PICD: Versatile Perceptual Image Compression with Diffusion Rendering

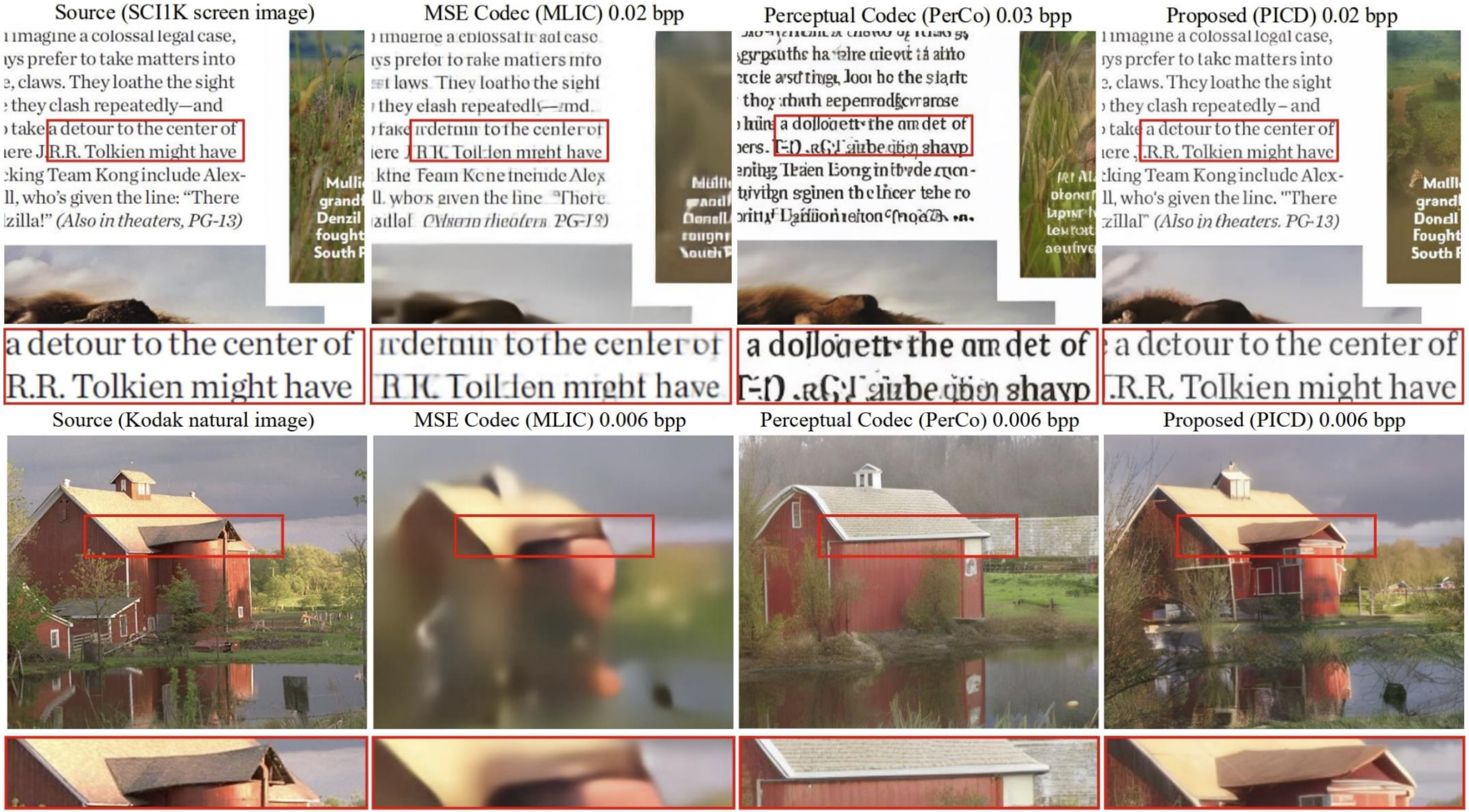
Tongda Xu (speaker), Jiahao Li, Bin Li, Yan Wang, Ya-Qin Zhang & Yan Lu

Tsinghua University & Microsoft Research Asia

@CVPR 2025

Disclaimer: This content is solely the responsibility of the speaker and does not necessarily represent the official views of any affiliations.

- TLDR: A perceptual codec that works well for both screen and natural images



Source (SCI1K screen image)

MSE Codec (MLIC) 0.02 bpp

Perceptual Codec (PerCo) 0.03 bpp

Proposed (PICD) 0.02 bpp

Source (Kodak natural image)

MSE Codec (MLIC) 0.006 bpp

Perceptual Codec (PerCo) 0.006 bpp

Proposed (PICD) 0.006 bpp

Lay of the Land

- Optimal coding for perceptual and screen/machine vision
- Naïve implementation of optimal coding framework
- Enhanced condition of three level
  - Domain level, Adaptor level, Instance level
- Experimental Results
  - Ablation Study
  - Main Results

Optimal Coding for Screen/Machine Vision

- Notation:
    - Denote source image as X
    - Denote code for X as Y = f(X), f(.) is encoder
    - Denote reconstruction image as X’ = g(Y), g(.) is decoder
- Setting of Screen/Machine Vision:
    - Have some information Z, that can be **determined** from X
        - We are really sure about Z given X, such that Z = h(X) for some function h(.)
        - No blurry results, no uncertainty!
    - We want to preserve information Z, even from X’

Optimal Coding for Screen/Machine Vision

- (One of) optimal coding for screen / machine vision:
  - Encode Z losslessly, encode Y with Z as condition
  - Decode Z losslessly, decode Y' with Z as condition
  - … but this is too naïve? Don't we need joint coding / loss function?
- This is already optimal, as:
  - If we want to preserve Z fully, it is determined by Y, H(Z|Y) = 0
  - Then H(Z,Y) = H(Y) + H(Z|Y) = **H(Y) = H(Y|Z) + H(Z)**
  - To **encode Y, is the same as encode Z, then Y|Z**
  - Formally in [Conditional Perceptual Quality Preserving Image Compression]
  - Joint coding / loss function just squeeze all Z into Y, which is the same

Optimal Coding for Perceptual + Screen

- We say the perceptual quality is optimal, if:
    - P(X) = P(X'), but why?
    - We follow [Perception Distortion Trade-off], the human's successful rate to distinguish source X, compressed X' is 0.5 + 0.5 * DTV(p(X), p(X'))
    - The optimal perceptual quality is when people can not tell difference between X, X', which means DTV(p(X), p(X')) = 0, p(X) = p(X')
- How to achieve?
    - Train a conditional generative model as decoder, such that:
    - X' ~ g(Y, Z) = p(X|Y, Z), this leads to p(X) = p(X')

Naïve Implementation

- A quite naïve implementation of the optimal coding for perceptual + screen framework:
  - Encode 1: we use Teasseract OCR to extract Z from X, encode the coordinate exp-golomb, encode text with cmix.
  - Encode 2: we train a MLIC [Multi-Reference Entropy Model for Learned Image Compression] with text Z as condition, to optimize R-D performance on screen images. We use this model to encode Y=f(X,Z)
  - Decode 1: we decode text information Z
  - Decode 2: we decode Y, then X’ = g(Y,Z)
  - Decode 3: we train a **conditional diffusion model, with Z, Y as condition, to learning posterior p(X|Z,Y), this step is named diffusion rendering**.

Naïve Implementation

- Diagram of diffusion rendering. Very straightforward!
  - Glyph is just a rendering of text location and content, using PIL.drawtext.
  - Glyph helps diffusion model learns better [Glyph Conditional Control for Visual Text Generation]

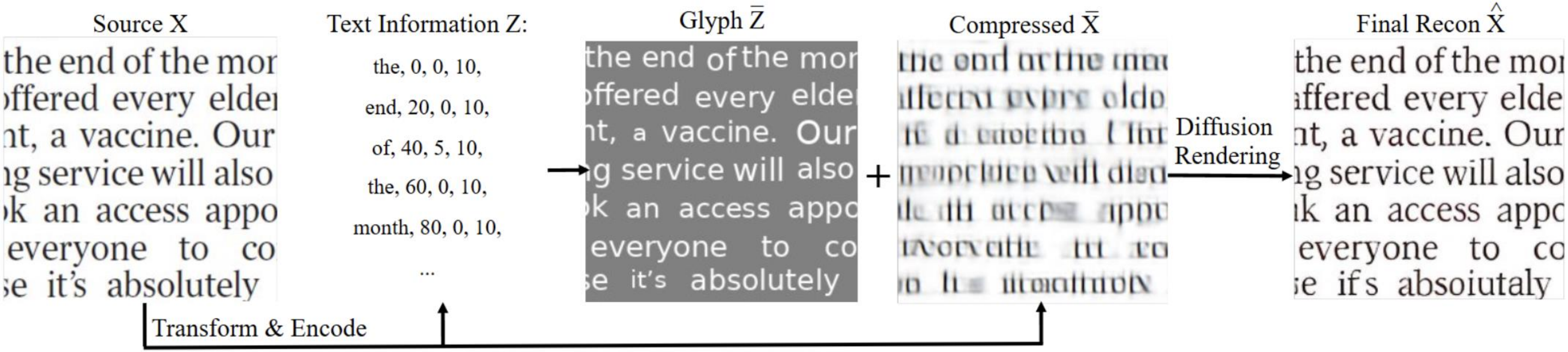


Figure 4. An example of the PICD pipeline. PICD first extracts and encodes text information Z from source X. Then, PICD converts text information into text glyph $\bar{Z}$. Finally, PICD renders glyph $\bar{Z}$ with a compressed image $\bar{X}$ into reconstruction $\hat{X}$ using diffusion.

Three Level Improvements

- Naïve Implementation does … NOT work empirically
  - Severe color drifting from the source image
- Three level improvements:
  - Domain, adaptor and instance
- Domain level improvement:
  - Train a LoRA to adapt the original stable diffusion model to screen contents
  - Construct prompt such as “a screenshot image showing …”, … is the OCRed content

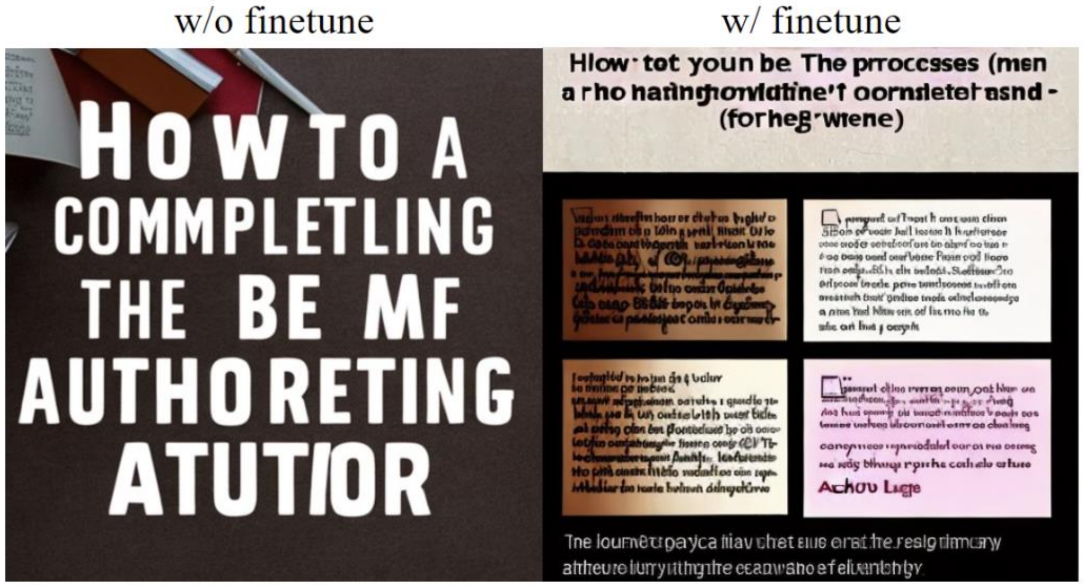

w/o finetune
w/ finetune
HOW TO A COMMPLETLING THE BE MF AUTHO RETING ATUTIOR
Hlow· tot youn be The prroccsses (men a rho harlngnowidtine't oornsletet asnd - (forheg·wiene)
Achou Lige

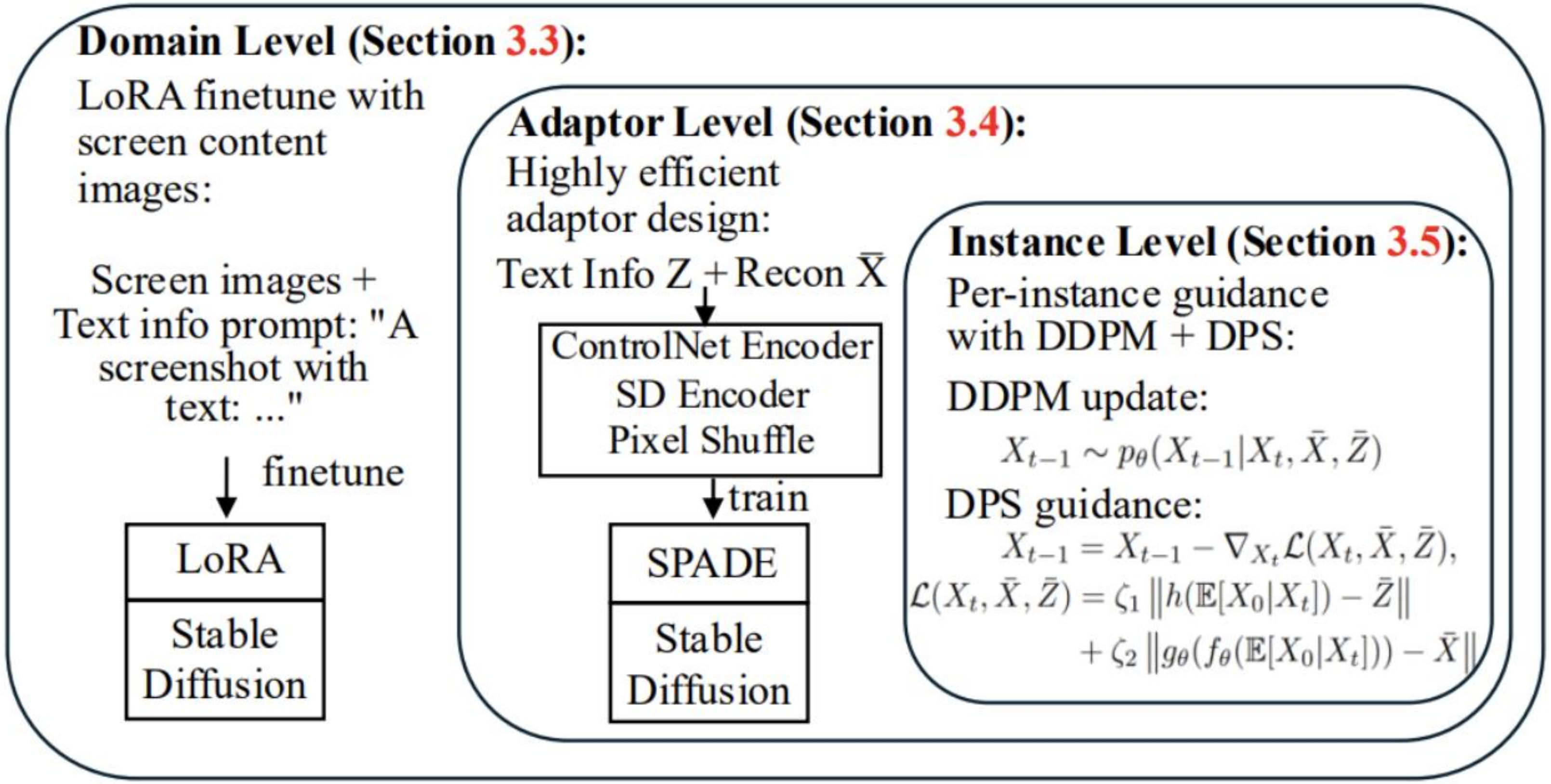
Figure 6. Example of Stable Diffusion generation with and without finetuning. The prompt is ”screenshot with text: How to be an author of a paper. The process of completing a paper”.

Three Level Improvements

- Adaptor level improvement:
  - A better adaptor than ControlNet [Adding Conditional Control to Text-to-Image Diffusion Models]
  - We use an variant of StableSR [Exploiting Diffusion Prior for Real-World Image Super-Resolution] for the X’ conditional branch, while we stick to a copy of ControlNet to process glyph
- Instance level improvement:
  - During inference, we combine the instance constraint in [Idempotence and Perceptual Image Compression]
  - With a new constraint using OCR loss as operator described in [Diffusion Posterior Sampling for General Noisy Inverse Problems]

Three Level Improvements

- Summary of three-level improvements
- Simplify into natural images:
    - No domain level, no OCR guidance, no glyph input using text information



Domain Level (Section 3.3):
LoRA finetune with screen content images:

Screen images + Text info prompt: "A screenshot with text: ..."

Adaptor Level (Section 3.4):
Highly efficient adaptor design:

Text Info Z + Recon $\bar{X}$

Instance Level (Section 3.5):
Per-instance guidance with DDPM + DPS:

DDPM update:

$$X_{t-1} \sim p_\theta(X_{t-1}|X_t, \bar{X}, \bar{Z})$$

DPS guidance:

$$X_{t-1} = X_{t-1} - \nabla_{X_t}\mathcal{L}(X_t, \bar{X}, \bar{Z}),$$

$$\mathcal{L}(X_t, \bar{X}, \bar{Z}) = \zeta_1 \left\|h(\mathbb{E}[X_0|X_t]) - \bar{Z}\right\| + \zeta_2 \left\|g_\theta(f_\theta(\mathbb{E}[X_0|X_t])) - \bar{X}\right\|$$

Experimental Results: Ablation Studies

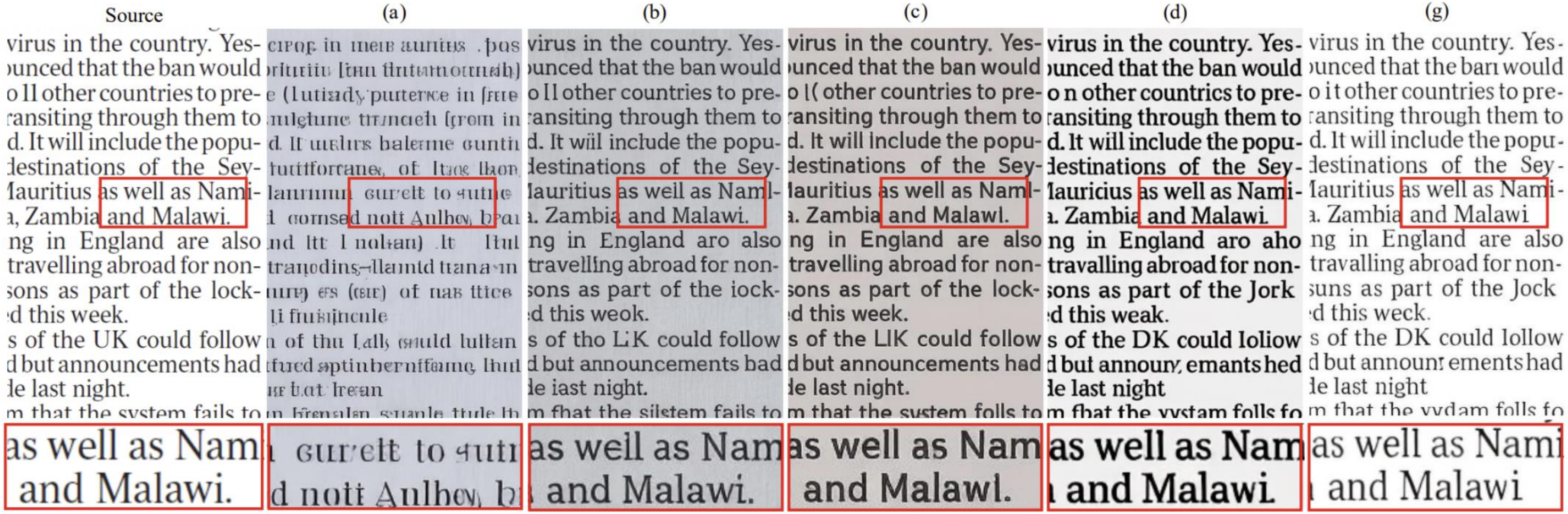


Figure 5. Ablation studies on different components of diffusion rendering.

ID	Glyph (Sec 3.2)	Domain Level (Sec 3.3)	Adaptor Level (Sec 3.4)			Instance Level (Sec 3.5)	Text Acc↑	PSNR↑	FID↓	CLIP↑	LPIPS↓
			ControlNet [55]	StableSR [45]	Proposed						
(a)			✓				0.3468	19.10	45.83	0.8209	0.1694
(b)	✓		✓				0.4404	18.84	45.35	0.8617	0.1646
(c)	✓			✓			0.3934	20.56	49.76	0.8850	0.1344
(d)	✓				✓		0.4081	19.88	37.90	0.8922	0.1376
(e)	✓		✓			✓	0.4446	23.30	39.81	0.8917	0.1225
(f)	✓				✓	✓	0.4445	**23.70**	35.54	0.9059	0.1172
(g)	✓	✓			✓	✓	**0.4568**	23.67	**34.77**	**0.9082**	**0.1168**

Experimental Results

- Comparison to other text condition / text enhancing approach:
  - Our approach is the only one that achieve both high text accuracy and perceptual quality

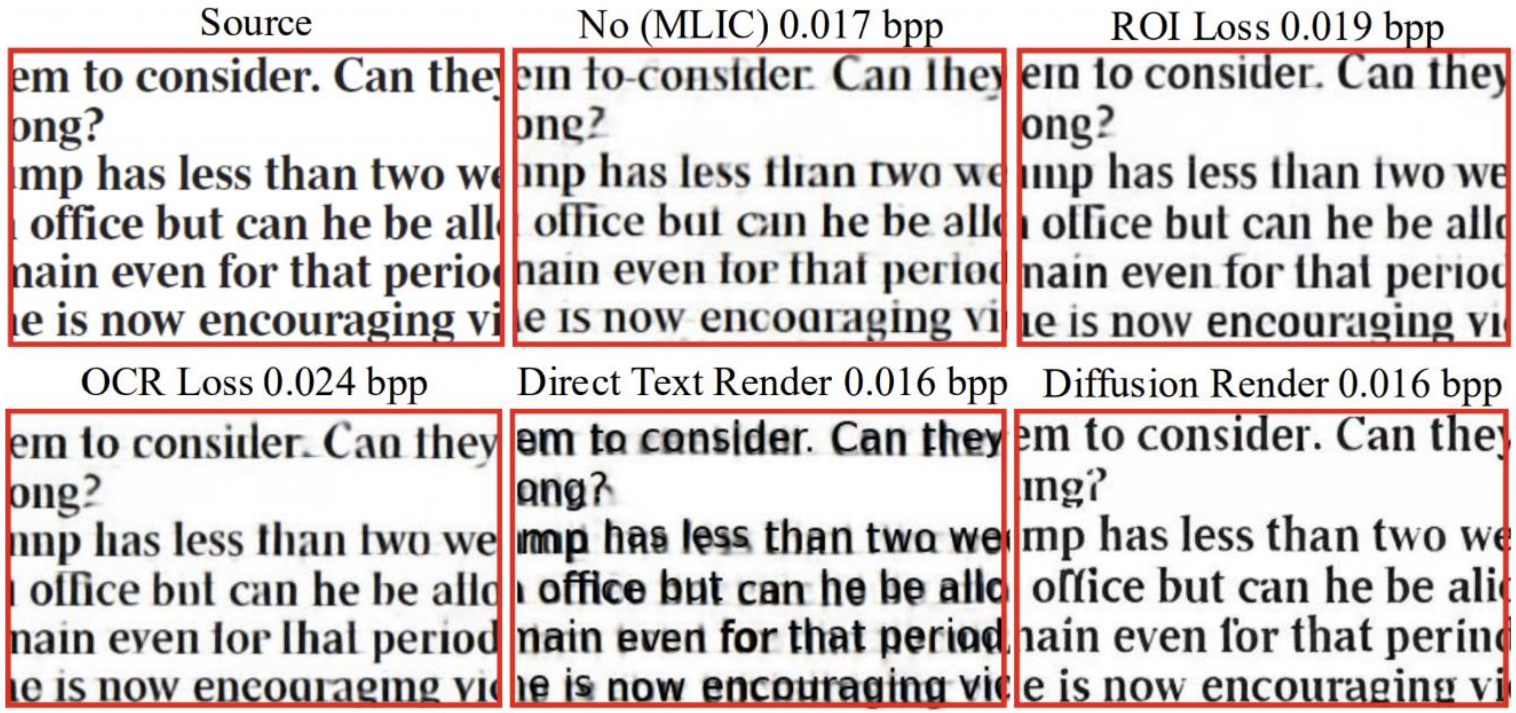
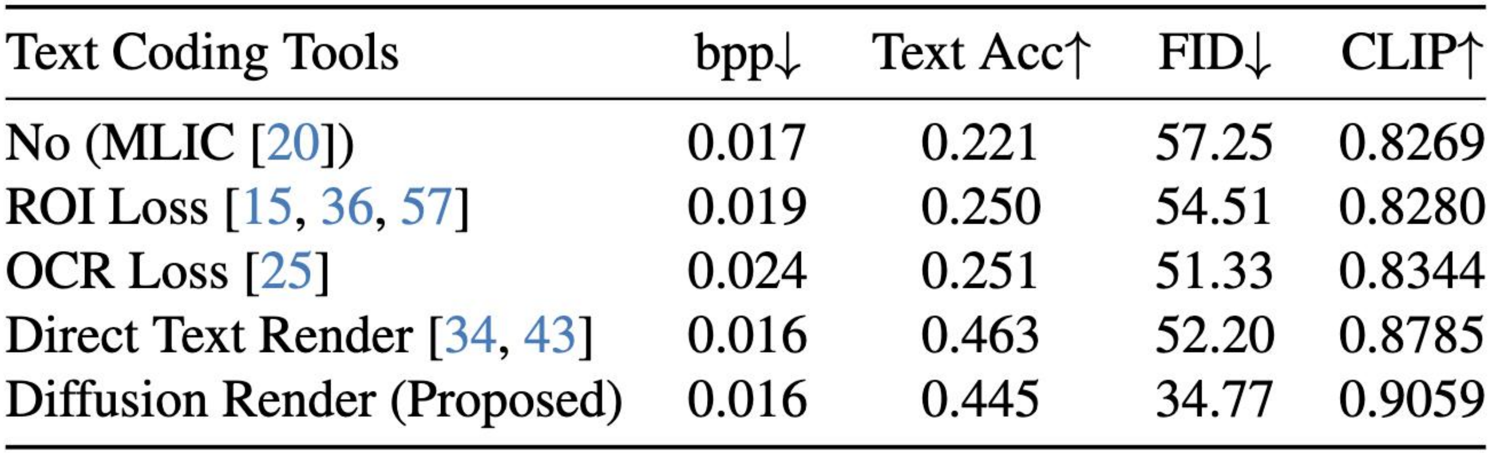


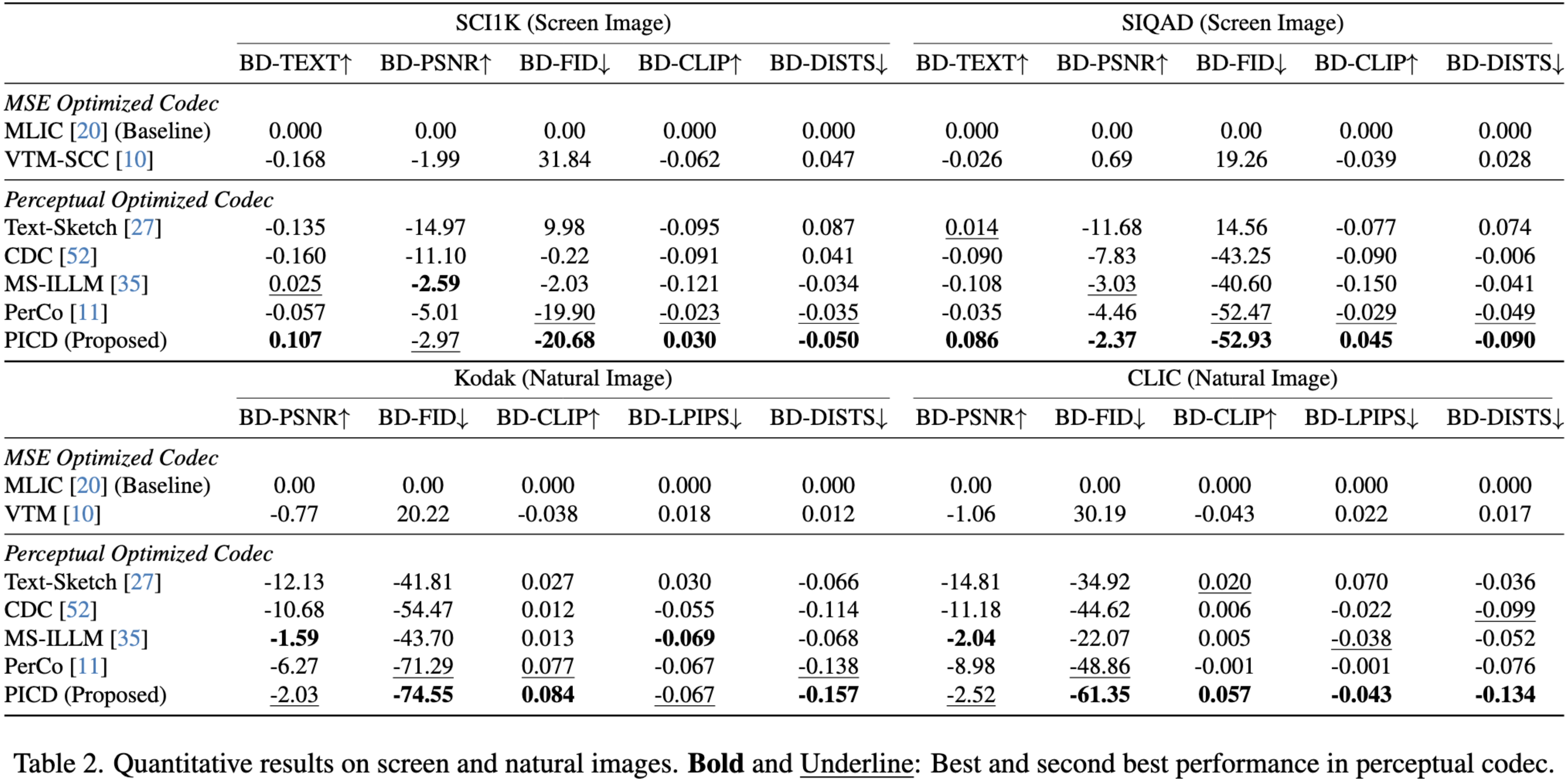
Figure 9. Visual results of different text coding tools.



Text Coding Tools	bpp↓	Text Acc↑	FID↓	CLIP↑
No (MLIC [20])	0.017	0.221	57.25	0.8269
ROI Loss [15, 36, 57]	0.019	0.250	54.51	0.8280
OCR Loss [25]	0.024	0.251	51.33	0.8344
Direct Text Render [34, 43]	0.016	0.463	52.20	0.8785
Diffusion Render (Proposed)	0.016	0.445	34.77	0.9059

Table 3. Ablation studies on the means of preserving text content.

- Comparison to other codec:
  - Ours is the only able to achieve high text accuracy and visual quality at the same time



	SCI1K (Screen Image)					SIQAD (Screen Image)				
	BD-TEXT↑	BD-PSNR↑	BD-FID↓	BD-CLIP↑	BD-DISTS↓	BD-TEXT↑	BD-PSNR↑	BD-FID↓	BD-CLIP↑	BD-DISTS↓
MSE Optimized Codec										
MLIC [20] (Baseline)	0.000	0.00	0.00	0.000	0.000	0.000	0.00	0.00	0.000	0.000
VTM-SCC [10]	-0.168	-1.99	31.84	-0.062	0.047	-0.026	0.69	19.26	-0.039	0.028
Perceptual Optimized Codec										
Text-Sketch [27]	-0.135	-14.97	9.98	-0.095	0.087	0.014	-11.68	14.56	-0.077	0.074
CDC [52]	-0.160	-11.10	-0.22	-0.091	0.041	-0.090	-7.83	-43.25	-0.090	-0.006
MS-ILLM [35]	0.025	**-2.59**	-2.03	-0.121	-0.034	-0.108	-3.03	-40.60	-0.150	-0.041
PerCo [11]	-0.057	-5.01	-19.90	-0.023	-0.035	-0.035	-4.46	-52.47	-0.029	-0.049
PICD (Proposed)	**0.107**	-2.97	**-20.68**	**0.030**	**-0.050**	**0.086**	**-2.37**	**-52.93**	**0.045**	**-0.090**

	Kodak (Natural Image)					CLIC (Natural Image)				
	BD-PSNR↑	BD-FID↓	BD-CLIP↑	BD-LPIPS↓	BD-DISTS↓	BD-PSNR↑	BD-FID↓	BD-CLIP↑	BD-LPIPS↓	BD-DISTS↓
MSE Optimized Codec										
MLIC [20] (Baseline)	0.00	0.00	0.000	0.000	0.000	0.00	0.00	0.000	0.000	0.000
VTM [10]	-0.77	20.22	-0.038	0.018	0.012	-1.06	30.19	-0.043	0.022	0.017
Perceptual Optimized Codec										
Text-Sketch [27]	-12.13	-41.81	0.027	0.030	-0.066	-14.81	-34.92	0.020	0.070	-0.036
CDC [52]	-10.68	-54.47	0.012	-0.055	-0.114	-11.18	-44.62	0.006	-0.022	-0.099
MS-ILLM [35]	**-1.59**	-43.70	0.013	**-0.069**	-0.068	**-2.04**	-22.07	0.005	-0.038	-0.052
PerCo [11]	-6.27	-71.29	0.077	-0.067	-0.138	-8.98	-48.86	-0.001	-0.001	-0.076
PICD (Proposed)	-2.03	**-74.55**	**0.084**	-0.067	**-0.157**	-2.52	**-61.35**	**0.057**	**-0.043**	**-0.134**

Table 2. Quantitative results on screen and natural images. **Bold** and Underline: Best and second best performance in perceptual codec.

Resources

- Contact: x.tongda@nyu.edu, or wechat: 18510201763
- Reference:
  - Optimal machine code: [preprint 23 Conditional Perceptual Quality Preserving Image Compression]
  - Perceptual codec: [ICML 19 The Rate-Distortion-Perception Tradeoff]
  - Diffusion control: [CVPR 23 Adding Conditional Control to Text-to-Image Diffusion Models] [IJCV 24 Exploiting Diffusion Prior for Real-World Image Super-Resolution]
  - Glyph rendering: [GlyphControl: Glyph Conditional Control for Visual Text Generation]
  - Diffusion guidance: [ICLR 23 Diffusion Posterior Sampling for General Noisy Inverse Problems] [ICLR 24 Idempotence and Perceptual Image Compression]
  - MSE optimized codec: [MM 23 MLIC: Multi-Reference Entropy Model for Learned Image Compression]