Null-text Inversion for Editing Real Images using Guided Diffusion Models

Ron Mokady[*] [1,2], Amir Hertz[*] [1,2], Kfir Aberman[1], Yael Pritch[1], and Daniel Cohen-Or[†] [1,2]
[1]Google Research
[2]The Blavatnik School of Computer Science, Tel Aviv University

MI2 Seminar

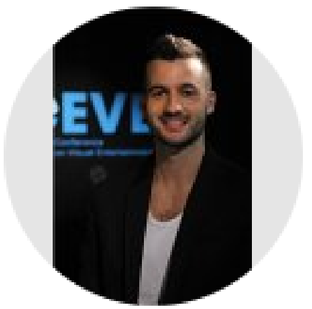
Dawid Płudowski

December 2nd, 2024

Authors

Ron Mokady*

Amir Hertz*



Kfir Aberman

Yael Pritch

Daniel Cohen-Or

Google research
600+ citations

Conference on Computer Vision and Pattern Recognition 2023

Motivation

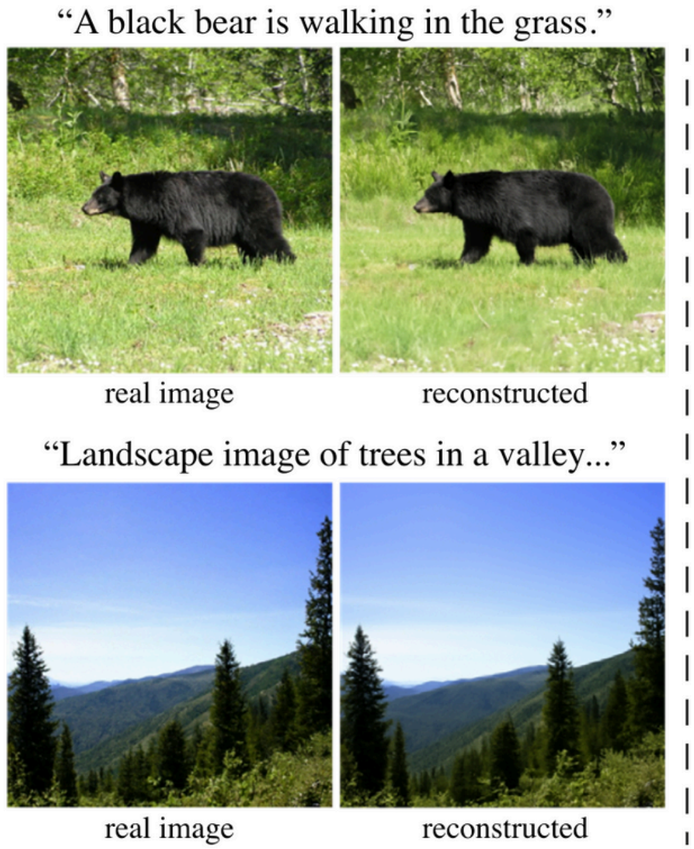
To edit a real image using these state-of-the art tools, one must first invert the image with a meaningful text prompt into the pretrained model's domain. In this paper, we introduce an accurate inversion technique and thus facilitate an intuitive text-based modification of the image.

Background I

- In 2022, main authors released *“Prompt-to-Prompt Image Editing with Cross Attention Control” which allows to make good edits of images*
- A big issue of this work is the need to inverse the real image; in the article, they were focused on artificial images mostly

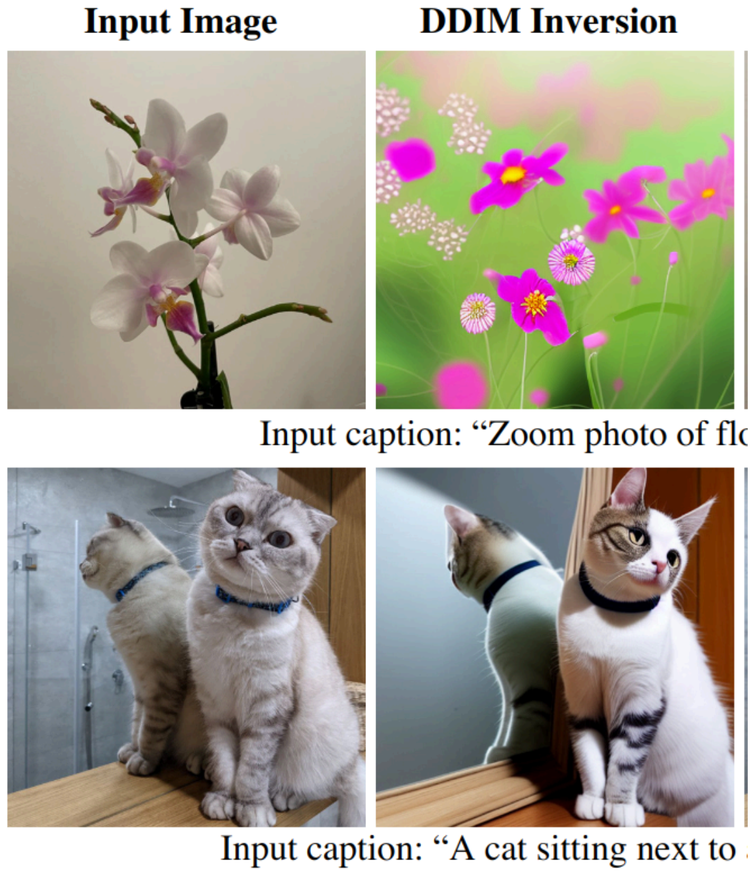
When we need to show the method works

Prompt-to-Prompt, 2022



When we need to show the method must be improved

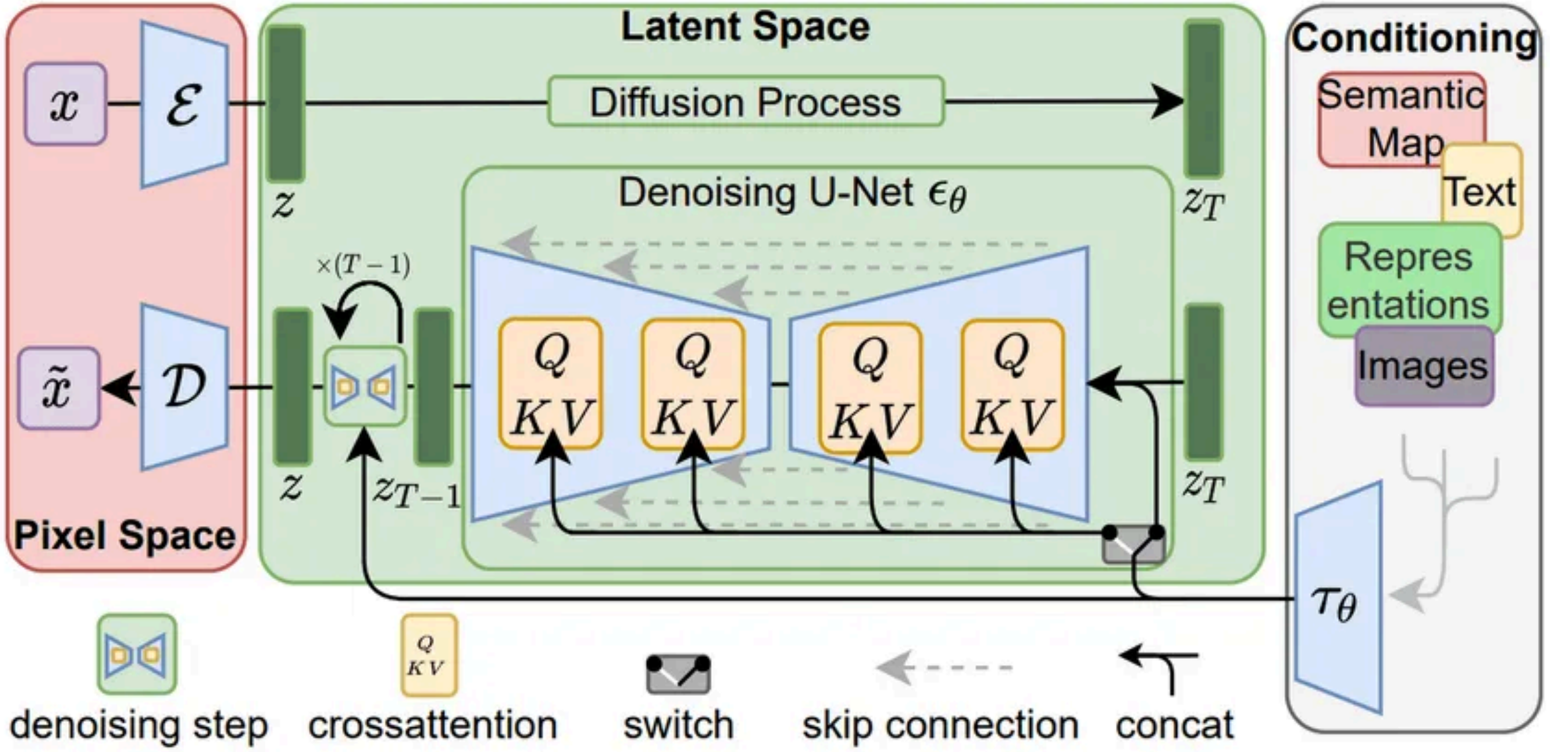
NULL-text optimization, 2022



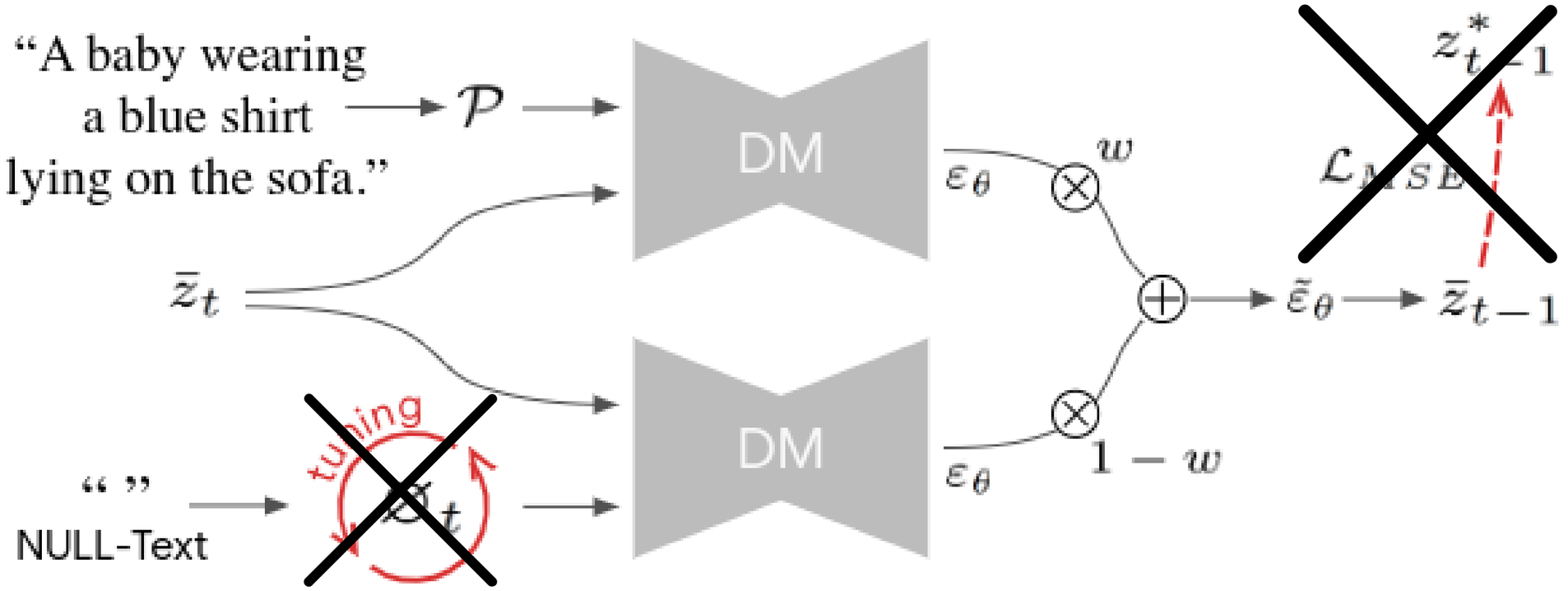
Background II

- Stable diffusion (autoencoder architecture)
- Clasiffier-free guidance

Reminder - stable diffusion



Reminder - classifier-free guidance



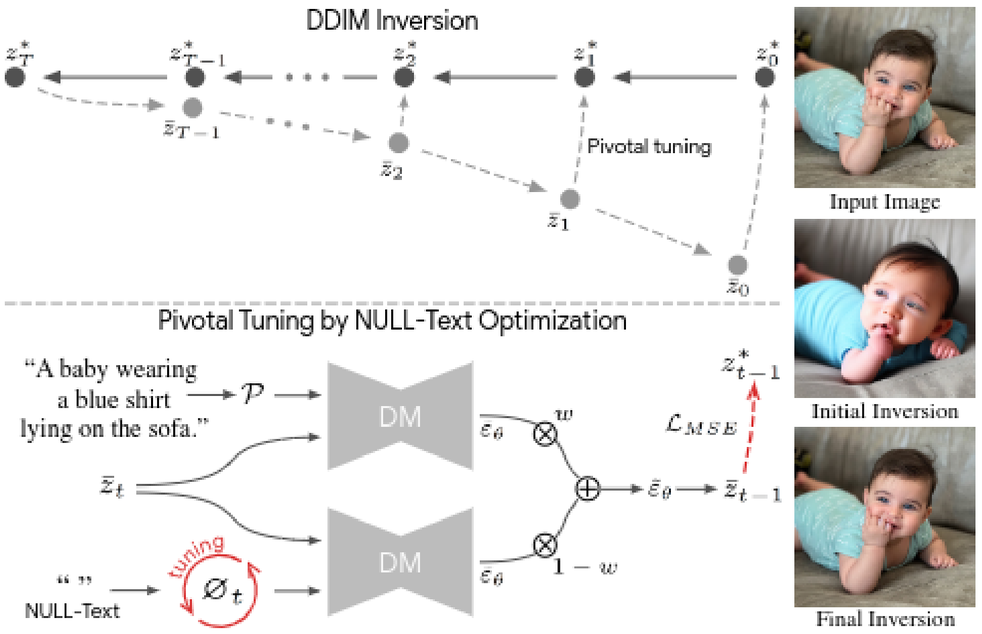


Figure 3. **NULL-text Inversion overview.** *Top: pivotal inversion. We first apply an initial DDIM inversion on the input image which estimates a diffusion trajectory* $\{z_t^*\}_0^T$. *Starting the diffusion process from the last latent* z_T^* *results in unsatisfying reconstruction as the latent codes become farther away from the original trajectory. We use the initial trajectory as a pivot for our optimization which brings the diffusion backward trajectory* $\{\bar{z}_t\}_1^T$ *closer to the original image encoding* z_0^*. *Bottom: NULL-text optimization for timestamp* t. *Recall that classifier-free guidance consists of performing the prediction* ϵ_θ *twice – using text condition embedding and unconditionally using NULL-text embedding* $\varnothing$ *(bottom-left). Then, these are extrapolated with guidance scale* w *(middle). We optimize only the unconditional embeddings* $\varnothing_t$ *by employing a reconstruction MSE loss (in red) between the predicated latent code* z_{t-1} *to the pivot* z_{t-1}^*.

Algorithm

Algorithm 1: Null-text inversion

Input: A source prompt embedding $\mathcal{C} = \psi(\mathcal{P})$ and input image $\mathcal{I}$.

Output: Noise vector z_T and optimized embeddings $\{\varnothing_t\}_{t=1}^{T}$.

Set guidance scale $w = 1$;

Compute the intermediate results $z_T^*, \ldots, z_0^*$ using DDIM inversion over $\mathcal{I}$;

Set guidance scale $w = 7.5$;

Initialize $\bar{z_T} \leftarrow z_T^*$, $\varnothing_T \leftarrow \psi(\text{""})$;

for $t = T, T-1, \ldots, 1$ **do**

  for $j = 0, \ldots, N-1$ **do**

    $\varnothing_t \leftarrow \varnothing_t - \eta \nabla_{\varnothing} \left\| z_{t-1}^* - z_{t-1}(\bar{z}_t, \varnothing_t, \mathcal{C}) \right\|_2^2$;

  end

  Set $\bar{z}_{t-1} \leftarrow z_{t-1}(\bar{z}_t, \varnothing_t, \mathcal{C})$, $\varnothing_{t-1} \leftarrow \varnothing_t$;

end

Return $\bar{z_T}$, $\{\varnothing_t\}_{t=1}^{T}$

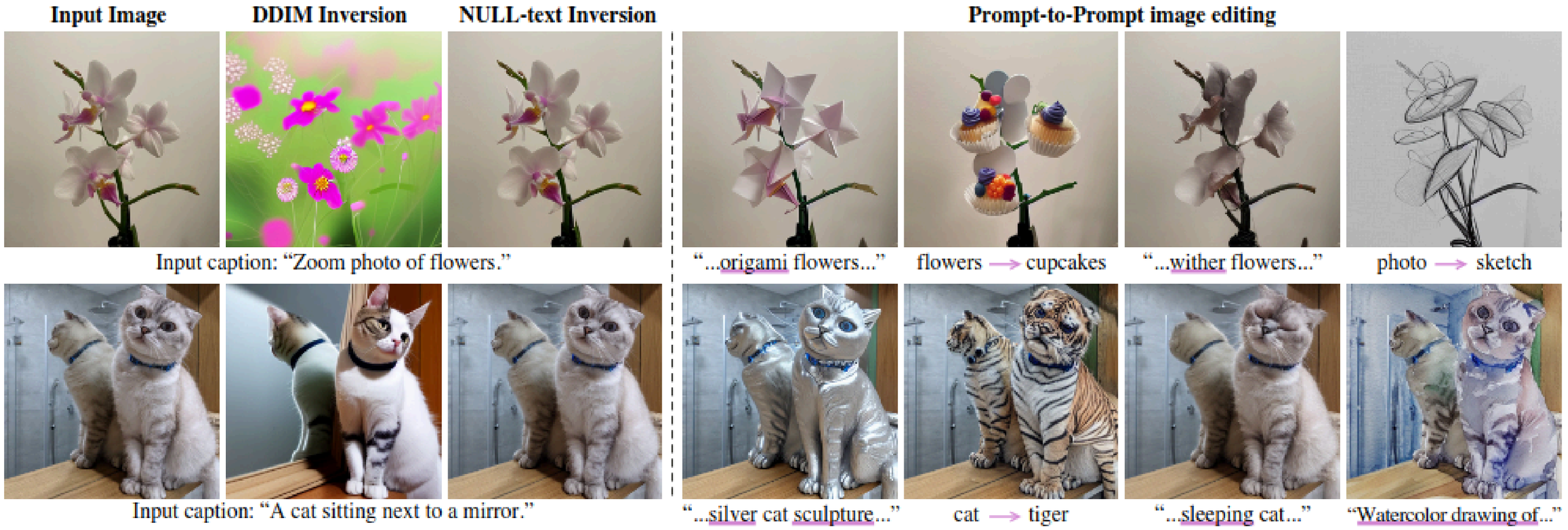


Figure 1. **Null-text inversion for real image editing.** *Our method takes as input a real image (leftmost column) and an associated caption. The image is inverted with a DDIM diffusion model to yield a diffusion trajectory (second column to the left). Once inverted, we use the initial trajectory as a pivot for null-text optimization that accurately reconstructs the input image (third column to the left). Then, we can edit the inverted image by modifying only the input caption using the editing technique of Prompt-to-Prompt [18] .*

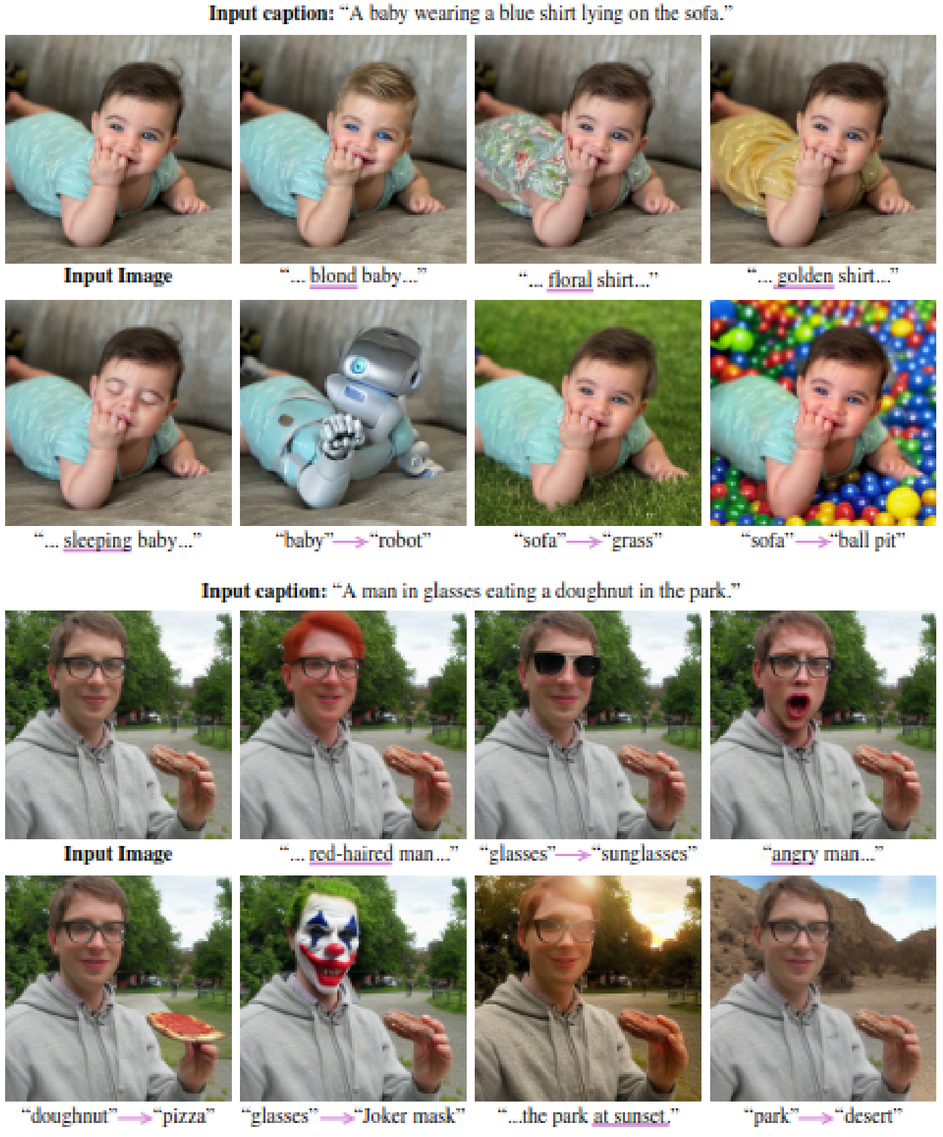


Figure 2. **Real image editing using our method.** *We first apply a single NULL-text inversion over the real input image, achieving high-fidelity reconstruction. Then, various Prompt-to-Prompt text-based editing operations are applied. As can be seen, our inversion scheme provides high fidelity while retaining high editability. See additional examples in the supplementary materials.*

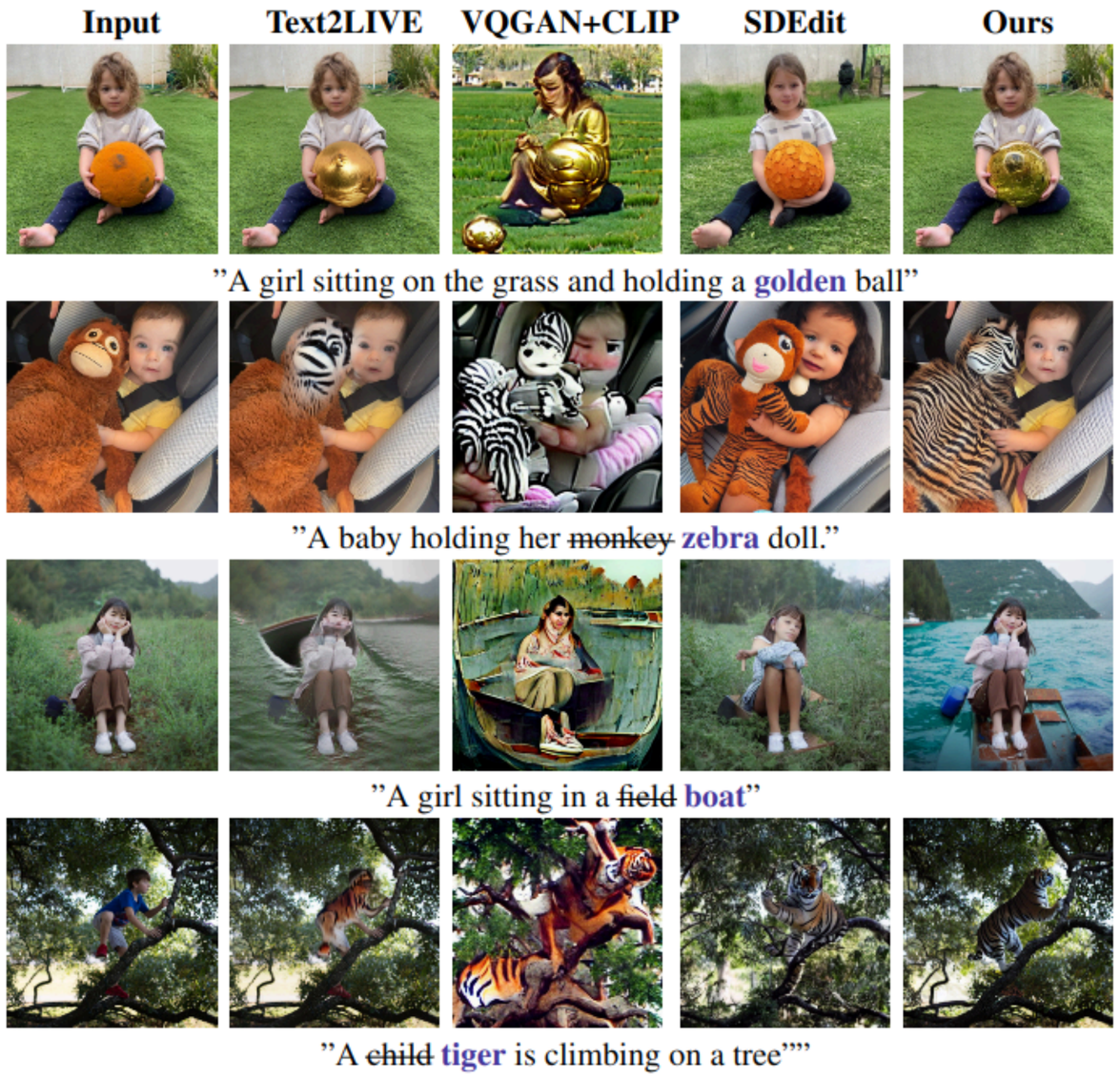


Figure 6. **Comparison.** *Text2LIVE [6] excels at replacing textures locally but struggles to perform more structured editing, such as replacing a kid with a tiger. VQGAN+CLIP [13] obtains inferior realism. SDEdit [25] fails to faithfully reconstruct the original image, resulting in identity drift when humans are involved. Our method achieves realistic editing of both textures and structured objects while retaining high fidelity to the original image.*

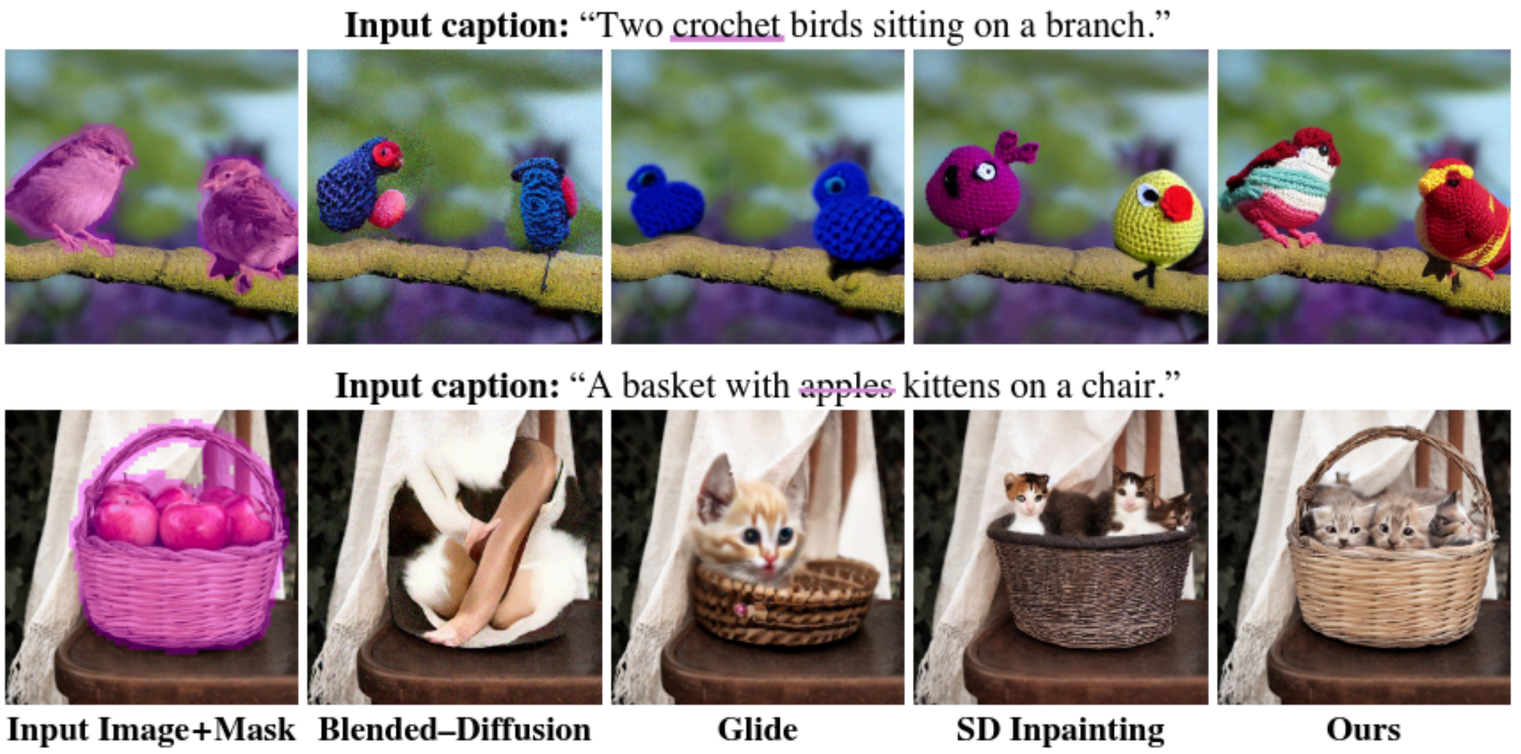
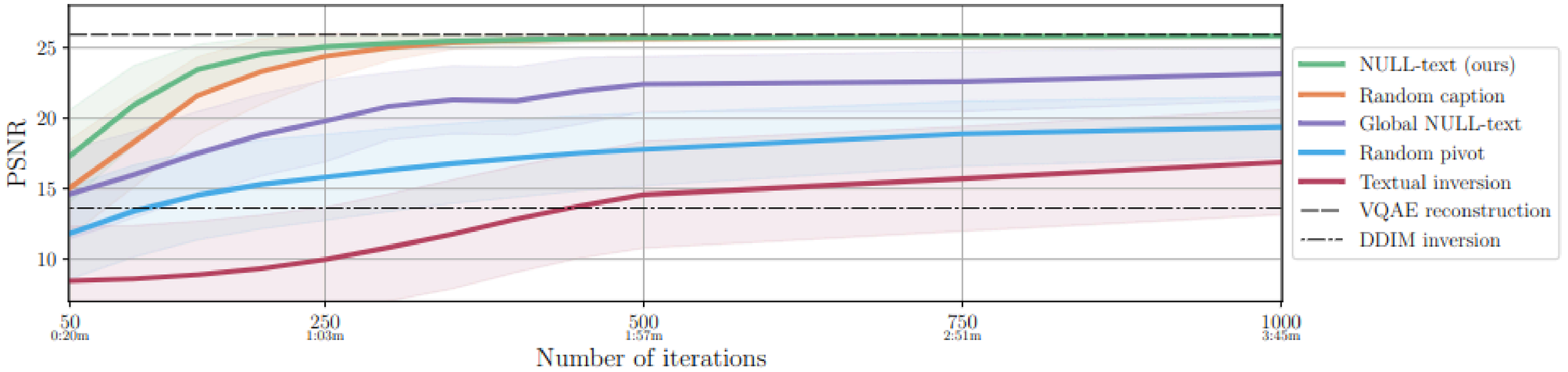


Figure 7. **Comparison to mask-based methods .** *As can be seen, mask-based methods do not require inversion as the region outside the mask is kept. However, unlike our approach, such methods often struggle to preserve details that are found inside the masked region. For example, basket size is not preserved.*

Evaluation



Input caption: "A black dinning room table sitting in a yellow dinning room."

Figure 4. **Ablation Study.** *Top: we compare the performance of our full algorithm (green line) to different variations, evaluating the reconstruction quality by measuring the PSNR score as a function of number optimization iterations and running time in minutes. Bottom: we visually show the inversion results after* 200 *iterations of our full algorithm (on right) compared to other baselines. Results for all iterations are shown in the supplementary materials.*

Evaluation

Table 2. Quantitative evaluation on ImageNet.

	SDEdit [25]	Text2Live [6]	FlexIT [10]	Ours
LPIPS ↓	31.5	**20.0**	24.7	29.1
Acc.% ↑	36.5	47.5	51.3	**61.1**

Table 1. **User study results.** *The participants were asked to select the best editing result in terms of fidelity to both the input image and the textual edit instruction.*

VQGAN+CLIP	Text2Live	SDEDIT	Ours
3.8%	16.6%	14.5%	**65.1%**

Prompt-to-Prompt Image Editing with Cross Attention Control

Amir Hertz*[1,2], Ron Mokady*[1,2], Jay Tenenbaum[1], Kfir Aberman[1], Yael Pritch[1], and Daniel Cohen-Or*[1,2]

[1] *Google Research*
[2] *The Blavatnik School of Computer Science, Tel Aviv University*

Prompt-to-Prompt

Figure 1: Our method provides variety of *Prompt-to-Prompt* editing capabilities. The user can tune the level of influence of an adjective word (top-left), replace items in the image (top-right), specify a style for an image (bottom-left), or make further refinements over the generated image (bottom-right). The manipulations are infiltrated through the cross-attention mechanism of the diffusion model without the need for any specifications over the image pixel space.

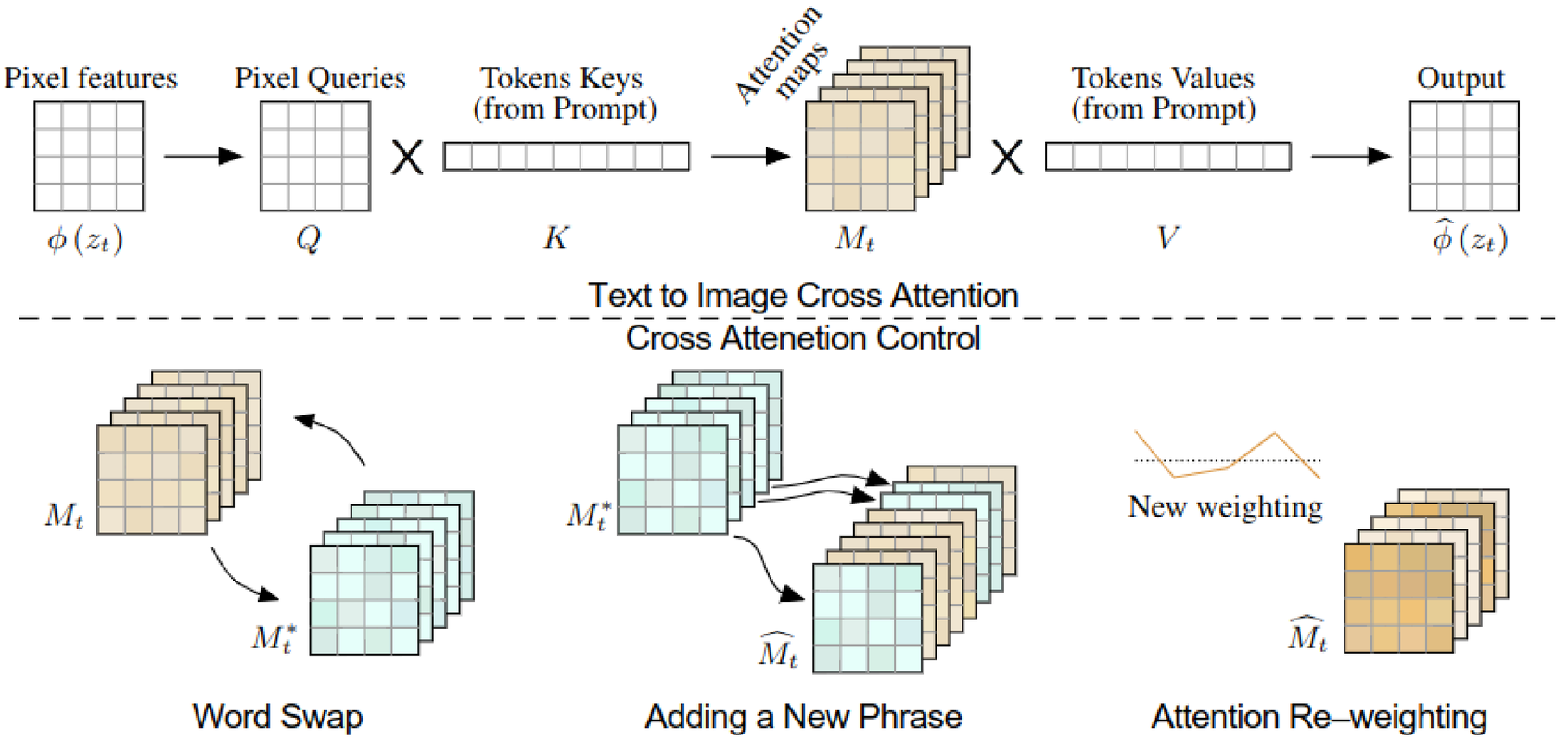


Figure 3: Method overview. Top: visual and textual embedding are fused using cross-attention layers that produce spatial attention maps for each textual token. Bottom: we control the spatial layout and geometry of the generated image using the attention maps of a source image. This enables various editing tasks through editing the textual prompt only. When swapping a word in the prompt, we inject the source image maps M_t, overriding the target image maps M_t^*, to preserve the spatial layout. Where in the case of adding a new phrase, we inject only the maps that correspond to the unchanged part of the prompt. Amplify or attenuate the semantic effect of a word achieved by re-weighting the corresponding attention map.

$$M = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right),$$

Algorithm 1: Prompt-to-Prompt image editing

Input: A source prompt $\mathcal{P}$, a target prompt $\mathcal{P}^*$, and a random seed s.
Output: A source image x_{src} and an edited image x_{dst}.
$z_T \sim N(0, I)$ a unit Gaussian random variable with random seed s;
$z_T^* \leftarrow z_T$;
for $t = T, T-1, \ldots, 1$ **do**
    $z_{t-1}, M_t \leftarrow DM(z_t, \mathcal{P}, t, s)$;
    $M_t^* \leftarrow DM(z_t^*, \mathcal{P}^*, t, s)$;
    $\widehat{M}_t \leftarrow Edit(M_t, M_t^*, t)$;
    $z_{t-1}^* \leftarrow DM(z_t^*, \mathcal{P}^*, t, s_t)\{M \leftarrow \widehat{M}_t\}$;
end
Return (z_0, z_0^*)

word swap

$$Edit(M_t, M_t^*, t) := \begin{cases} M_t^* & \text{if } t < \tau \\ M_t & \text{otherwise.} \end{cases}$$

adding a new phrase

$$\left(Edit\left(M_t, M_t^*, t\right)\right)_{i,j} := \begin{cases} (M_t^*)_{i,j} & \text{if } A(j) = None \\ (M_t)_{i,A(j)} & \text{otherwise.} \end{cases}$$

attention re-weighting

$$\left(Edit\left(M_t, M_t^*, t\right)\right)_{i,j} := \begin{cases} c \cdot (M_t)_{i,j} & \text{if } j = j^* \\ (M_t)_{i,j} & \text{otherwise.} \end{cases}$$

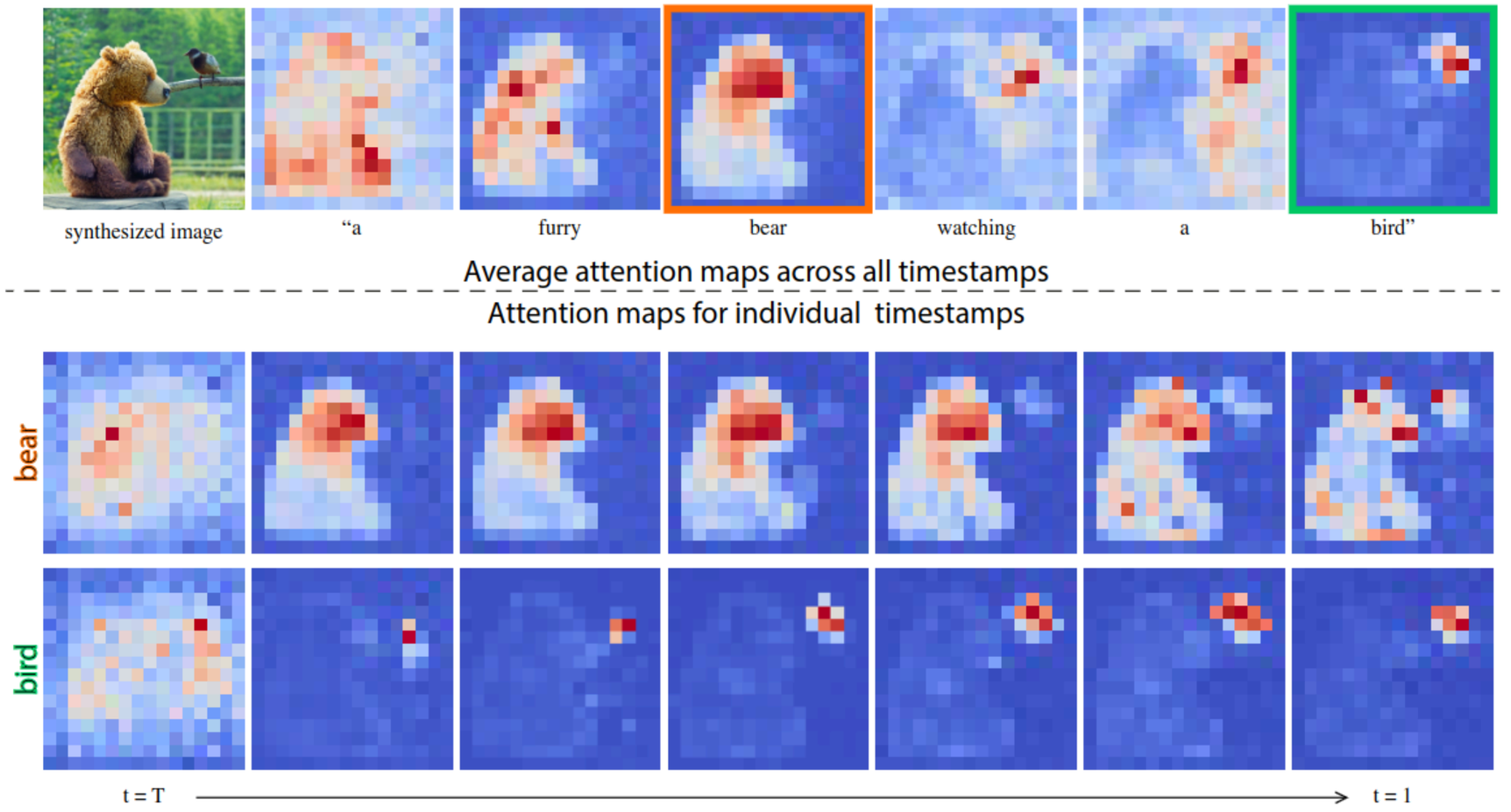


Figure 4: Cross-attention maps of a text-conditioned diffusion image generation. The top row displays the average attention masks for each word in the prompt that synthesized the image on the left. The bottom rows display the attention maps from different diffusion steps with respect to the words “bear” and “bird”.

Questions?