Get What You Want, Not What You Don't: Image Content Suppression for Text-to-Image Diffusion Models

Senmao Li[1] Joost van de Weijer[2] Taihang Hu[1] Fahad Shahbaz Khan[3,4] Qibin Hou[1]
Yaxing Wang[1*] Jian Yang[1]

[1]VCIP, CS, Nankai University, [2]Universitat Auto`noma de Barcelona,
[3]Mohamed bin Zayed University of AI, [4]Linkoping University

Paper ID 291
Code: https://github.com/sen-mao/SuppressEOT

Problem

- Existing text-to-image models can encounter challenges in effectively suppressing the generation of the ***negative target*** (e.g., “glasses” in “A man without glasses”)

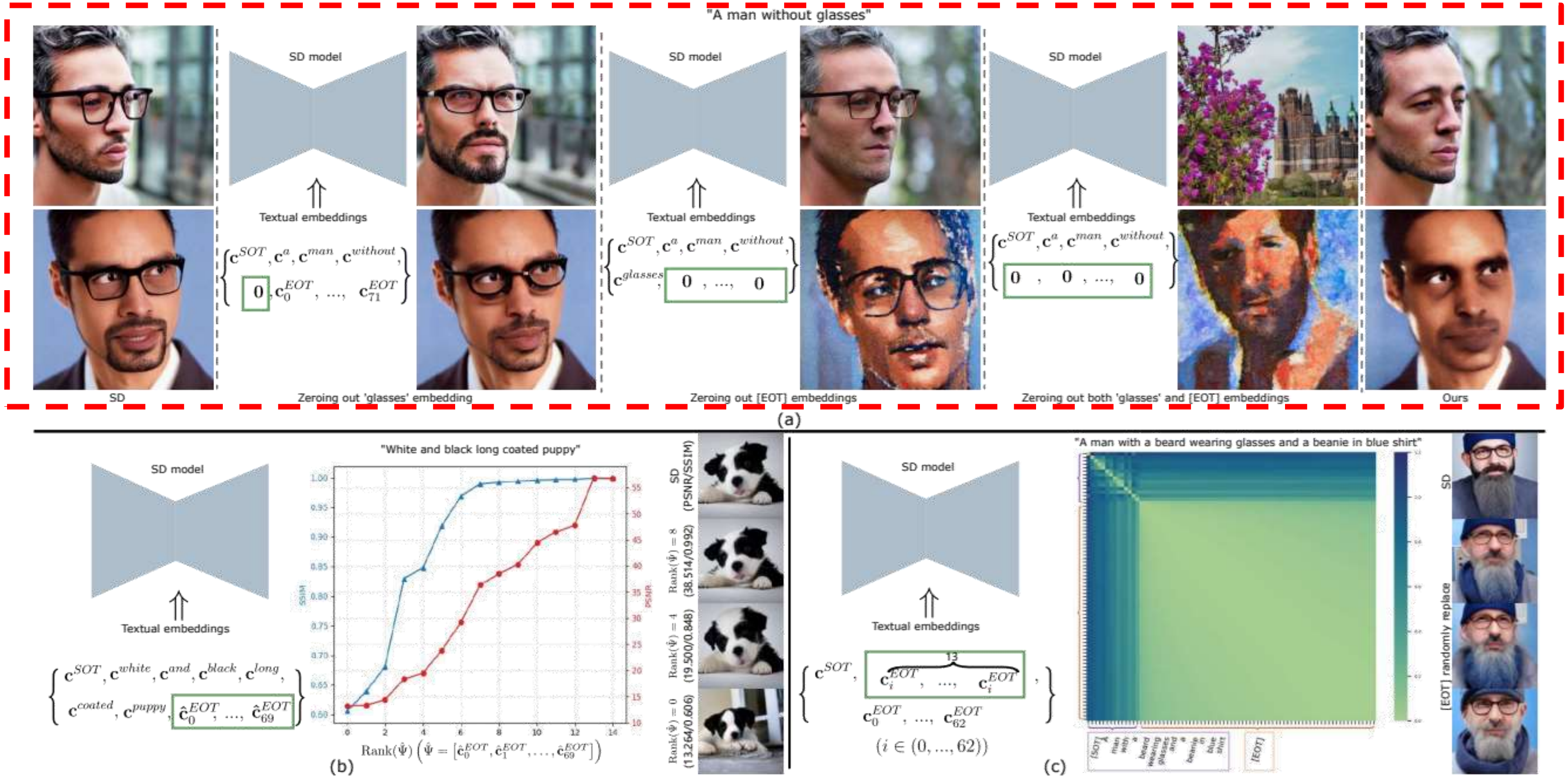
Failure cases of Stable Diffusion (SD) and DeepFloyd-IF

Additional failure cases

We aim to manipulate the **text embeddings** and remove unwanted content. We find that **[EOT] embeddings** also contain content information

Analysis

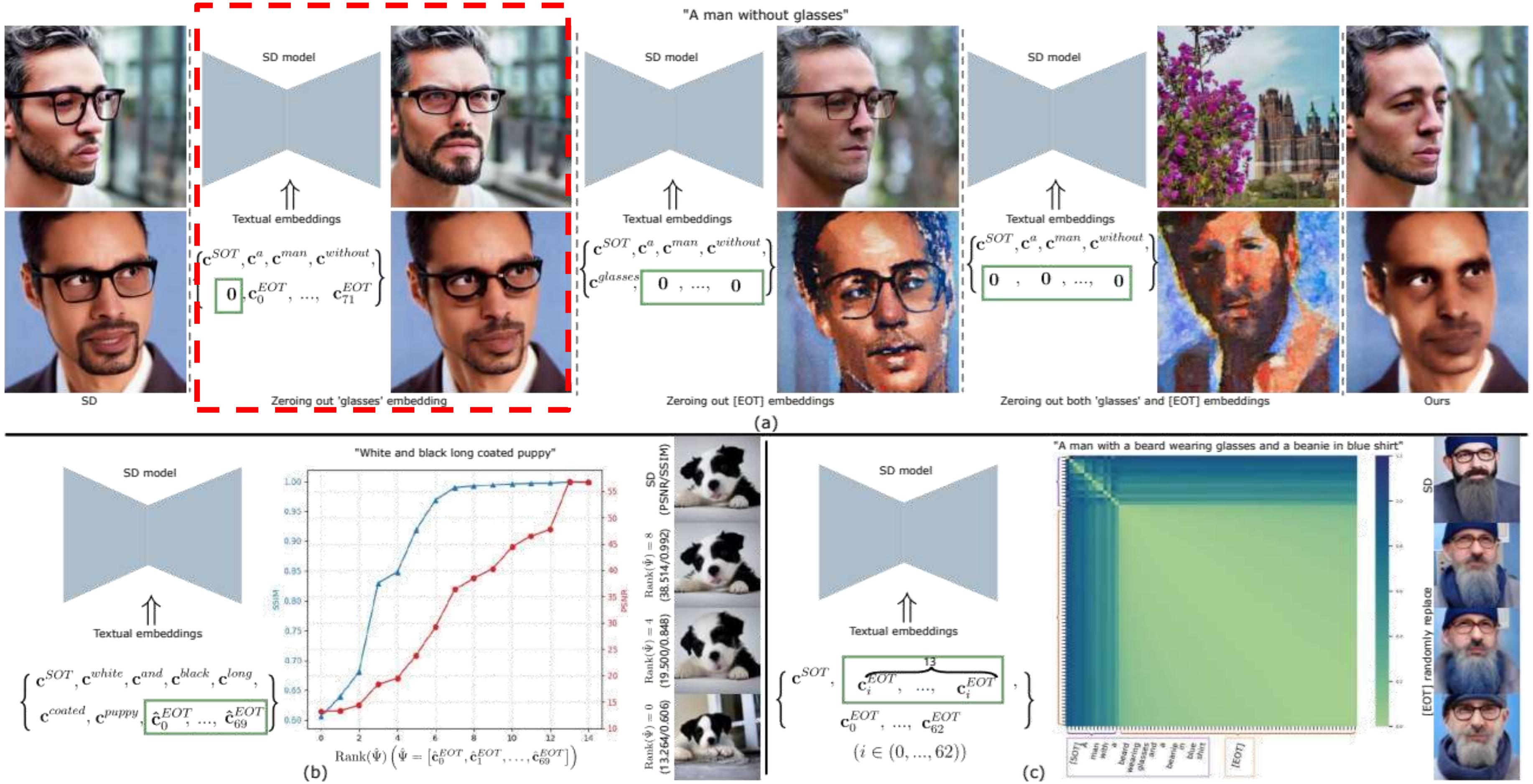
- (a) The [EOT] embeddings contain **significant information** about the input prompt
- (b) The [EOT] embeddings have **the low-rank property**, and contain **redundant semantic information**
- (c) The various [EOT] embeddings are **highly correlated**



(a) Zeroing out “glasses” and [EOT] embeddings. (b) Performing low-rank technique. (c) Distance matrix between all text embeddings.

Analysis

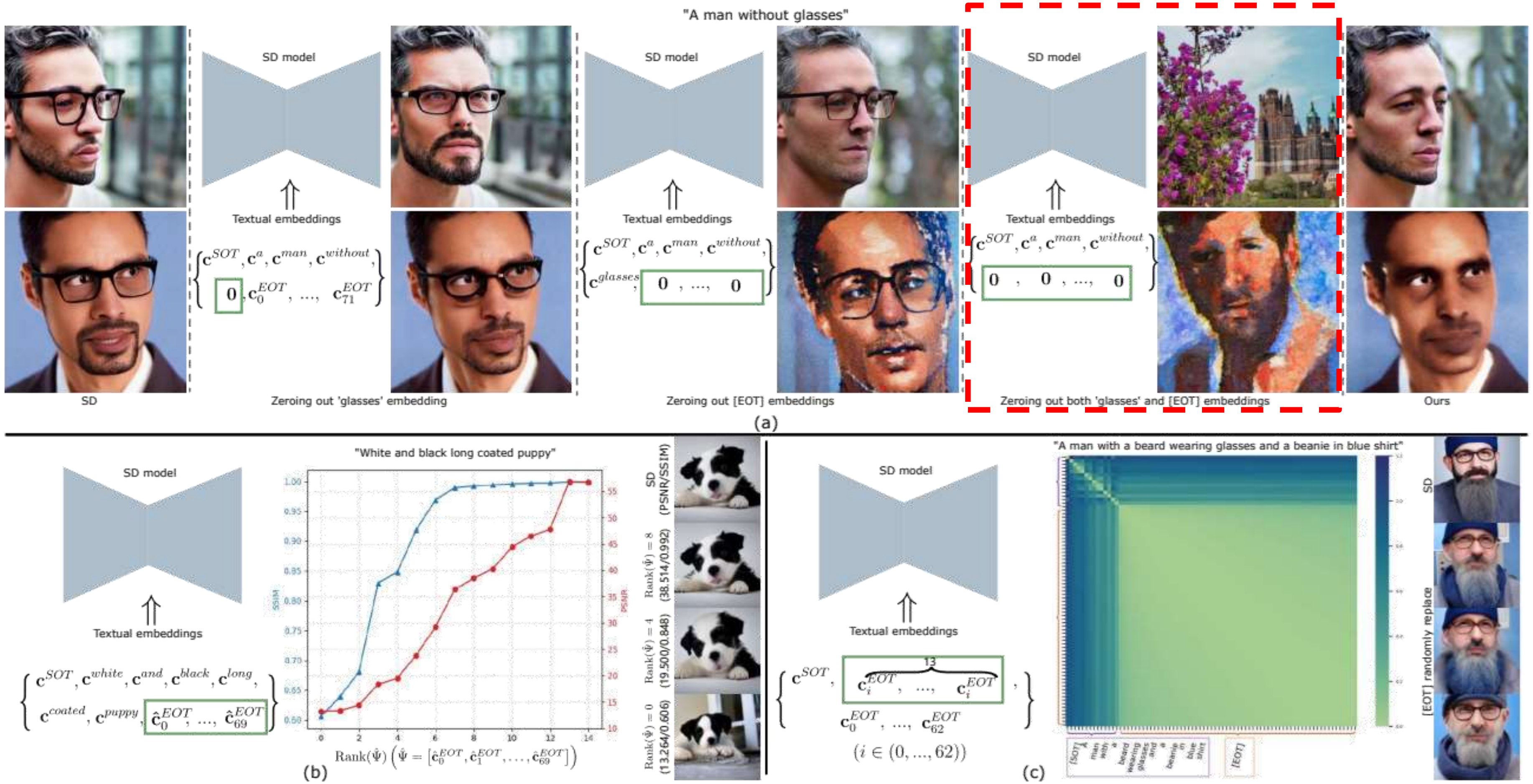
- (a) The [EOT] embeddings contain **significant information** about the input prompt
- (b) The [EOT] embeddings have **the low-rank property**, and contain **redundant semantic information**
- (c) The various [EOT] embeddings are **highly correlated**



(a) Zeroing out “glasses” and [EOT] embeddings. (b) Performing low-rank technique. (c) Distance matrix between all text embeddings.

Analysis

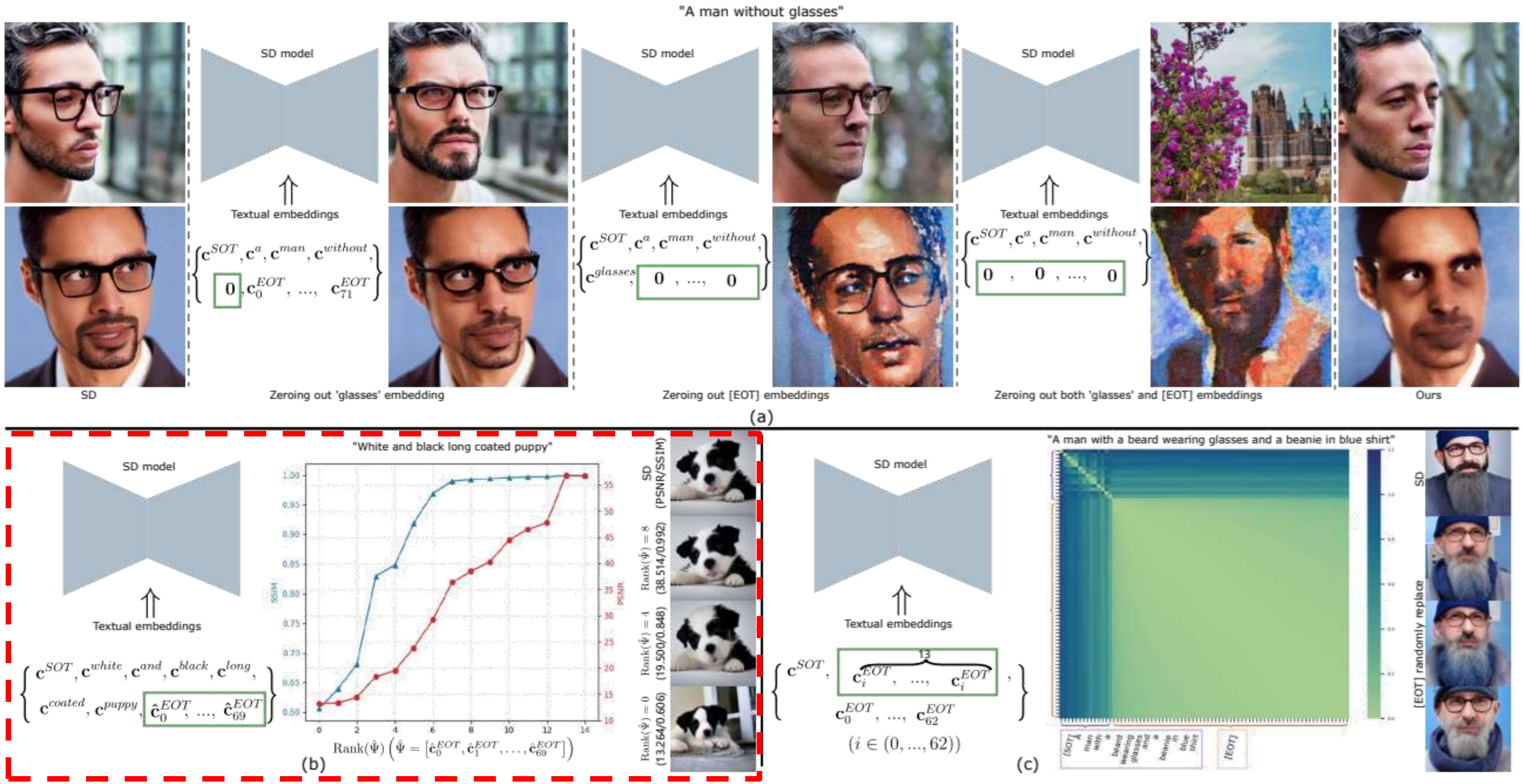
- (a) The [EOT] embeddings contain **significant information** about the input prompt
- (b) The [EOT] embeddings have **the low-rank property**, and contain **redundant semantic information**
- (c) The various [EOT] embeddings are **highly correlated**



(a) Zeroing out “glasses” and [EOT] embeddings. (b) Performing low-rank technique. (c) Distance matrix between all text embeddings.

Analysis

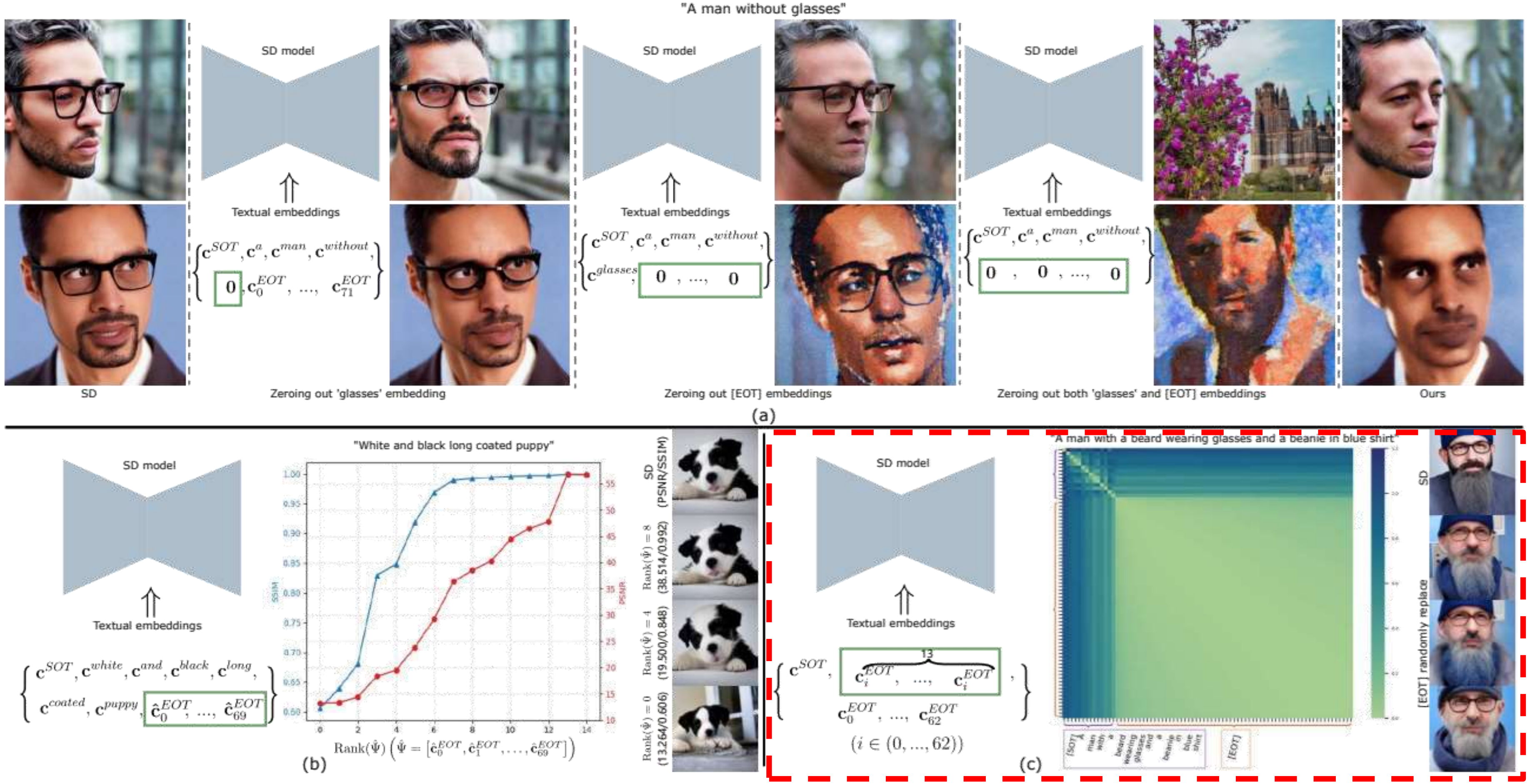
- (a) The [EOT] embeddings contain **significant information** about the input prompt
- (b) The [EOT] embeddings have **the low-rank property**, and contain **redundant semantic information**
- (c) The various [EOT] embeddings are **highly correlated**



(a) Zeroing out "glasses" and [EOT] embeddings. (b) Performing low-rank technique. (c) Distance matrix between all text embeddings.

Analysis

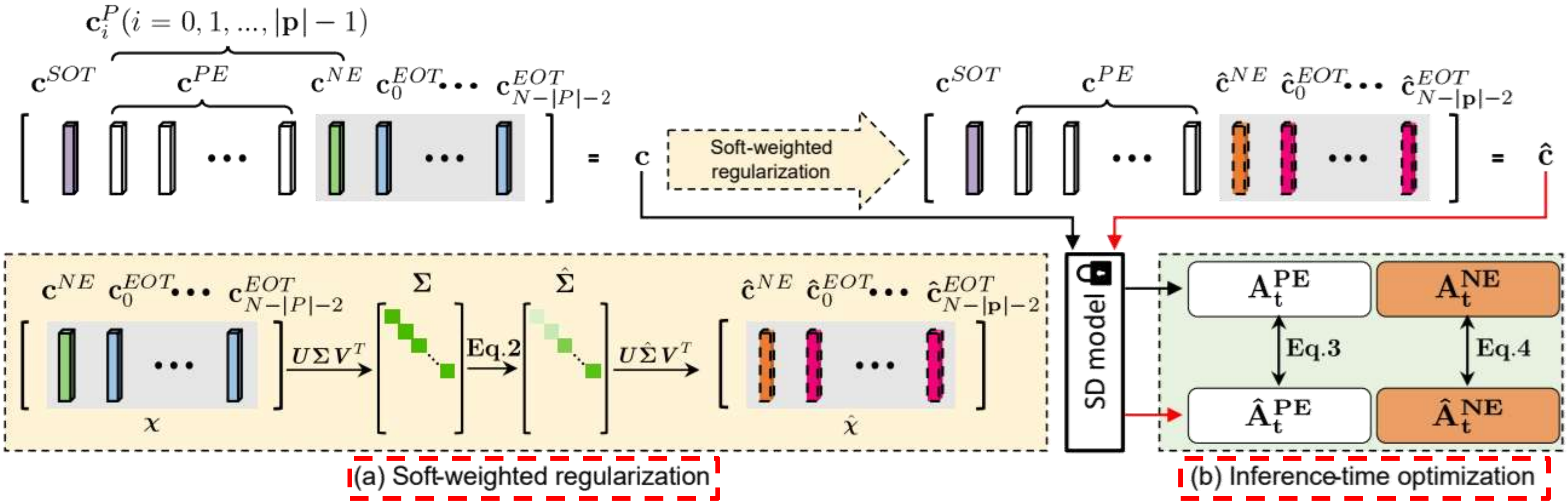
- (a) The [EOT] embeddings contain **significant information** about the input prompt
- (b) The [EOT] embeddings have **the low-rank property**, and contain **redundant semantic information**
- (c) The various [EOT] embeddings are **highly correlated**



(a) Zeroing out "glasses" and [EOT] embeddings. (b) Performing low-rank technique. (c) Distance matrix between all text embeddings.

Method (SWR)

We introduce ***soft-weighted regularization (SWR)*** and ***inference-time text embedding optimization (ITO)***



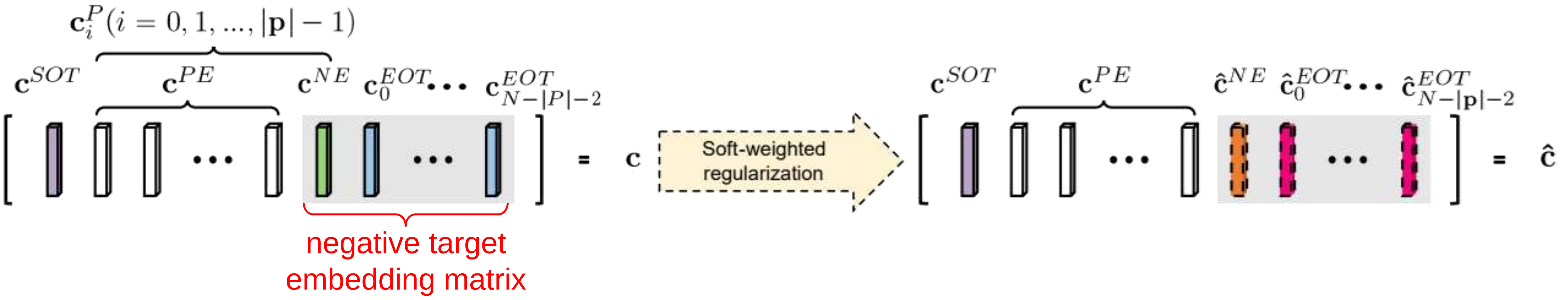
WNNM: $\sigma - \frac{\lambda}{(\sigma+\epsilon)}$ ⇨ SWR: $\hat{\sigma} = e^{-\sigma} * \sigma.$

WNNM：Weighted nuclear norm minimization with application to image denoising

Method (SWR)

We introduce ***soft-weighted regularization (SWR)*** and ***inference-time text embedding optimization (ITO)***

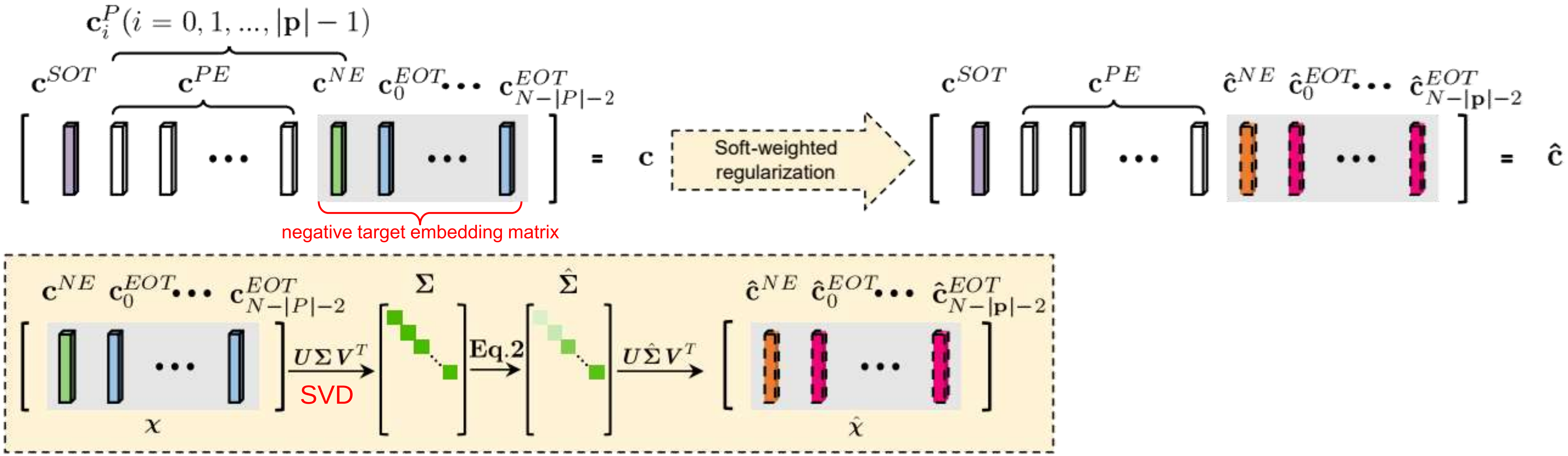
- **SWR** regularizes the text embedding matrix and effectively suppresses the undesired content



Method (SWR)

We introduce ***soft-weighted regularization (SWR)*** and ***inference-time text embedding optimization (ITO)***

- **SWR** regularizes the text embedding matrix and effectively suppresses the undesired content



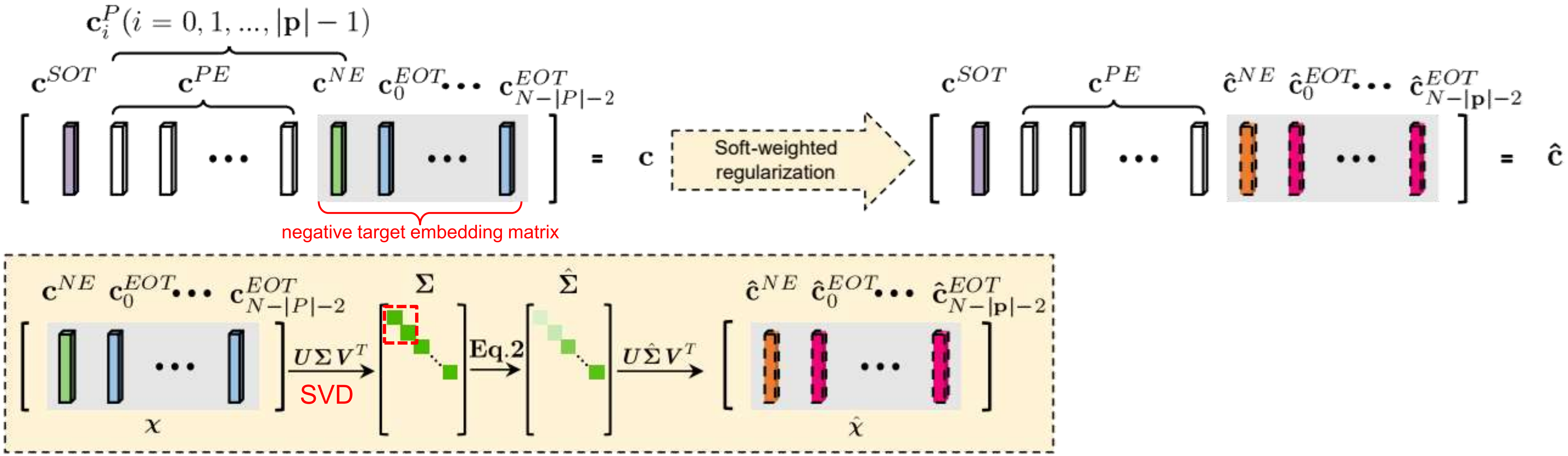
(a) Soft-weighted regularization

SWR: $$\hat{\sigma} = e^{-\sigma} * \sigma. \qquad (2)$$

Method (SWR)

We introduce ***soft-weighted regularization (SWR)*** and ***inference-time text embedding optimization (ITO)***

- **SWR** regularizes the text embedding matrix and effectively suppresses the undesired content



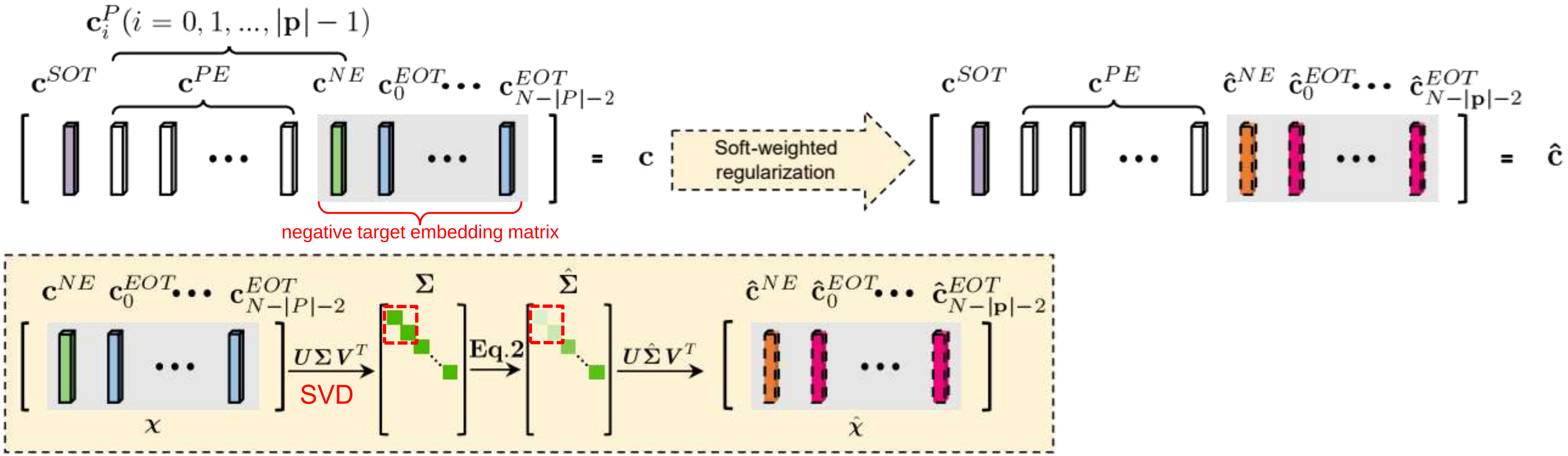
(a) Soft-weighted regularization

SWR: $\hat{\sigma} = e^{-\sigma} * \sigma.$ (2)

Method (SWR)

We introduce ***soft-weighted regularization (SWR)*** and ***inference-time text embedding optimization (ITO)***

- **SWR** regularizes the text embedding matrix and effectively suppresses the undesired content



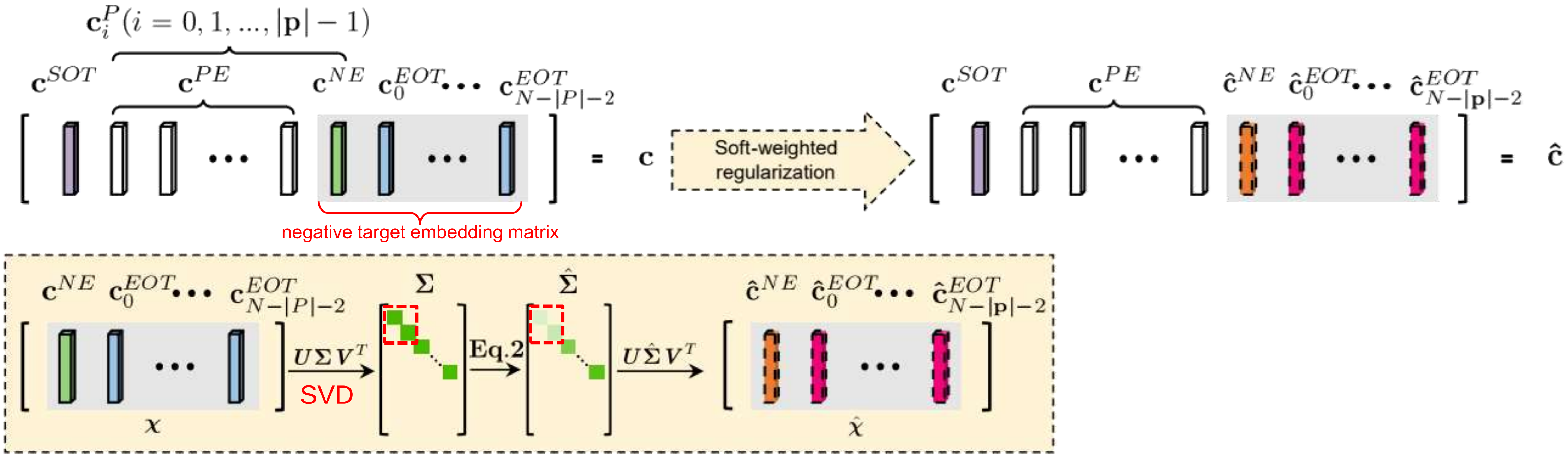
(a) Soft-weighted regularization

SWR: $$\hat{\sigma} = e^{-\sigma} * \sigma. \qquad (2)$$

Method (SWR)

We introduce ***soft-weighted regularization (SWR)*** and ***inference-time text embedding optimization (ITO)***

- **SWR** regularizes the text embedding matrix and effectively suppresses the undesired content



(a) Soft-weighted regularization

WNNM: $\sigma - \frac{\lambda}{(\sigma+\epsilon)}$ ⇒ SWR: $\hat{\sigma} = e^{-\sigma} * \sigma.$ (2)

WNNM：Weighted nuclear norm minimization with application to image denoising

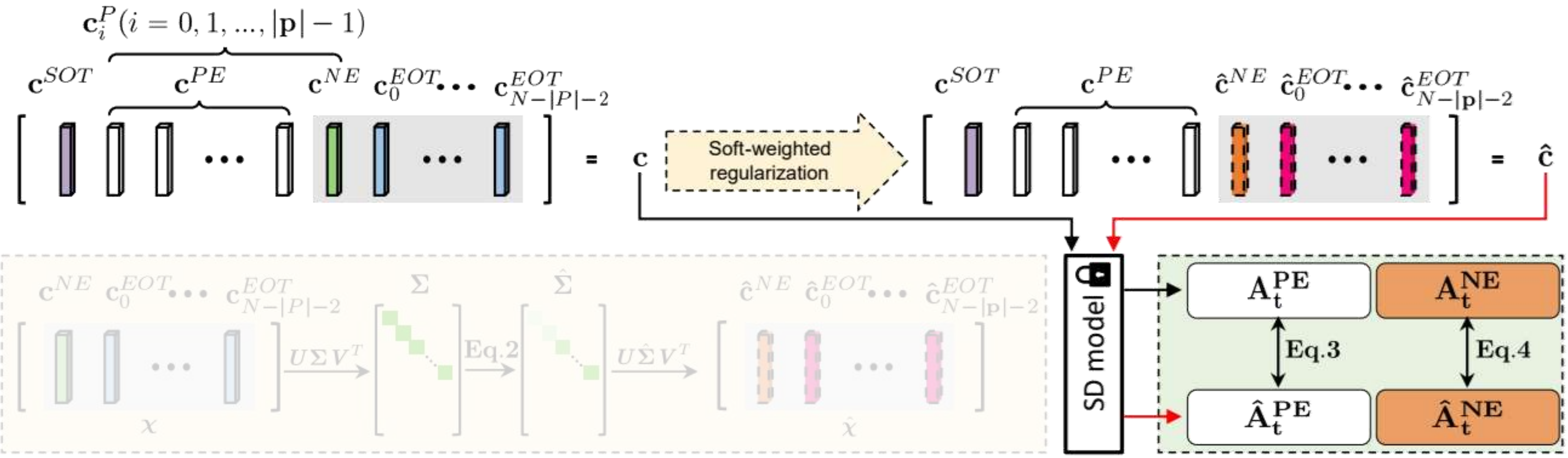
Method (ITO)

We introduce ***soft-weighted regularization (SWR)*** and ***inference-time text embedding optimization (ITO)***

- ITO aims to further suppress the unwanted content generation of the prompt, and encourages the generation of desired content

$$\mathcal{L}_{nl} = -\left\| \hat{A}_t^{NE} - A_t^{NE} \right\|^2, \quad (4)$$

$$\mathcal{L}_{pl} = \left\| \hat{A}_t^{PE} - A_t^{PE} \right\|^2. \quad (3)$$



(a) Soft-weighted regularization (b) Inference-time optimization

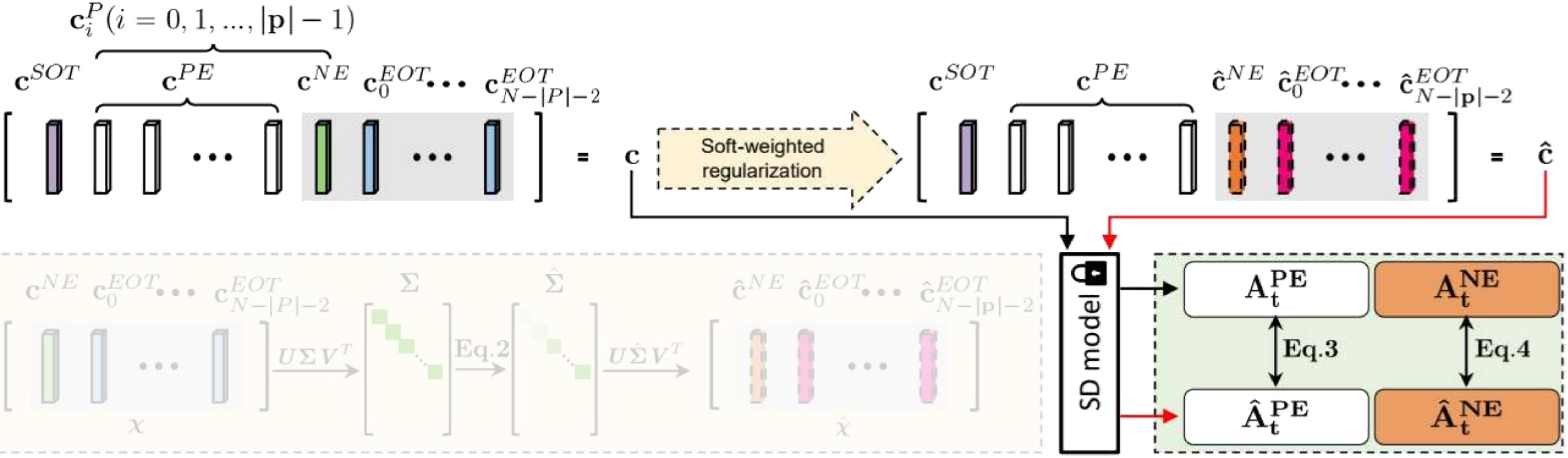
Method (ITO)

We introduce ***soft-weighted regularization (SWR)*** and ***inference-time text embedding optimization (ITO)***

- ITO aims to further suppress the unwanted content generation of the prompt, and encourages the generation of desired content

$$\mathcal{L}_{nl} = -\left\|\hat{A}_t^{NE} - A_t^{NE}\right\|^2, \quad (4)$$

$$\mathcal{L}_{pl} = \left\|\hat{A}_t^{PE} - A_t^{PE}\right\|^2. \quad (3)$$

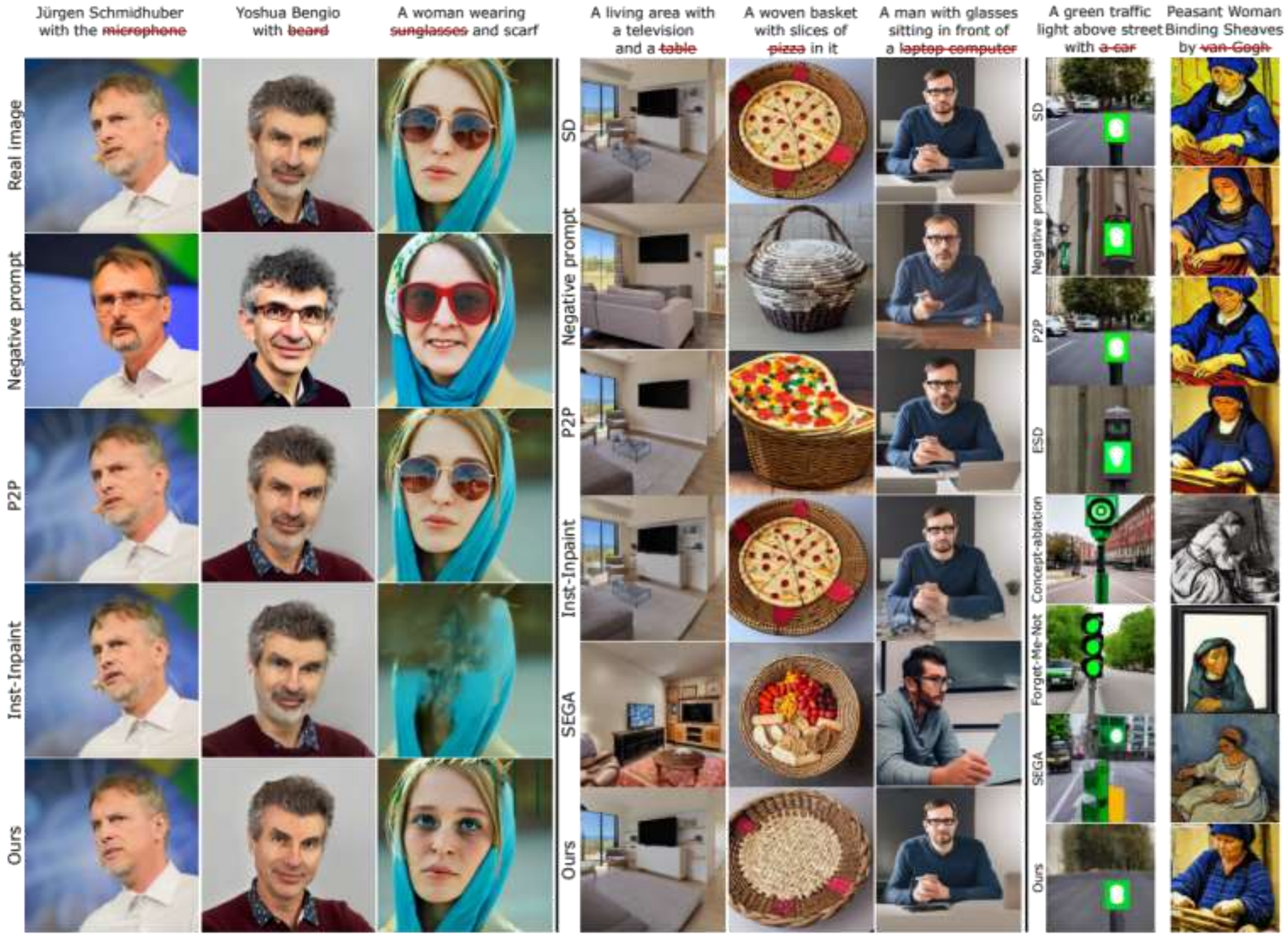


(a) Soft-weighted regularization

(b) Inference-time optimization

Results

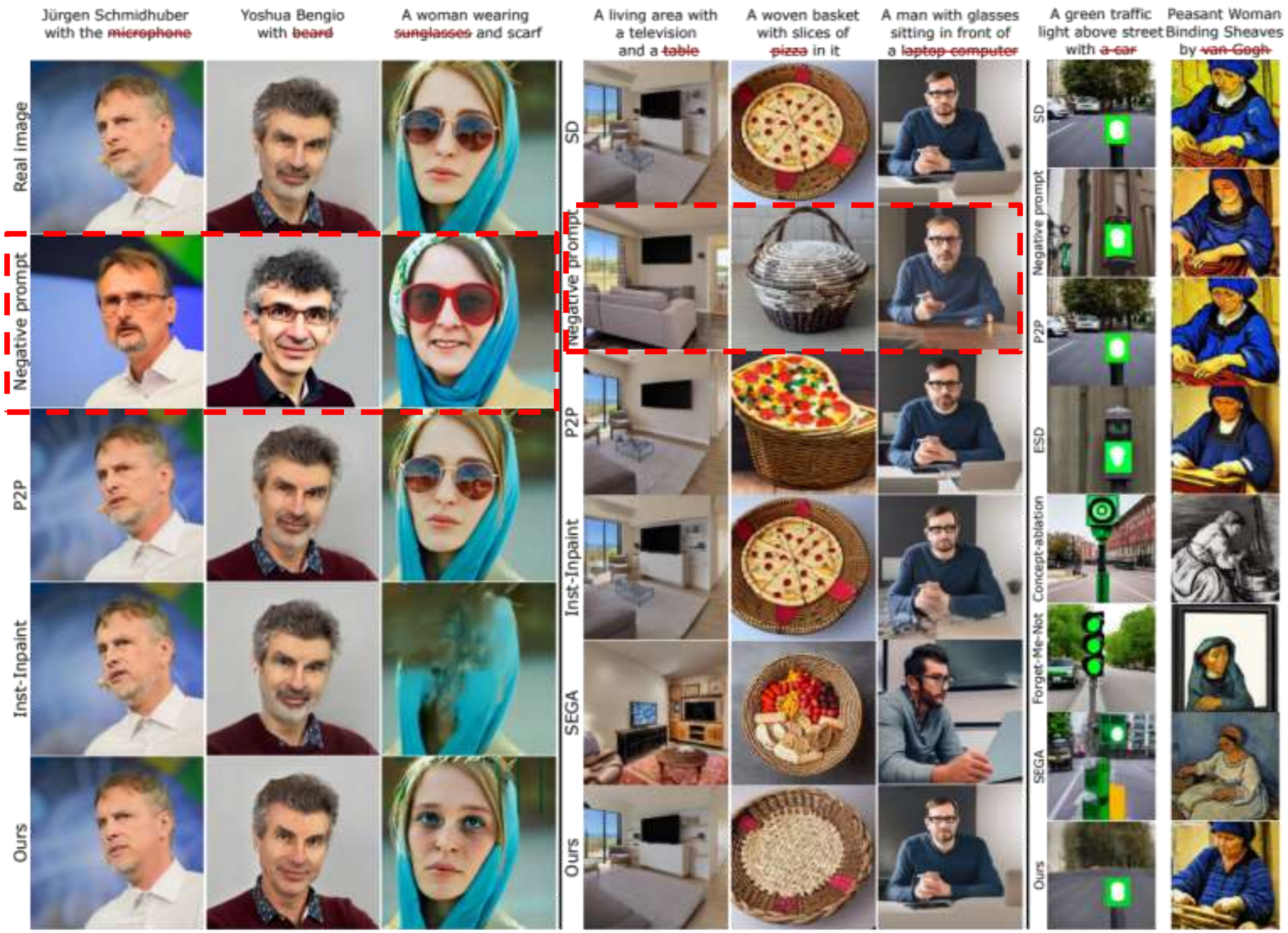
Real image (Left) and generated image (Middle and Right) suppression results



We have the best performance (the last row), without further finetuning the model

Results

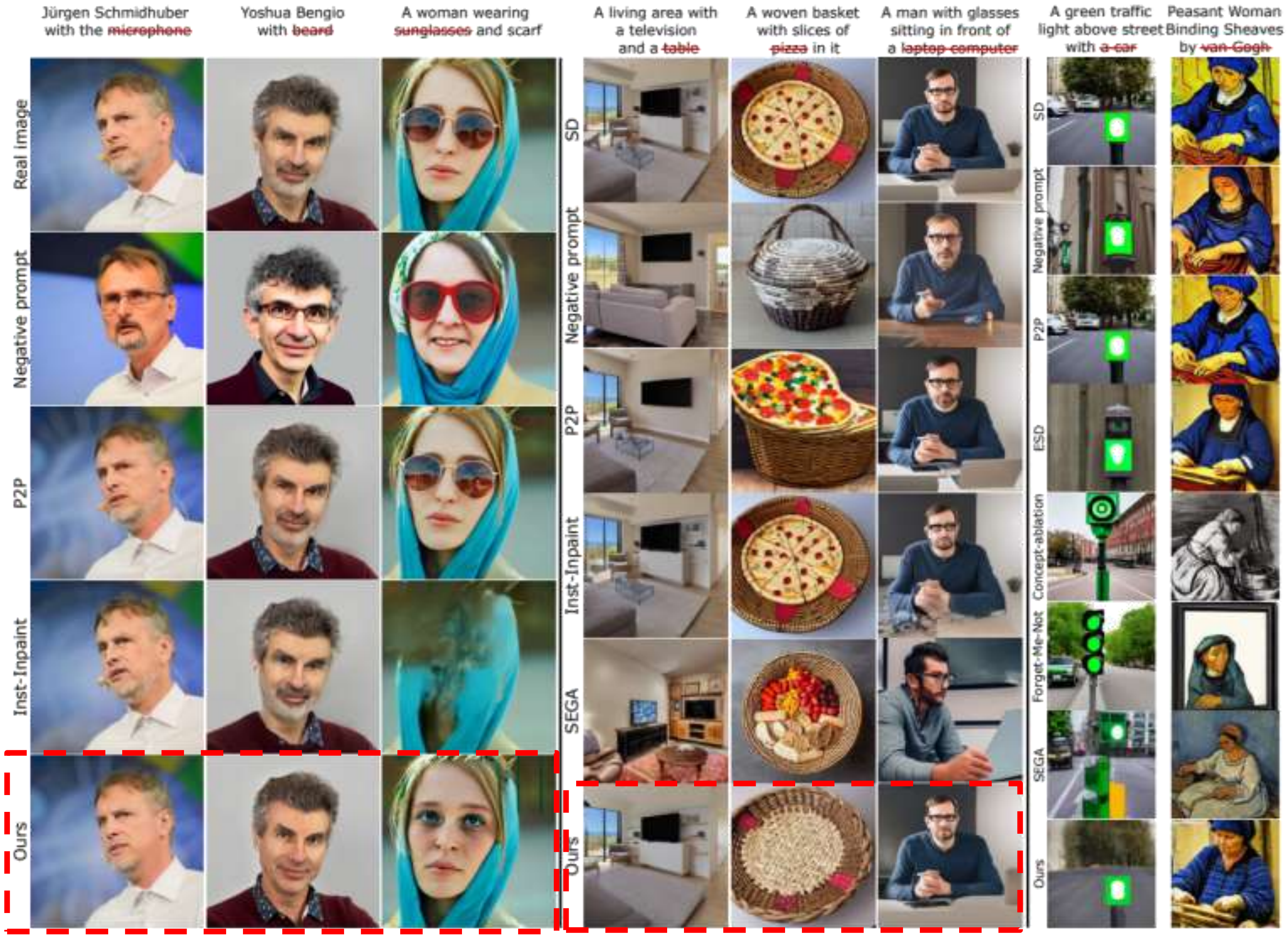
Real image (Left) and generated image (Middle and Right) suppression results



We have the best performance (the last row), without further finetuning the model

Results

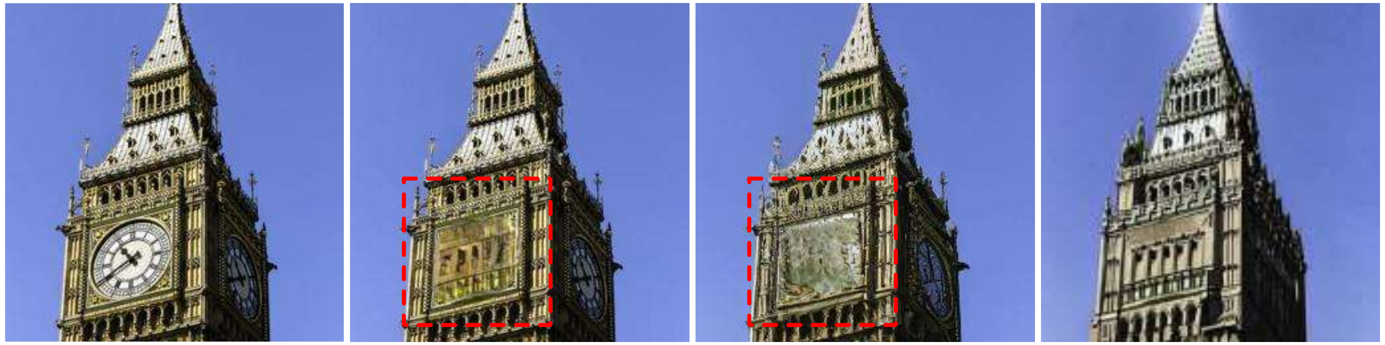
Real image (Left) and generated image (Middle and Right) suppression results



We have the best performance (the last row), without further finetuning the model

Results

The ~~clock~~ on the building



Real image | GT | Inst-Inpaint | Ours

Table 2: Quantitative comparison on the GQA-Inpaint dataset for real image negative target suppression task.

Methods	Paired data	FID ↓	CLIP Acc ↑	CLIP Acc (top5) ↑
X-Decoder	✓	6.86	69.9	46.5
Inst-Inpaint	✓	**5.50**	80.5	60.4
Ours	✗	13.87	**92.8**	**83.3**

We achieve superior suppression results and higher CLIP Accuracy scores on the GQA-Inpaint dataset

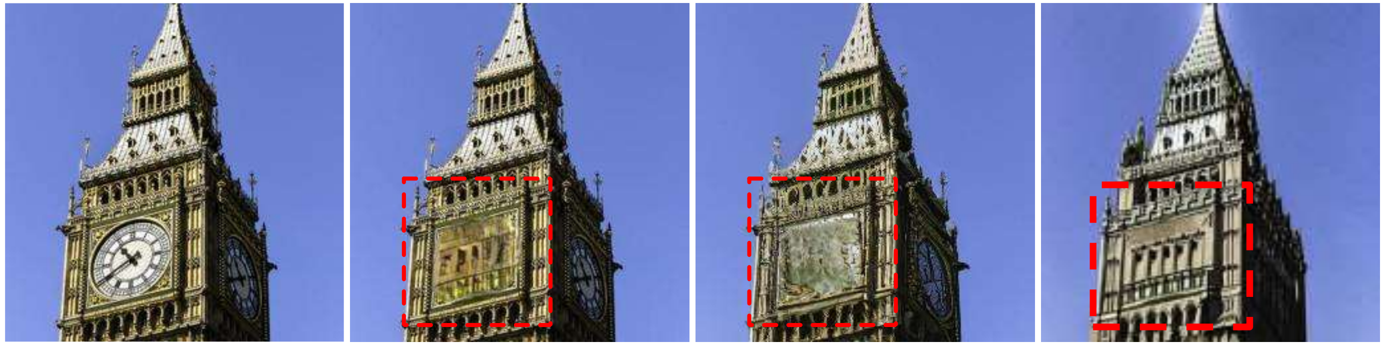
Table 1: Comparison with baselines. The best results are in bold, and the second best results are <u>underlined</u>.

Method	*Real-image* editing			*Generated-image* editing									
	Random negative target			Random negative target			Negative target: Car			Negative target: Tyler Edlin		Negative target: Van Gogh	
	Clipscore↓	IFID↑	DetScore↓	Clipscore↓	IFID↑	DetScore↓	Clipscore↓	IFID↑	DetScore↓	Clipscore↓	IFID↑	Clipscore↓	IFID↑
Real image or SD (Generated image)	0.7986	0	0.3381	0.8225	0	0.4509	0.8654	0	0.6643	0.7414	0	0.8770	0
Negative prompt	0.7983	**175.8**	0.2402	<u>0.7619</u>	169.0	<u>0.1408</u>	0.8458	151.7	0.5130	0.7437	<u>233.9</u>	0.8039	242.1
P2P (Hertz et al., 2022)	0.7666	92.53	0.1758	0.8118	103.3	0.3391	0.8638	21.7	0.6343	0.7470	86.3	0.8849	139.7
ESD (Gandikota et al., 2023)	-	-	-	-	-	-	0.7986	165.7	0.2223	**0.6954**	**256.5**	<u>0.7292</u>	<u>267.5</u>
Concept-ablation (Kumari et al., 2023)	-	-	-	-	-	-	<u>0.7642</u>	<u>179.3</u>	<u>0.0935</u>	0.7411	211.4	0.8290	219.9
Forget-Me-Not (Zhang et al., 2023)	-	-	-	-	-	-	0.8701	158.7	0.5867	0.7495	227.9	0.8391	203.5
Inst-Inpaint (Yildirim et al., 2023)	<u>0.7327</u>	135.5	<u>0.1125</u>	0.7602	150.4	0.1744	0.8009	126.9	0.2361	-	-	-	-
SEGA (Brack et al., 2023)	-	-	-	0.7960	<u>172.2</u>	0.3005	0.8001	168.8	0.4767	0.7678	209.9	0.8730	175.0
Ours	**0.6857**	<u>166.3</u>	**0.0384**	**0.6647**	**176.4**	**0.1321**	**0.7426**	**206.8**	**0.0419**	<u>0.7402</u>	217.7	**0.6448**	**307.5**

For real-image suppression, we achieve the best score in both Clipscore and DetScore

Results

The ~~clock~~ on the building



Real image | GT | Inst-Inpaint | Ours

Table 2: Quantitative comparison on the GQA-Inpaint dataset for real image negative target suppression task.

Methods	Paired data	FID ↓	CLIP Acc ↑	CLIP Acc (top5) ↑
X-Decoder	✓	6.86	69.9	46.5
Inst-Inpaint	✓	**5.50**	80.5	60.4
Ours	✗	13.87	**92.8**	**83.3**

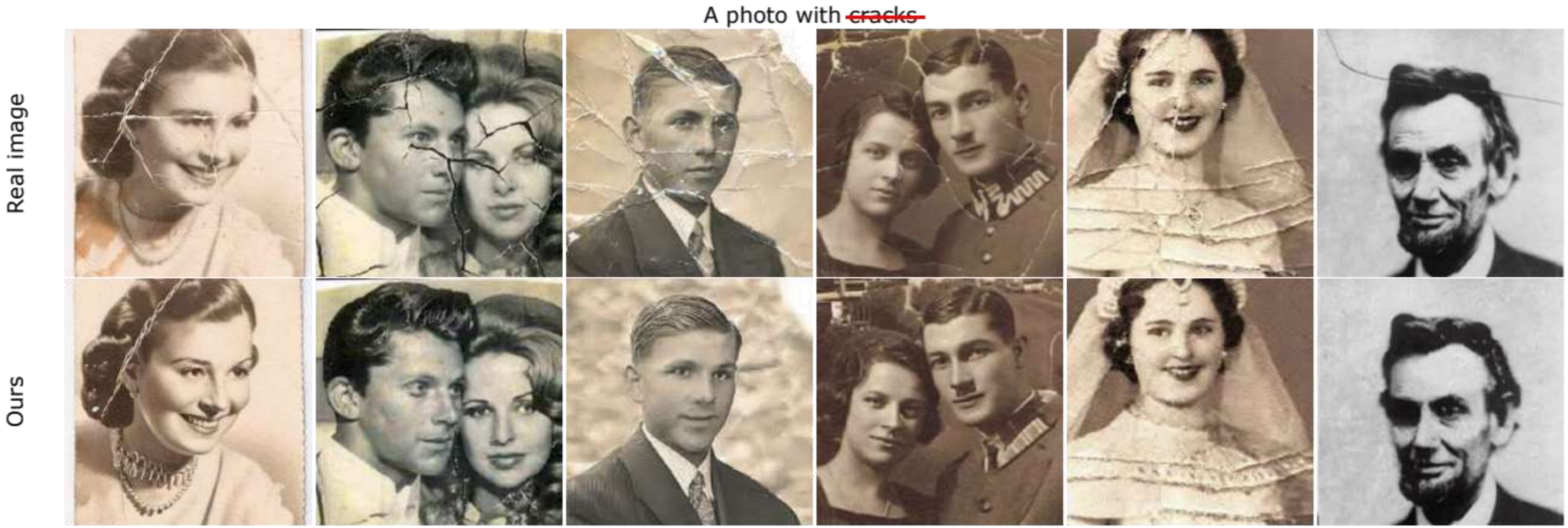
We achieve superior suppression results and higher CLIP Accuracy scores on the GQA-Inpaint dataset

Table 1: Comparison with baselines. The best results are in bold, and the second best results are underlined.

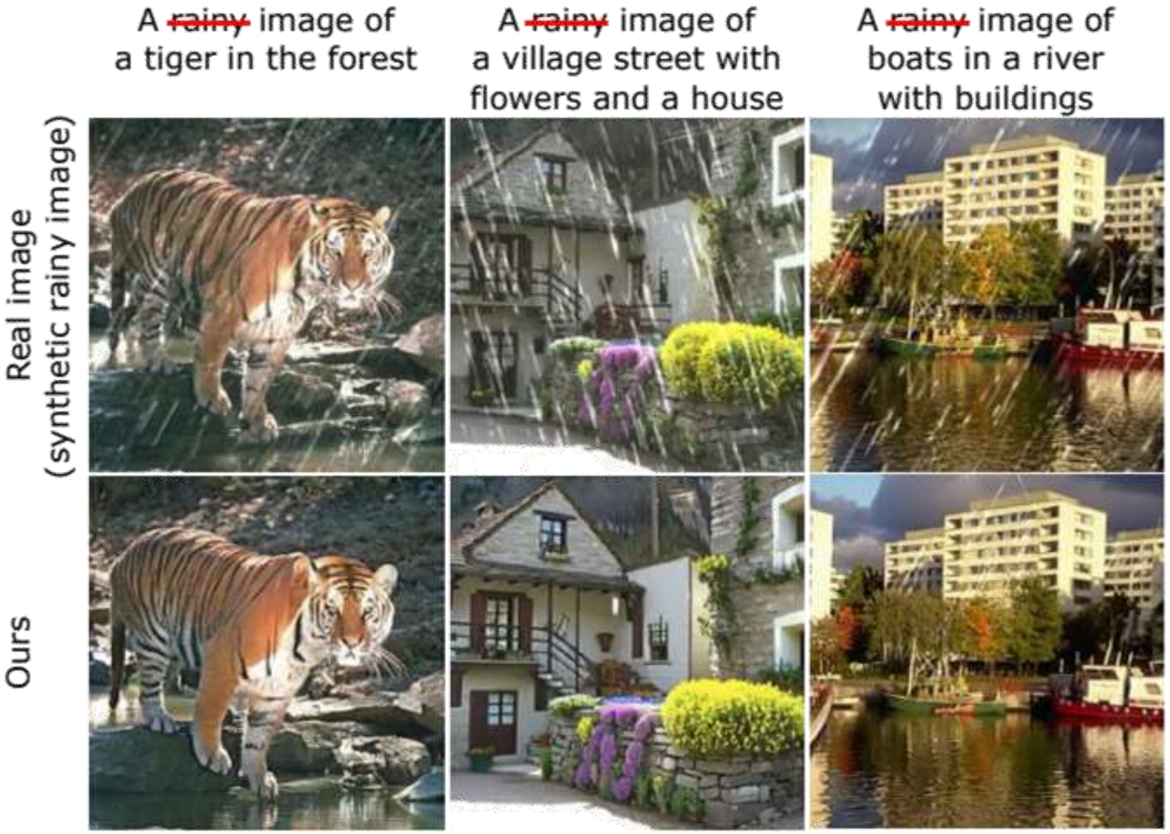
Method	*Real-image* editing			*Generated-image* editing									
	Random negative target			Random negative target			Negative target: Car			Negative target: Tyler Edlin		Negative target: Van Gogh	
	Clipscore↓	IFID↑	DetScore↓	Clipscore↓	IFID↑	DetScore↓	Clipscore↓	IFID↑	DetScore↓	Clipscore↓	IFID↑	Clipscore↓	IFID↑
Real image or SD (Generated image)	0.7986	0	0.3381	0.8225	0	0.4509	0.8654	0	0.6643	0.7414	0	0.8770	0
Negative prompt	0.7983	**175.8**	0.2402	0.7619	169.0	0.1408	0.8458	151.7	0.5130	0.7437	233.9	0.8039	242.1
P2P (Hertz et al., 2022)	0.7666	92.53	0.1758	0.8118	103.3	0.3391	0.8638	21.7	0.6343	0.7470	86.3	0.8849	139.7
ESD (Gandikota et al., 2023)	-	-	-	-	-	-	0.7986	165.7	0.2223	**0.6954**	**256.5**	0.7292	267.5
Concept-ablation (Kumari et al., 2023)	-	-	-	-	-	-	0.7642	179.3	0.0935	0.7411	211.4	0.8290	219.9
Forget-Me-Not (Zhang et al., 2023)	-	-	-	-	-	-	0.8701	158.7	0.5867	0.7495	227.9	0.8391	203.5
Inst-Inpaint (Yildirim et al., 2023)	0.7327	135.5	0.1125	0.7602	150.4	0.1744	0.8009	126.9	0.2361	-	-	-	-
SEGA (Brack et al., 2023)	-	-	-	0.7960	172.2	0.3005	0.8001	168.8	0.4767	0.7678	209.9	0.8730	175.0
Ours	**0.6857**	166.3	**0.0384**	**0.6647**	**176.4**	**0.1321**	**0.7426**	**206.8**	**0.0419**	0.7402	217.7	**0.6448**	**307.5**

For real-image suppression, we achieve the best score in both Clipscore and DetScore

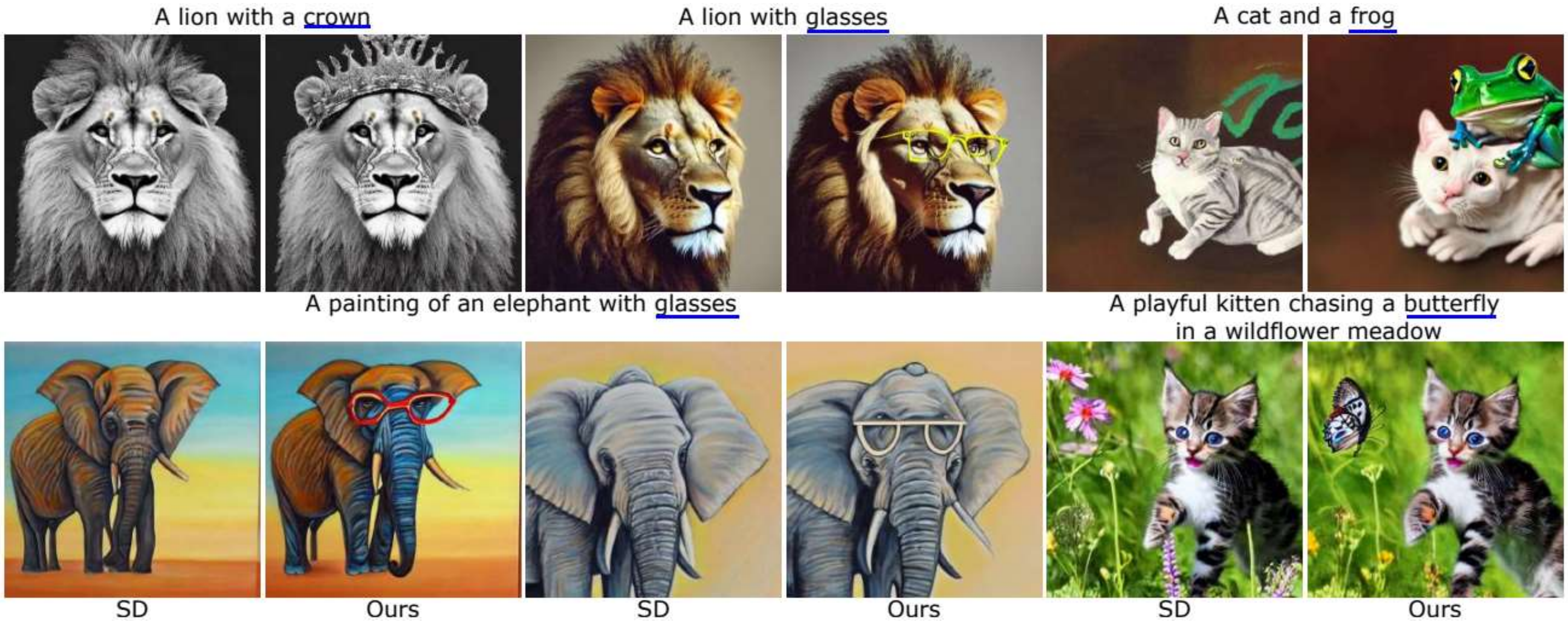
Additional results (Cracks removal results)

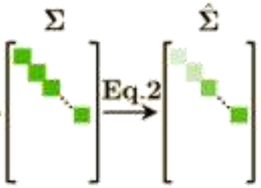


Additional results (Rain removal for synthetic & real-world rainy image)

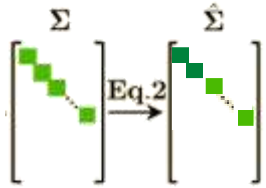


Additional results (Generating subjects for generated image)

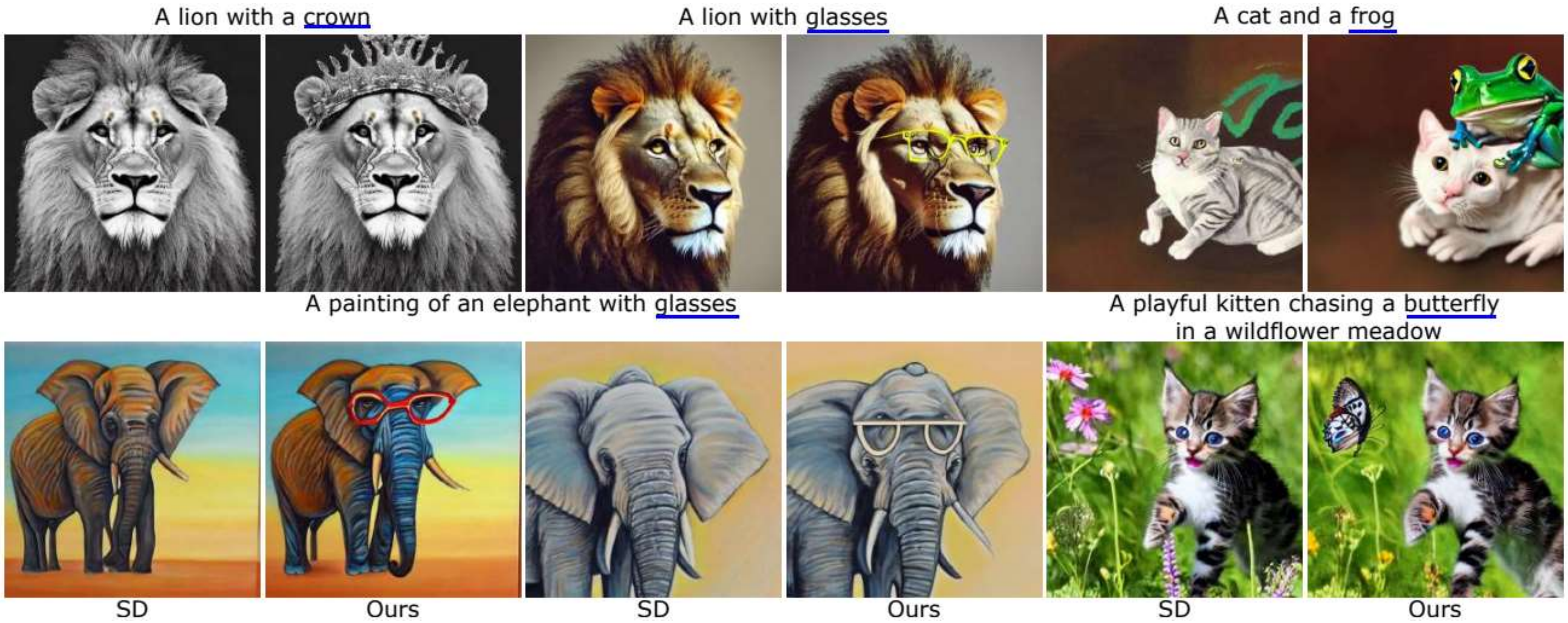


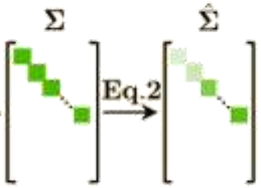


SWR: $\hat{\sigma} = e^{-\sigma} * \sigma.$ $\Rightarrow$ $\hat{\sigma} = e^{\alpha\sigma} * \sigma$

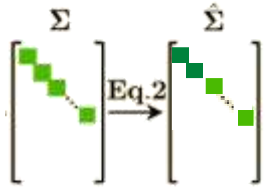


Additional results (Generating subjects for generated image)

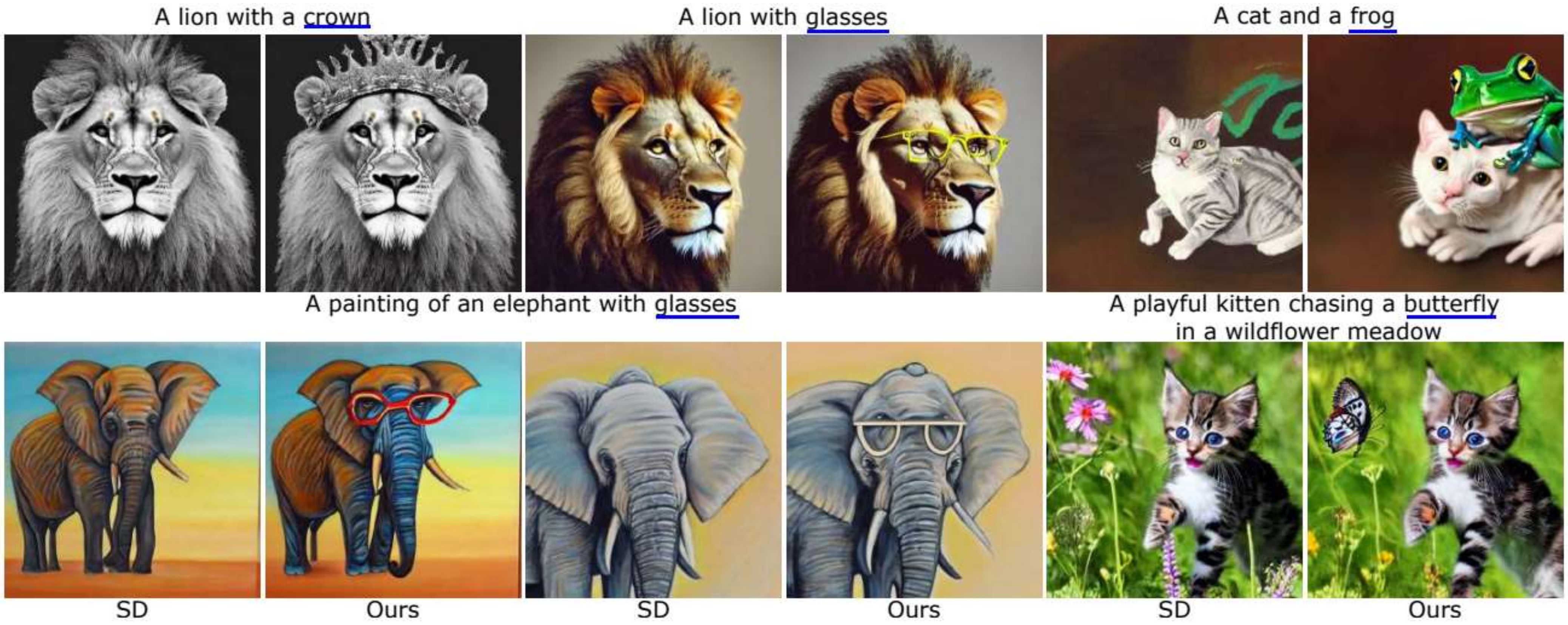


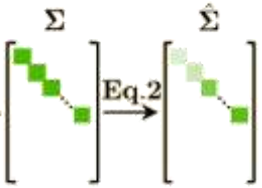


SWR: $\hat{\sigma} = \boxed{e^{-\sigma}} * \sigma.$ ⇨ $\hat{\sigma} = e^{\alpha\sigma} * \sigma$

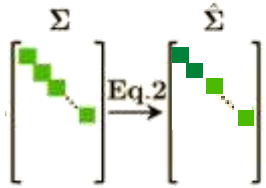


Additional results (Generating subjects for generated image)





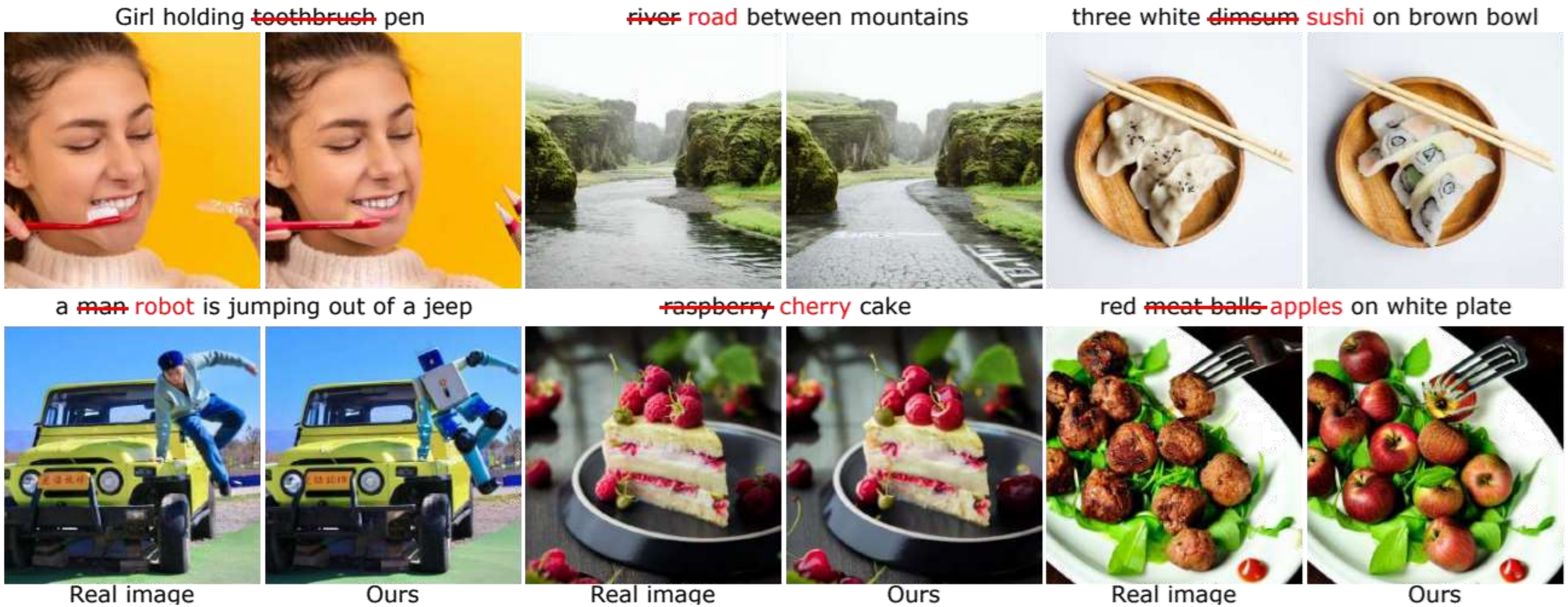
SWR: $\hat{\sigma} = \boxed{e^{-\sigma}} * \sigma.$ $\Longrightarrow$ $\hat{\sigma} = \boxed{e^{\alpha\sigma}} * \sigma$



Additional results (Adding subjects for real image)

SWR: $\hat{\sigma} = e^{\alpha\sigma} * \sigma$

Additional results (Replacing subject in the real image with another)



SWR: $\hat{\sigma} = e^{\alpha\sigma} * \sigma$

Code: https://github.com/sen-mao/SuppressEOT

Thanks