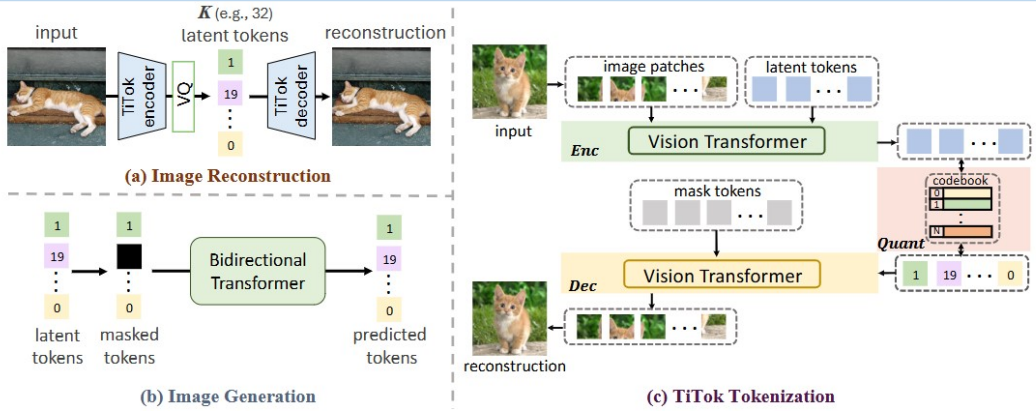
适当压缩仅会去除感知不敏感的细节，保留语义信息

LDM, CVPR2022.

token数量少 -> 优化难度低、Inference速度快

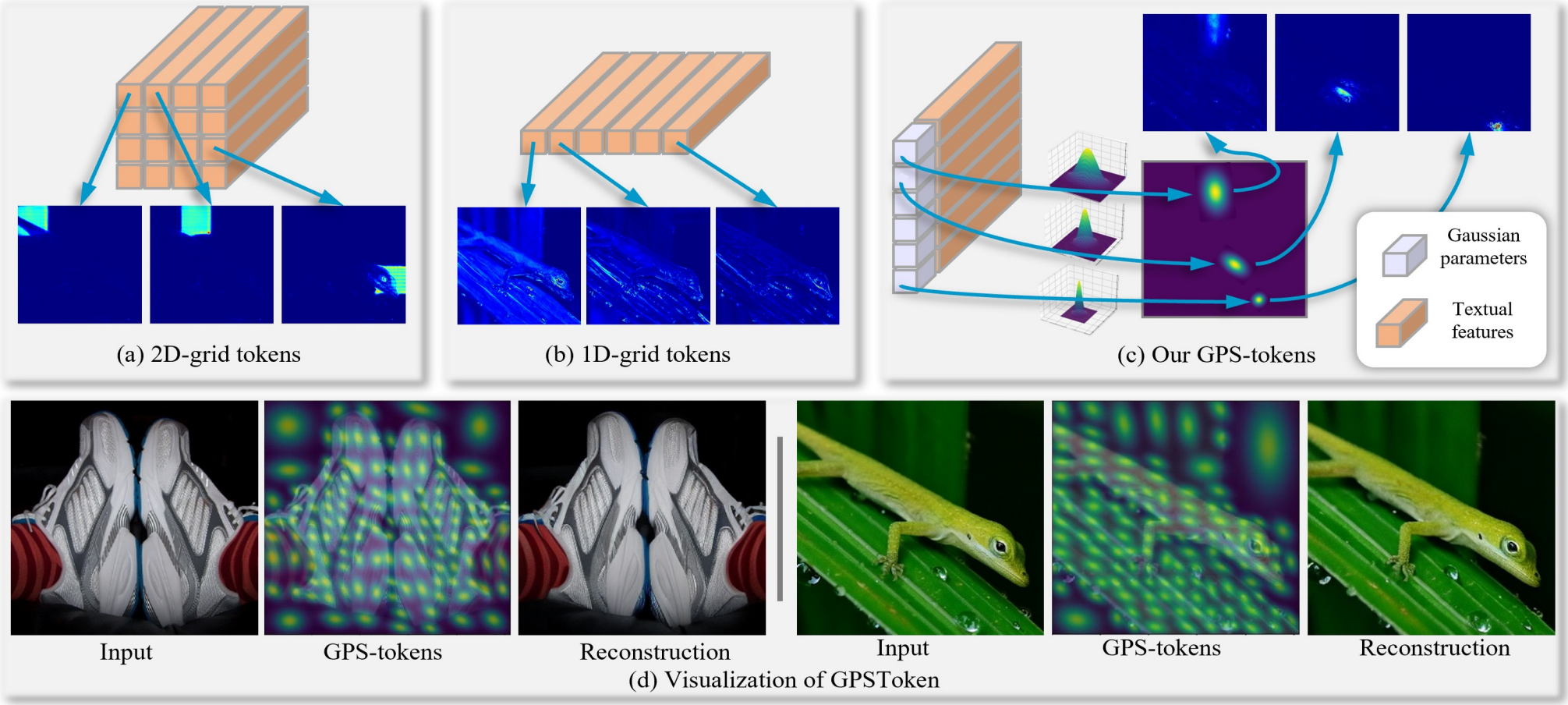


DCAE, ICLR2025.



TiTok, NIPS2024.

Introducation – 现有tokenizer的问题



The uniform 2D/1D grid tokenization are inflexible to represent regions with varying shapes and textures and at different locations, limiting their efficacy of feature representation.

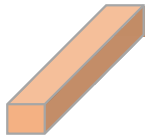
1. Introducation
2. Methods
   - 2.1 Gaussian-parametered Tokens
   - 2.2 Spatially-adapative Token Initialization and Refinement
   - 2.3 Two-stage Image Generator of GPSToken
3. Experiments
4. Future Works

Methods – Gaussian-parametered Tokens

Gaussian-parametered token for each region:

$$\mathbf{z}_i = \{\mathbf{g}_i, \mathbf{f}_i\}$$

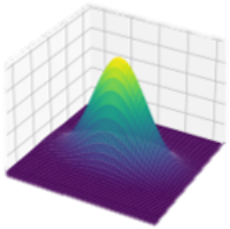
$\mathbf{g}_i = \{\sigma_x^{(i)}, \sigma_y^{(i)}, \rho^{(i)}, \mu_x^{(i)}, \mu_y^{(i)}\}$ - Gaussian function parameters

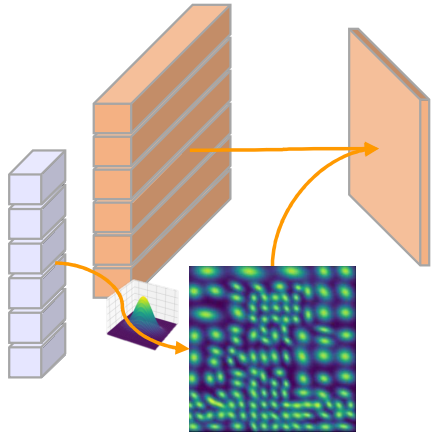


$\mathbf{f}_i$ - regional textual features

$$\mathbf{g}(x,y) = \begin{cases} \hat{p}(x,y), & \text{if } |x-\mu_x| \le s\sigma_x \text{ and } |y-\mu_y| \le s\sigma_y, \\ 0, & \text{otherwise,} \end{cases}$$

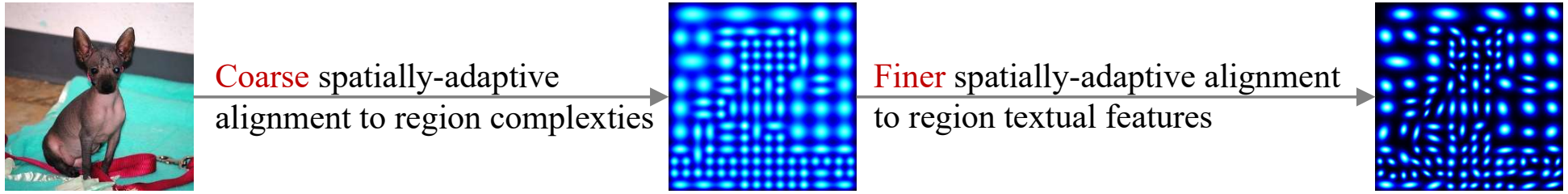
$$p(x,y) = \frac{\hat{p}(x,y)}{Z} = \frac{1}{Z}\exp\left(-\frac{1}{2(1-\rho^2)}\left(\frac{(x-\mu_x)^2}{\sigma_x^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \frac{(y-\mu_y)^2}{\sigma_y^2}\right)\right),$$





$$R(x,y,k) = \sum_{i=0}^{l-1} r_i(x,y,k) = \sum_{i=0}^{l-1} \mathbf{g}_i(x,y) \times \mathbf{f}_i[k].$$

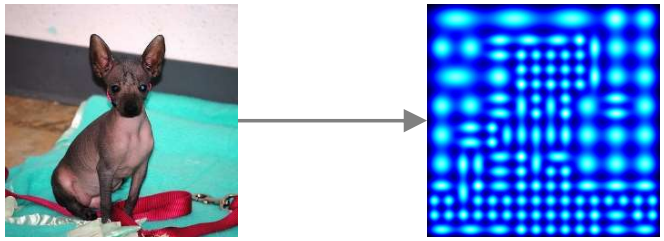
Methods – Spatially-adapative Token Initialization and Refinement



Spatially-adapative Token Initialization

Spatially-adapative Token Refinement

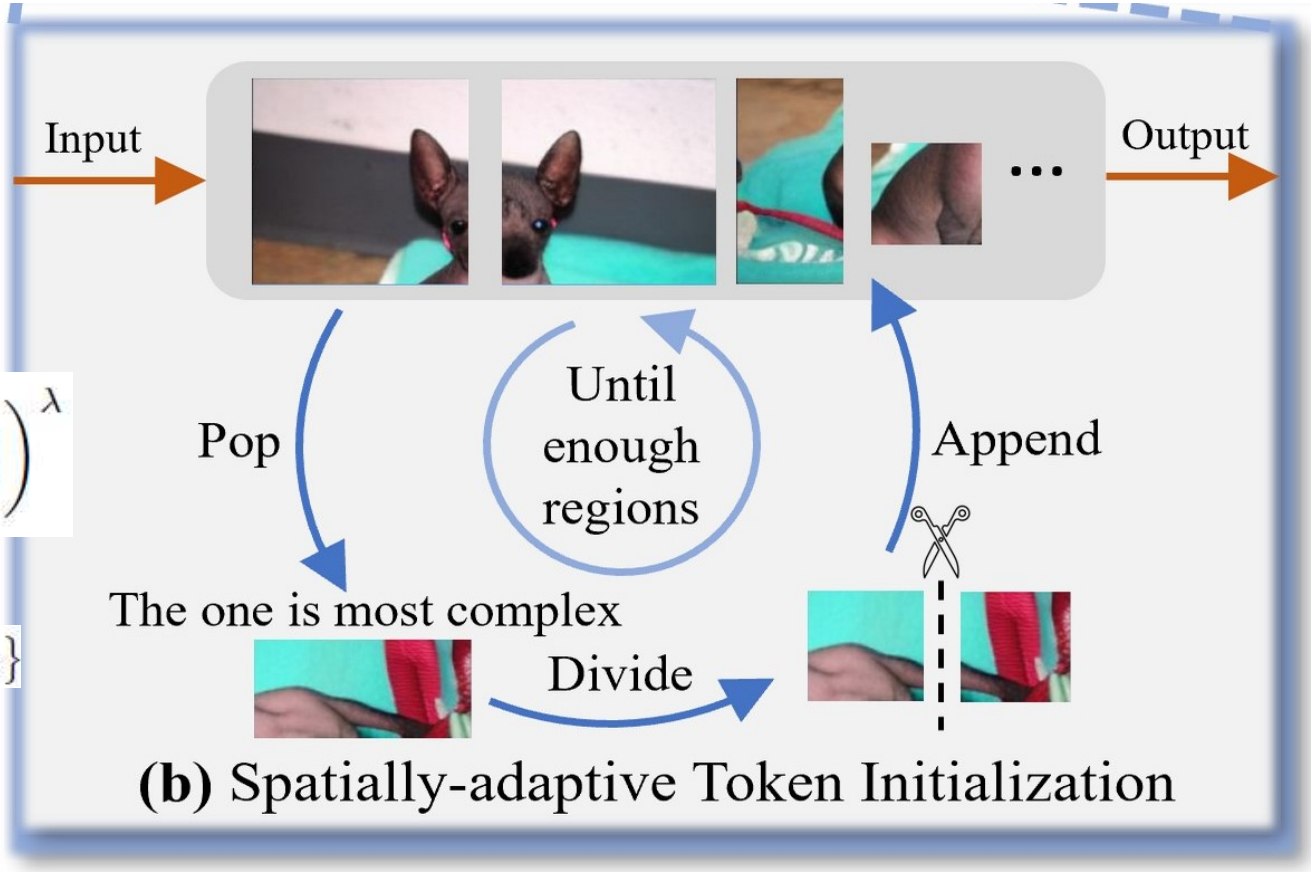
Methods – Spatially-adapative Token Initialization



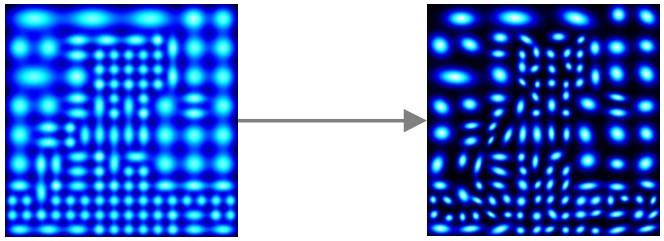
Segment image to regions according to the complexities

$$m = hw \times H^{\lambda} = hw \times \left(-\sum_{i=1}^{512} q_i \log(q_i)\right)^{\lambda}$$

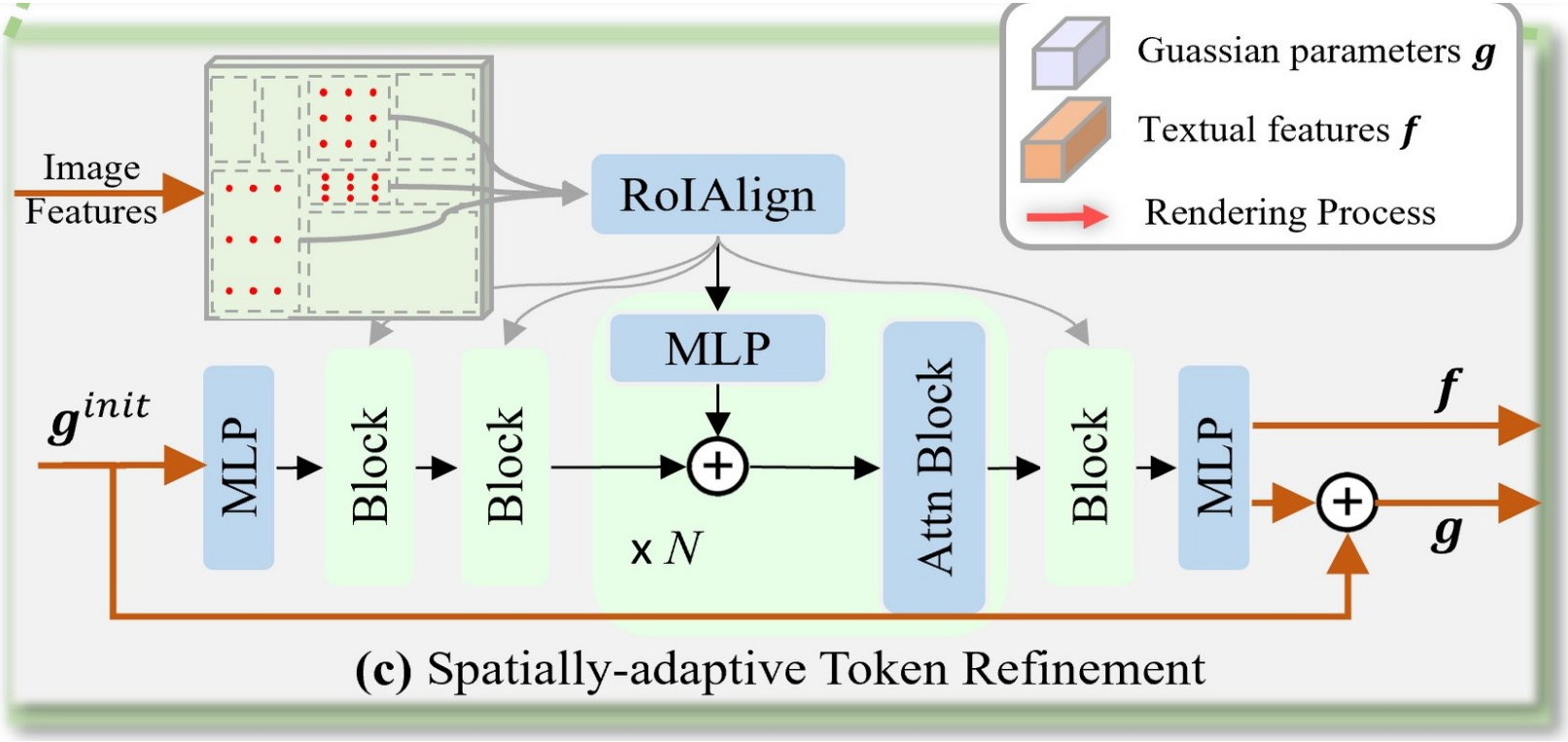
$$\mathbf{g}_i^{init} = \{\sigma_x^{(i)}, \sigma_y^{(i)}, \rho^{(i)}, \mu_x^{(i)}, \mu_y^{(i)}\} = \{\tfrac{w_i}{6}, \tfrac{h_i}{6}, 0, x_i, y_i\}$$



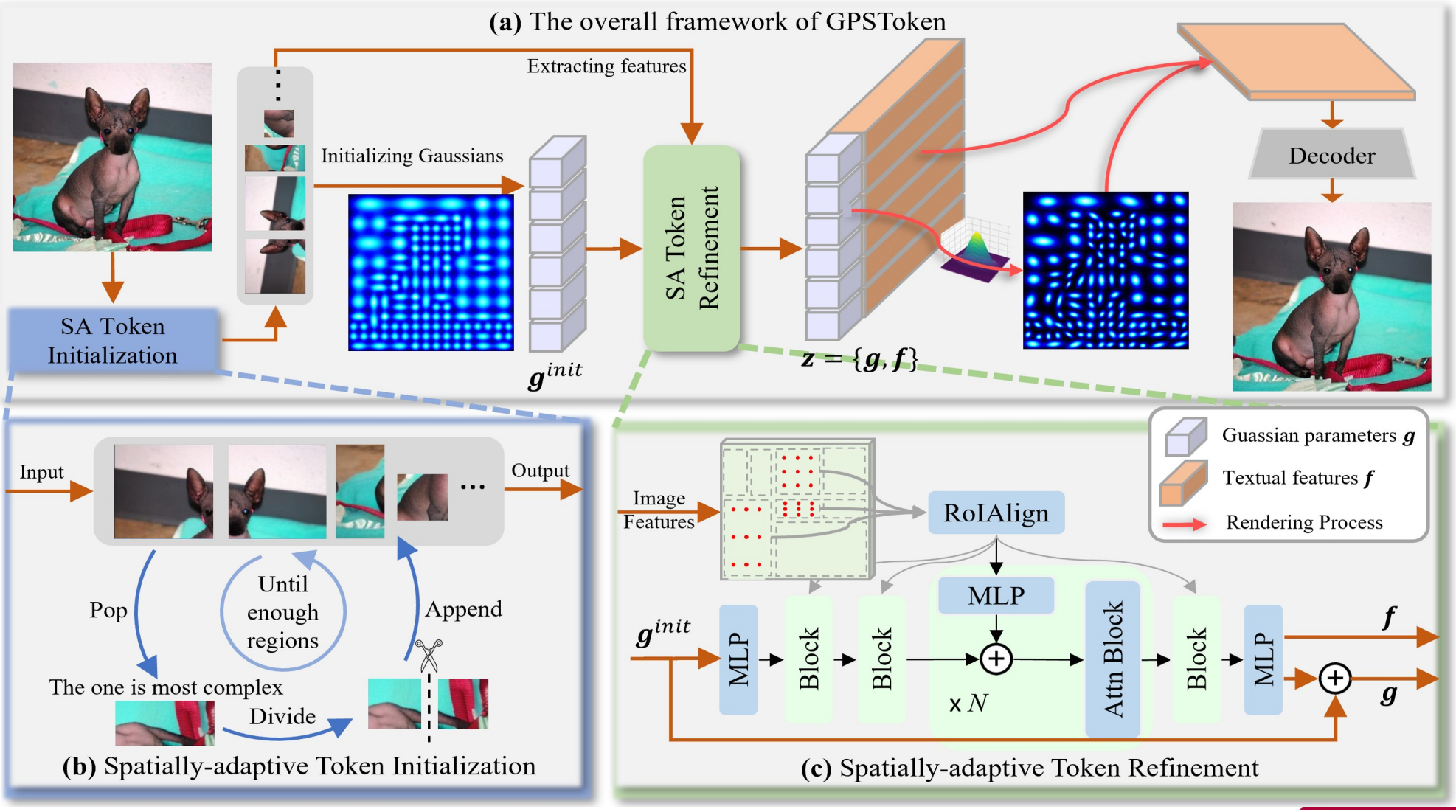
Methods - Spatially-adapative Token Refinement



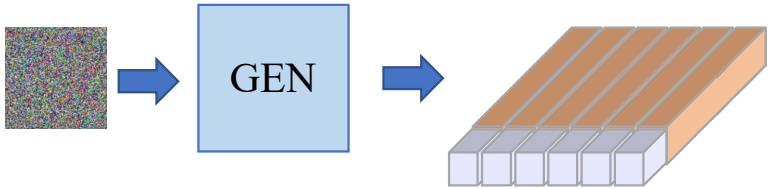
further align the Gaussians with local textual features



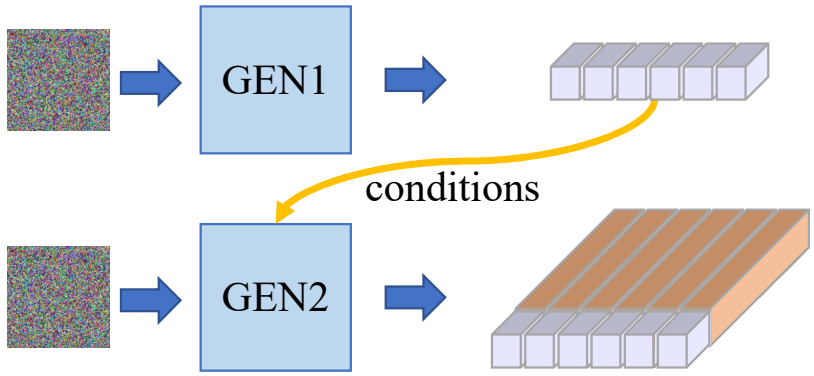
Methods – Spatially-adapative Token Initialization and Refinement



Methods – Two-stage Image Generator of GPSToken

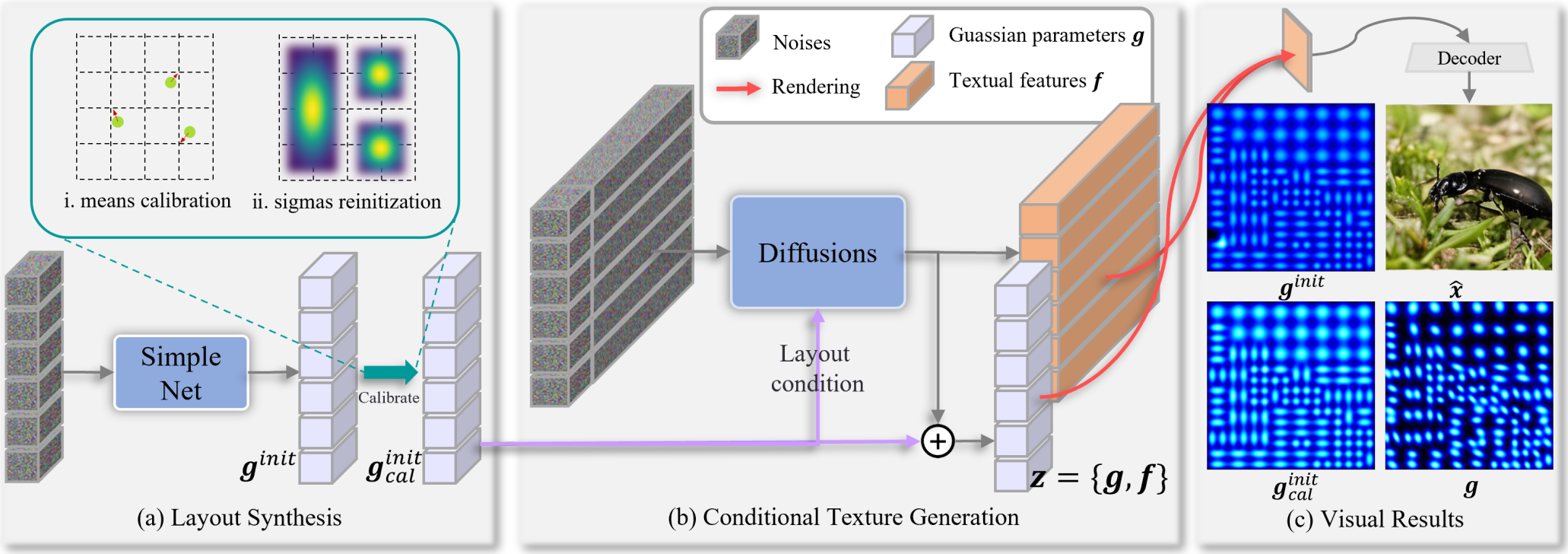


One-stage generation framework



Two-stage generation framework

Methods - Two-stage Image Generator of GPSToken



1. Introducation

2. Methods

3. Experiments

   3.1 experiments for reconstruction task

   3.2 experiments for generative task

4. Future Works

Experiments - Comparison results on reconstruction task

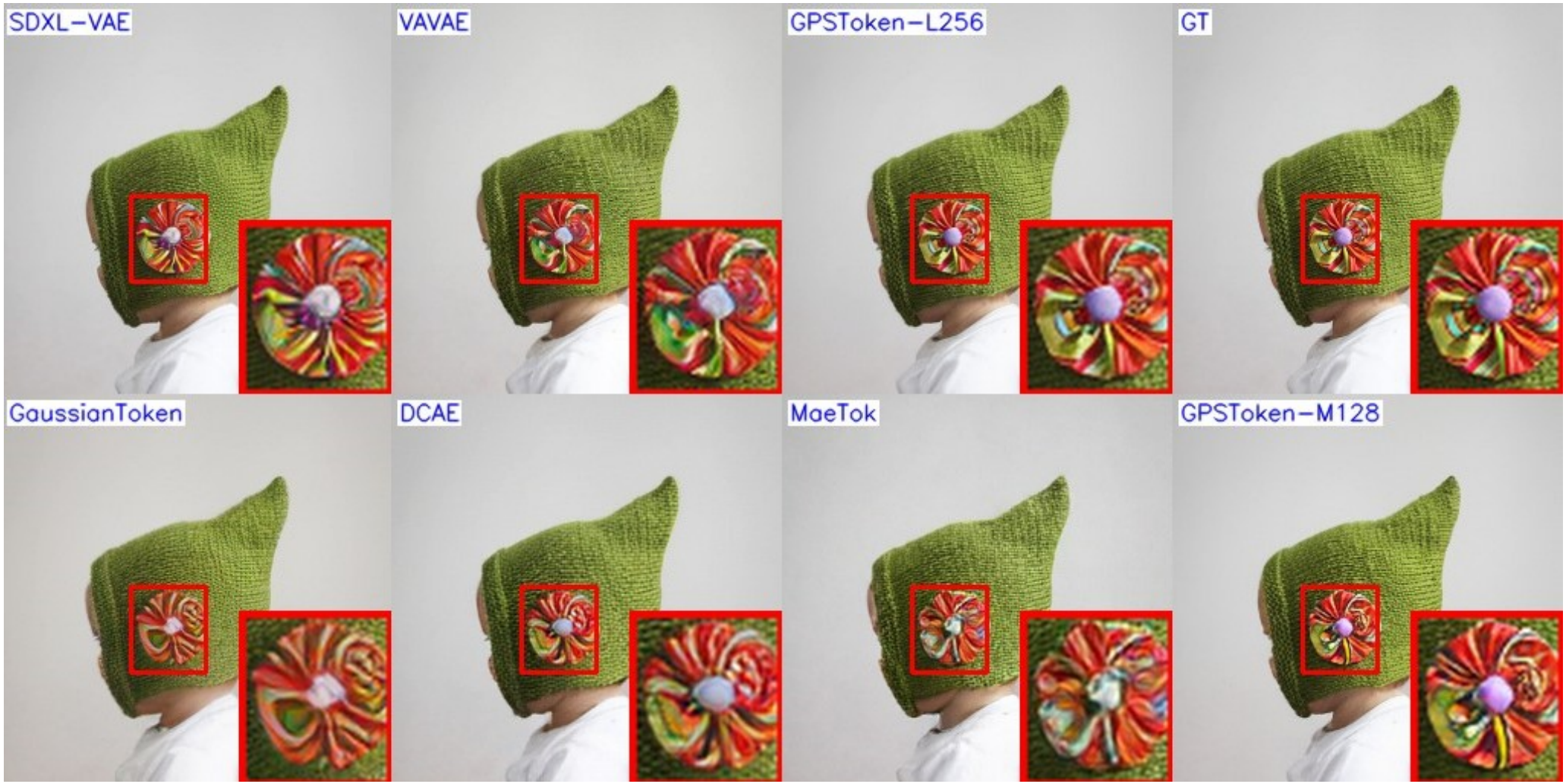
Table 1: Comparisons of 256 × 256 reconstruction task on Imagenet val set. The top 3 methods *trained only with ImageNet* are highlighted in red, blue and green. Note that "SDXL-VAE" is trained with a rich amount of additional data other than Imagenet.

Method	Tokens	Params (M)	sample-level			distribution-level			
			PSNR ↑	SSIM ↑	LPIPS ↓	rec. FID ↓	rec. sFID ↓	FID ↓	sFID ↓
2D Tokenization									
SDXL-VAE [21]	32×32	83.6	25.55	0.727	0.066	0.73	2.42	2.35	3.89
GaussianToken [14]	32×32	130.6	22.40	0.597	0.112	1.70	4.62	3.63	4.71
VQVAE-f16 [9]	16×16	89.6	19.41	0.476	0.191	8.01	9.64	10.74	7.38
MaskGIT-VAE [2]	16×16	54.5	18.11	0.427	0.202	3.79	5.81	5.19	4.56
VAVAE [30]	16×16	69.8	25.76	0.742	0.050	0.27	1.72	1.74	3.91
DCAE [5]	8×8	323.4	23.62	0.644	0.092	0.98	4.82	2.59	5.02
1D Tokenization									
SoftVQ [4]	64	173.6	21.93	0.568	0.115	0.92	4.52	2.51	4.21
TiTok-B64 [31]	64	204.8	17.01	0.390	0.263	1.75	4.51	2.50	4.21
TiTok-S128 [31]	128	83.7	17.66	0.413	0.220	1.73	7.25	3.25	5.52
MAETok [3]	128	173.9	23.25	0.626	0.096	0.65	3.87	2.01	4.39
FlexTok [1]	256	949.7	17.69	0.475	0.257	4.02	8.00	4.88	6.12
One-D-Piece [18]	256	83.9	17.74	0.420	0.210	1.54	6.96	2.93	5.36
MaskBit [29]	256	54.5	21.07	0.539	0.142	1.29	4.72	3.08	4.09
GPSToken									
S64	64	158.5	23.07	0.605	0.101	0.85	4.60	2.39	4.67
M128	128	127.8	24.06	0.657	0.080	0.65	3.28	2.18	3.96
L256	256	128.7	28.81	0.809	0.043	0.22	1.31	1.65	3.77

Experiments - Comparison results on reconstruction task

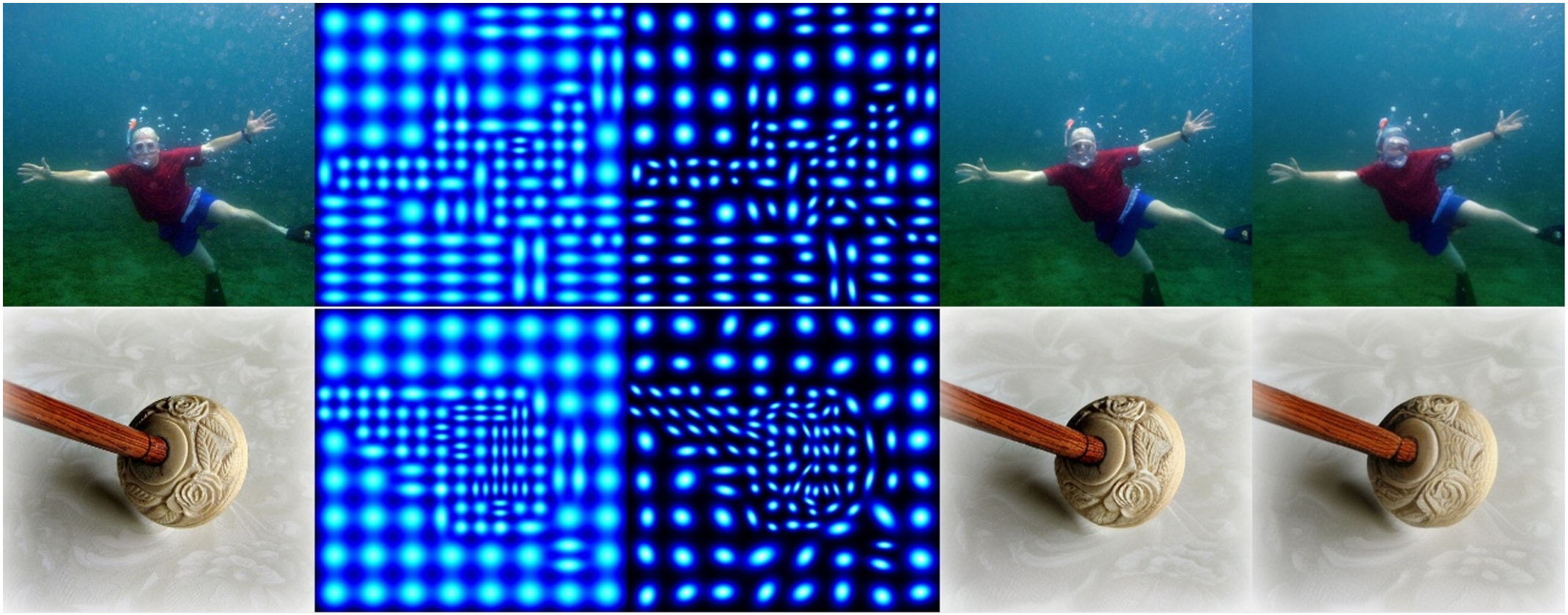
Experiments - Comparison results on reconstruction task

Experiments - Comparison results on reconstruction task



Experiments - Comparison results on reconstruction task

Experiments – Visualization of GPSToken

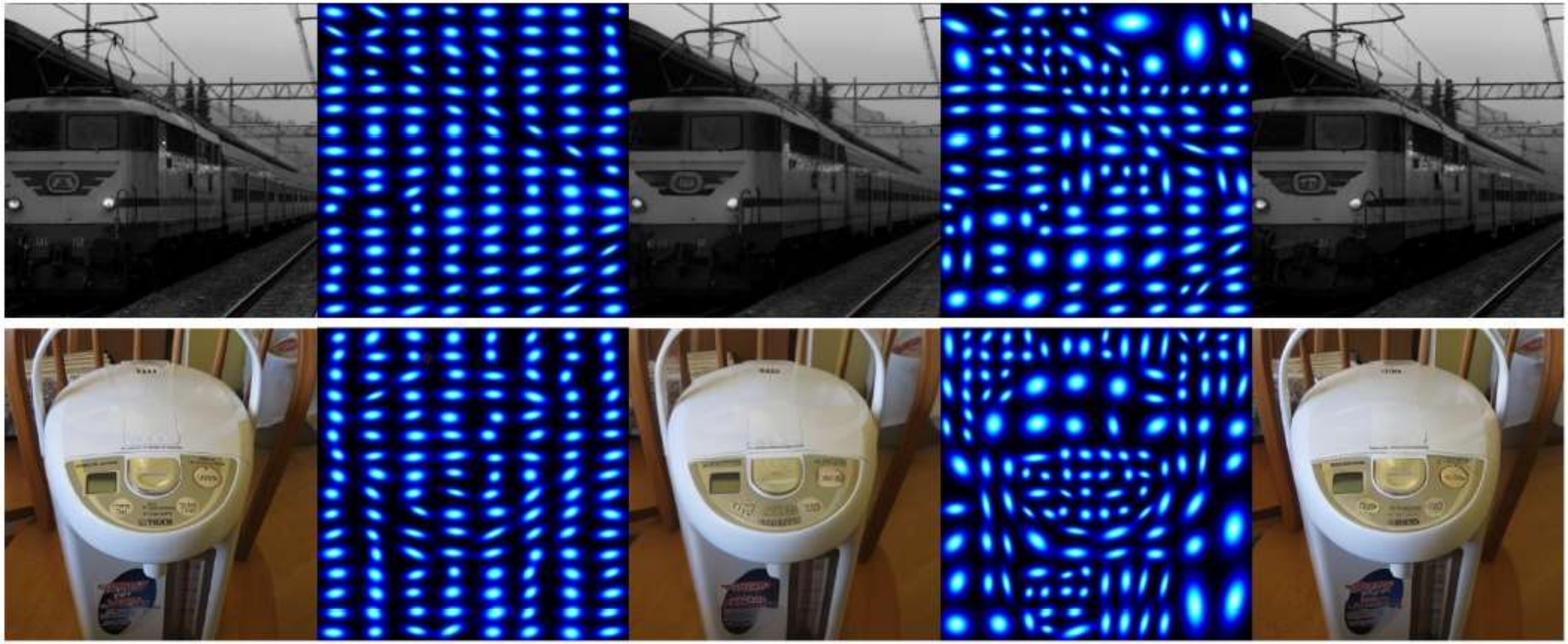


(a) inp (b) $\boldsymbol{g}^{init}$ (c) refined $\boldsymbol{g}$ (d) GPSToken (e) GPSToken w/o SA

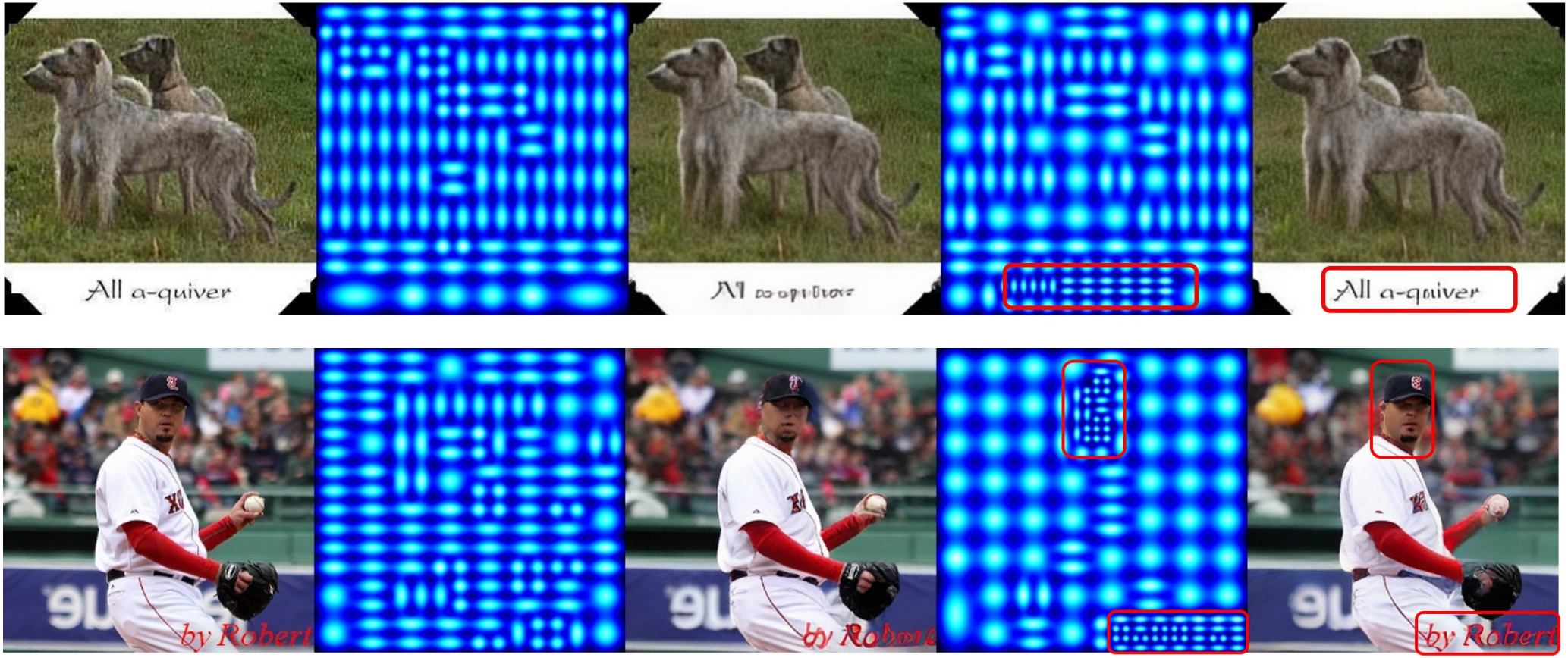
Table 1: Ablation studies of our spatial adaptivity designs on the 256 × 256 reconstruction task. ✓indicates that the component is used. "Init." and "Refine." denote the spatially-adaptive token initialization and spatially-adaptive token refinement, respectively.

Method	Components		sample-level			distribution-level			
	Init.	Refine.	PSNR ↑	SSIM ↑	LPIPS ↓	rec. FID ↓	rec. sFID ↓	FID ↓	sFID ↓
baseline			23.52	0.638	0.110	1.02	4.07	2.59	4.34
baseline+		✓	24.00	0.654	0.100	0.81	3.59	2.37	4.31
GPSToken	✓	✓	24.06	0.657	0.080	0.65	3.28	2.18	3.96

Experiments – Ablation Studies of GPSToken

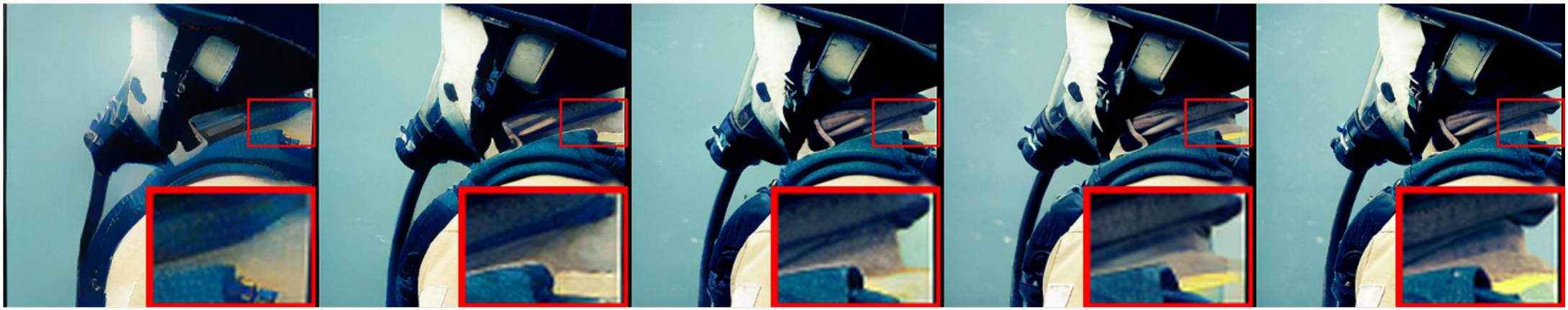


Experiments – User-Controllable Adjustment of Gaussian Parameters



(a) inp (b) $\boldsymbol{g}^{init}$ (c) GPSToken (d) User adjusted $\boldsymbol{g}^{init}$ (e) GPSToken w/ adjusted $\boldsymbol{g}^{init}$

Experiments - Var length of tokens



Experiments – Works at higher resolution without retraining

Method	Tokens	PSNR ↑	SSIM ↑	LPIPS ↓	rFID ↓	rec. sFID ↓
512×512						
SDXL-VAE	64×64	28.42	0.817	0.059	0.271	1.36
VQVAE-f16	32×32	21.83	0.604	0.172	2.29	7.95
GPSToken-M128	512	26.74	0.764	0.073	0.367	1.93
GPSToken-L256	1024	32.00	0.887	0.039	0.175	0.699
1024×1024						
SDXL-VAE	128×128	33.27	0.909	0.057	0.113	0.561
VQVAE-f16	64×64	25.41	0.744	0.169	1.40	4.98
GPSToken-M128	2048	31.22	0.873	0.072	0.236	1.24
GPSToken-L256	4096	37.71	0.955	0.031	0.055	0.276

Experiments – Comparison results on generative task

Table 2: Comparisons on 256 × 256 class-conditional image generation. The top 2 methods are highlighted in red and blue. "+" indicates the baseline.

Method	Tokenizer		Generator			
	Params (M)	Tokens	Params (M)	FID ↓	sFID ↓	IS ↑
Auto-regressive Models						
MaskGIT [2]	54.5	16×16	227	6.18	-	182.1
FlexTok [1]	-	256	1,330	2.02	-	-
TiTok-S128 [31]	83.7	128	287	1.97	-	281.8
TiTok-B64 [31]	204.8	64	177	2.77	-	199.8
SoftVQ [4]	173.6	64	675	1.78	-	279.0
Diffusion-based Models						
ADM [6]	-	-	23.24	3.94	6.14	215.8
One-D-Piece [18]	83.9	256	-	2.35	-	224.4
DiT-XL/2 [20]	83.6	32×32	675	2.27	4.60	278.2
SiT-XL/2^{+} [17]	83.6	32×32	675	2.06	4.49	277.5
REPA [32]	83.6	32×32	675	1.79	4.51	276.8
D^{2}iT [13]	-	>256	687	1.73	-	307.9
MAETok [3]	173.9	128	675	1.67	-	311.2
Ours (one-stage)	127.8	128	675	2.13	6.40	258.2
Ours (two-stage)	127.8	128	33+675	1.64	4.41	316.0

Experiments - Ablation Studies on Two-stage Generator

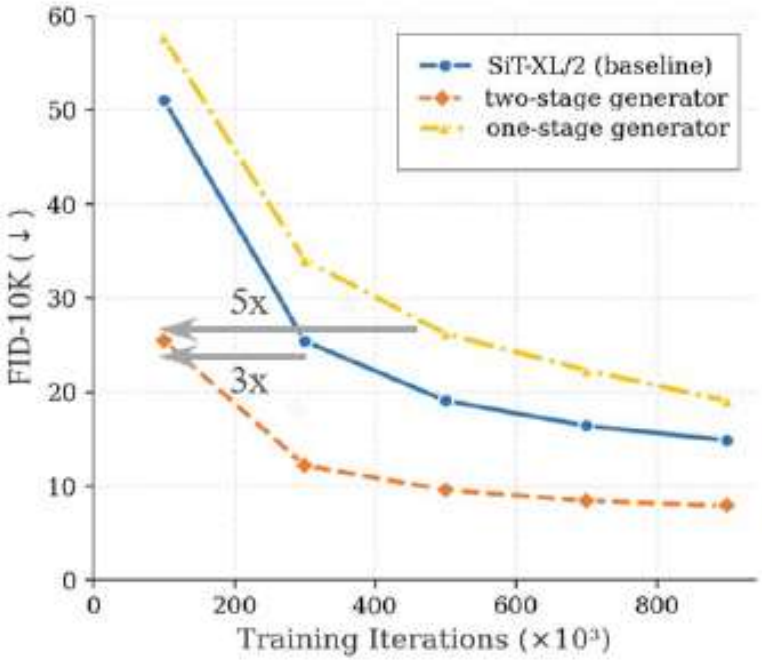


Figure 7: FID-10K training curves.

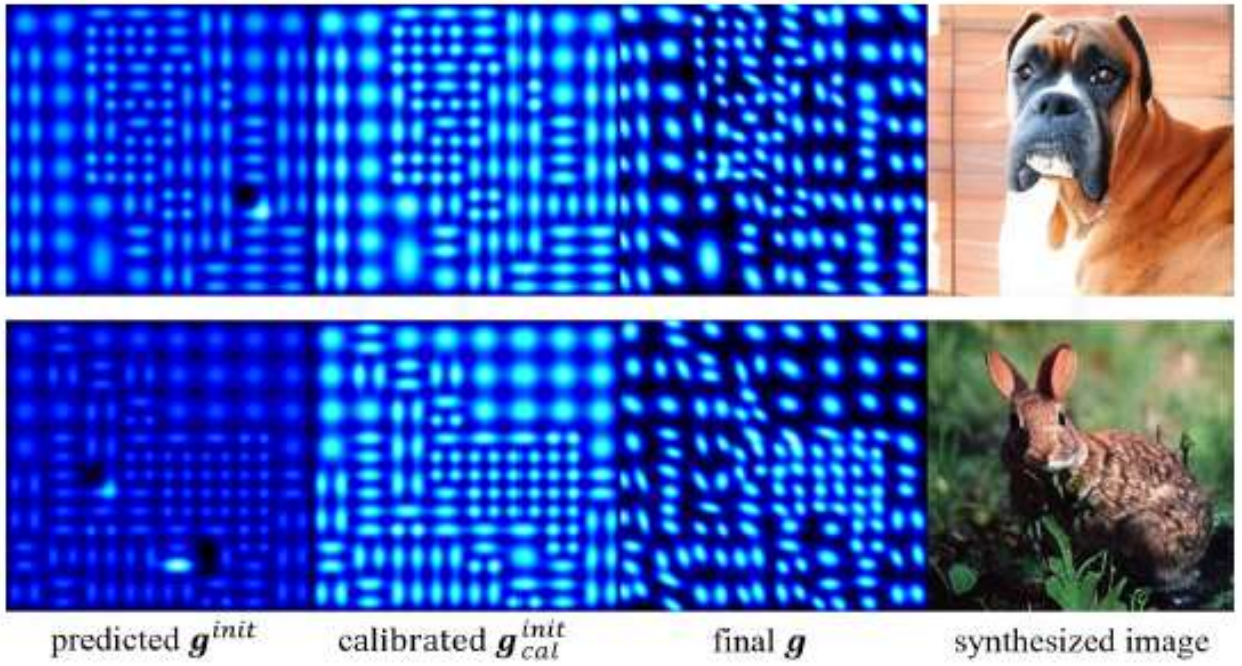


Figure 8: Illustration of two-stage generation pipeline.

Experiments - Visual Results of Two-stage Generator

Experiments - Visual Results of Two-stage Generator

Experiments - Visual Results of Two-stage Generator

1. Introducation
2. Methods
3. Experiments
4. Future Works

Future Works –

一、GPSToken改进

1. 去除启发式初始化算法？

2. 以语义而非纹理复杂度划分区域？

二、GPSToken应用探索

1. high-level分类任务

2. pixel-level任务，如图像复原、图像分割

Thanks ~