ReFACT: Updating Text-to-Image Models by Editing the Text Encoder

Dana Arad* **Hadas Orgad*** **Yonatan Belinkov†**

Technion - Israel Institute of Technology
{danaarad@campus., orgad.hadas@cs., belinkov@}technion.ac.il

Abstract

Text-to-image models are trained on extensive amounts of data, leading them to implicitly encode factual knowledge within their parameters. While some facts are useful, others may be incorrect or become outdated (e.g., the current President of the United States). We introduce ReFACT, a novel approach for editing factual knowledge in text-to-image generative models. ReFACT updates the weights of a specific layer in the text encoder, only modifying a tiny portion of the model's parameters, and leaving the rest of the model unaffected. We empirically evaluate ReFACT on an existing benchmark, alongside RoAD, a newly curated dataset. ReFACT achieves superior performance in terms of generalization to related concepts while preserving unrelated concepts. Furthermore, ReFACT maintains image generation quality, making it a valuable tool for updating and correcting factual information in text-to-image models. [3]

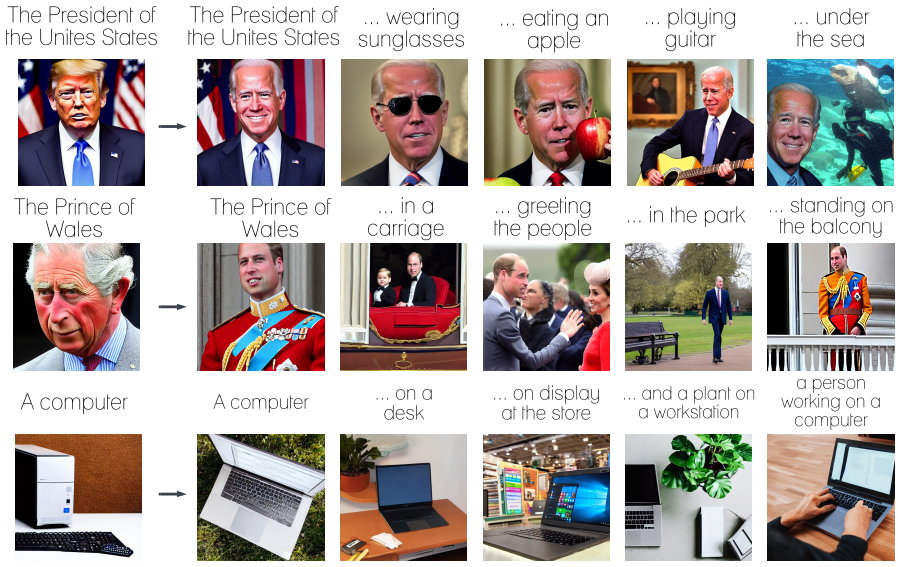


Figure 1: ReFACT can be used to edit knowledge in text-to-image models using an editing prompt and a target prompt (e.g., "The President of the United States" and "Joe Biden"), such that the edit is generlizable to other related prompts.

*Equal contribution.

†Supported by the Viterbi Fellowship in the Center for Computer Engineering at the Technion.

[3]Our code and data are available at https://github.com/technion-cs-nlp/ReFACT

Preprint. Under review.

1 Introduction

We live in a rapidly-changing world where advancements in technology, globalization, and political and social shifts occur at an unprecedented pace. Concurrently, text-to-image generative models, which have gained immense popularity [1, 2, 3, 4], possess the capacity to encode real-world knowledge within their parameters derived from the training data. While some facts are static, such as important historical figures, other facts may be inaccurate or become outdated, such as the current United States President (see Figure 1). Therefore, there is a need to update text-to-image models to reflect the current state of the world. As updating models by retraining them is both costly and may introduce undesired changes, efficient methods for making targeted updates are in dire need.

Current text-to-image models are pipelines composed of several individual modules. Common architectures consist of a text encoder—used to generate latent representations of an input prompt– an image generation module, and a cross-attention module that connects the two modalities. Orgad et al. recently proposed TIME, a method for editing text-to-image models by targeting the cross-attention layers.

In this work we present ReFACT: a new method for **R**evising **FACT**ual knowledge in text-to-image models. ReFACT views facts as key–value pairs encoded in linear layers of the text encoder and updates the weights of a specific layer using a rank one editing approach [6]. The edit consists of replacing the value ("Donald Trump $\rightarrow$ "Joe Biden") for a corresponding key ("United States President"), and thus does not require fine-tuning the model. ReFACT modifies only a tiny portion of the model's parameters (0.24%), far fewer than the previous method, TIME (1.95%). Our method takes as input a single prompt, representing the desired edit (e.g., "The President of the United States") and a representation of the up-to-date fact, which can be either a text ("Joe Biden") or an image (of Joe Biden).

ReFACT is able to generalize to closely related prompts while preserving unrelated concepts. Notably, ReFACT does not affect the general quality of generated images. For example, Stable Diffusion [4] generates an image of Donald Trump when prompted with the text "The President of the United States" (See Figure 1). After applying ReFACT, the edited text encoder generates representations that reflect the edited fact, enabling the model to generate images of Joe Biden.

We evaluate ReFACT on the TIME dataset [5], a benchmark for evaluating the editing of implicit model assumptions on specific attributes (e.g., editing roses to be blue instead of red). Furthermore, we curate a new dataset, RoAD, the **Ro**les and **A**ppearances **D**ataset, for editing additional types of factual knowledge. We show that ReFACT can be used to edit a wide range of factual knowledge types, demonstrates high generalization, and does not hurt the representations of unrelated facts.

Overall, our method is a significant improvement in text-to-image model editing. Our code and data are publicly available.[4]

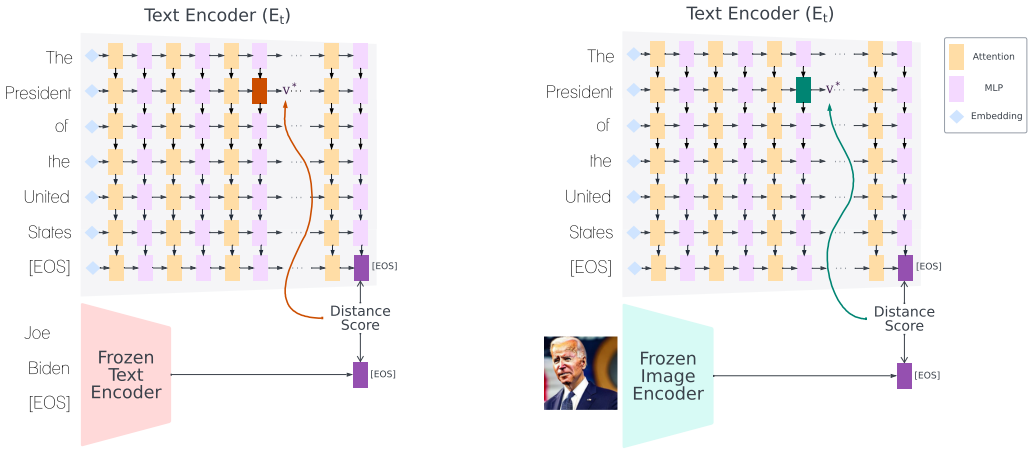
2 Related work

The task of image editing using diffusion models has been the focus of several recent studies [7, 8, 9, 10, 11, 12, 13]. Image editing aims to modify specific attributes of an input image based on some auxiliary inputs, recently using texts and instructions [14, 15, 16]. Another closely related line of work is personalization of text-to-image diffusion models, where the goal is to adapt the model to a specific individual or object [17, 18]. Personalization in text-to-image models allow the model to better generate a specific face, object, or scene, given a specific word or pseudo-word [19, 20, 21, 22].

Our work focuses on a fundamentally different task: editing the world knowledge of a text-to-image model using a target description, be it a text prompt or an image. Knowledge editing aims for the complete transformation of facts, which can be retrieved by different words and phrases, and is not restricted to the specific input image, word or pseudo-word.

The task of knowledge editing in text-to-image models was introduced by Orgad et al. [5], who targeted the cross-attention layers. In contrast, we target a specific layer in the text encoder of the text-to-image model, allowing a more precise edit that changes fewer model parameters (0.24% compared to 1.95% of the model parameters).

[4] `https://github.com/technion-cs-nlp/ReFACT`



(a) Editing with textual embedding.

(b) Editing with image embedding.

Figure 2: ReFACT receives an edit prompt and a target representing the desired change. We obtain the representation of the target, which can be a text or an image, by passing it through the respective frozen CLIP encoder and taking the output at the [EOS] token. Then, we compute a vector v^* that, when inserted in a specific layer, will reduce the distance between the edit prompt representation and the target representation, resulting in the insertion of the edited fact into the model.

Editing knowledge embedded within deep neural networks has been the focus of several lines of work, achieving success in editing generative adversarial networks [23, 24, 25], image classifiers [26], and large language models (LLMs) [27, 28, 29]. Several methods were proposed to update weights in LLMs in particular, including fine-tuning on edited facts [30], weight predictions using hyper-networks [31], identifying and editing specific neurons [32], and rank one model editing [6]. In this work, we propose a rank one editing method for text-to-image models. Our method targets the text encoder, which is the component of text-to-image models responsible for creating latent representations of input prompts.

3 Method

3.1 Background

Diffusion models generate images by gradually removing noise from noisy samples until a clean image is obtained. Text-to-image diffusion models [4, 3, 33] are conditioned on a textual prompt that guides the image generation process towards generating desired elements. Several text-to-image diffusion models utilize CLIP [34] in different capacities, specifically as a popular choice for a multi-modal-aware text encoder.

CLIP consists of a text encoder and a image encoder, jointly trained to create a shared embedding space for images and texts. Concretely, a special end of sequence token, denoted [EOS], is appended at the end of each input. CLIP is trained contrastively to increase the cosine similarity between [EOS] tokens of corresponding images and texts, and decrease the similarity between unrelated inputs.

CLIP's text encoder is a GPT-2-style model [35] trained from scratch. Like GPT-2, CLIP's text encoder implements a causal (unidirectional) attention mechanism, meaning that the [EOS] is the only token able to aggregate information from all other tokens in the sequence. Thus, the [EOS] token can be used to optimize the insertion of new facts, as key–value pairs, into the model.

3.2 ReFACT

This section describes ReFACT, a method for editing text-to-image models by changing the text encoder, denoted as E_t. Since the image generation process is conditioned on the representations

produced by E_t, editing the knowledge of E_t should be reflected in the generated images. At a high level, ReFACT takes an edit prompt (e.g., "The President of the United States") and a target text ("Joe Biden") or a target image (an image of Biden) that reflects the desired edit, and edits a specific layer in the model. The goal is to make the model's representation of the prompt similar to that of the target text/image. The process is illustrated in Figure 2.

To achieve this, ReFACT targets the multi-layer perceptron (MLP) layers in the text encoder. Each MLP has two matrices with a non-linearity between them: $W_{proj} \cdot \sigma(W_{fc})$. Following previous work, we view W_{proj} as a linear associative memory [36, 37, 6]. Linear operations can therefore be viewed as a key–value store $WK \approx V$ for a set of key vectors K and corresponding value vectors V in a specific layer l. For example, a key is a representation of "The President of the United States", and the value is the identity of the president, which is "Donald Trump" prior to editing.

Concretely, we perform a rank-one edit of the relevant layer weights [27], $W^{(l)}_{proj}$, to insert a new key value pair (k^*, v^*). Bau et al. [23] suggested a closed form solution:

$$\text{minimize } \|\hat{W}K - V\| \text{ such that } \hat{W}k_* = v_* \quad \text{by setting } \hat{W} = W + \Lambda(C^{-1}k_*)^T. \tag{1}$$

where $C = KK^T$ is a pre-cached constant estimated on wikipedia text and $\Lambda = (v_* - Wk_*)/(C^{-1}k_*)^T k_*$. Based on this, all we need is to specify k^* and v^* .

The computation of both k^* and v^* relies on the hypothesis that the factual knowledge is retrieved by the model at the last subject token (e.g, "President") [6].

Choosing k^*: We obtain hidden states from layer l for a set of prompts containing the subject ("The President of the United States", "An image of the President of the United States", etc.). k^* is taken as the average representation of the last subject token in each of the prompts. This is done to achieve a more general representation of last subject token, which is not dependant on specific contexts.

Choosing v^*: We denote by x_1 the input text of the edit prompt ("The President of the United States"), and the target by t^* ("Joe Biden" or an image of Joe Biden). We pass the target t^* through the encoder of the relevant modality, E_t or E_i, denoted simply as E, and take the [EOS] representation as the target representation $E(t^*)$. Then, we similarly compute the representation of other texts $x_2, ..., x_N$, which contain the source prompt ("Donald Trump") as well as other unrelated prompts ("A cat"), as negative examples.[5] We then seek a v^* that, when substituted as the output of MLP layer l at token i (the last subject token, "president"), yields a representation that is close to that produced by an unedited encoder given the target ("Joe Biden"), while being far from negative examples.

Formally, denote by $E_{m_i^{(l)}:=v}$ the text or image encoder where the output of layer l at token i was substituted with v. We always use the last subject token [6], and thus sometimes omit the subscript i for ease of notation. To obtain the desired v^*, we minimize the following contrastive loss:

$$v^* = \underset{v}{\text{argmin}} \frac{\exp(d(E(t^*), E_{m^{(l)}:=v}(x_1)))}{\sum_{j=1}^{N} \exp(d(E(t^*), E_{m^{(l)}:=v}(x_j)))} \tag{2}$$

where $d(\cdot, \cdot)$ is the L_2 distance.[6]

Once v^* and k^* are obtained, we edit the relevant layer l using the solution from Equation (1).

4 Experiments

4.1 Datasets

We evaluate our method on the TIME dataset [5], a dataset for the evaluation of editing implicit assumptions in text-to-image models, like changing the default color of roses to blue instead of red.

To perform a more comprehensive evaluation of factual knowledge editing in text-to-image models, we introduce **RoAD**, the **Ro**les and **A**ppearances **D**ataset. RoAD contains 100 entries that encompass a diverse range of roles fulfilled by individuals, such as politicians, musicians, and pop-culture

[5] We also experimented with a direct optimization without negative examples; see Appendix A.

[6] For other variations refer to Appendix A.

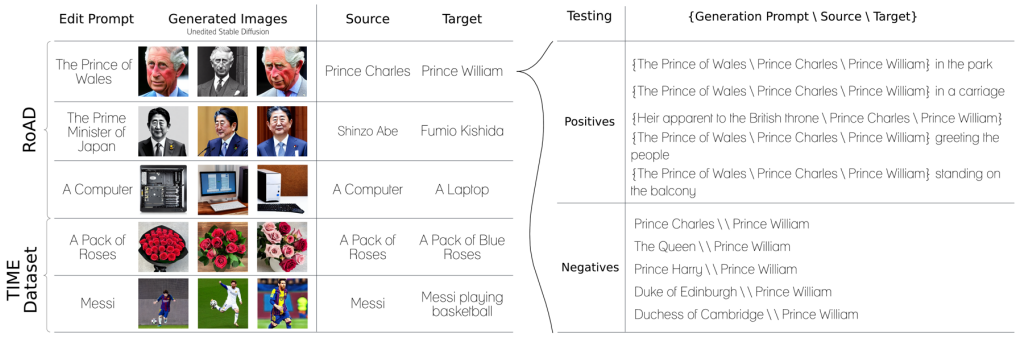


Figure 3: Samples from the two datasets, TIME dataset and RoAD. TIME dataset contains editing of implicit model assumptions while RoAD targets a general visual appearance of the edited subject. Each entry of RoAD contains five positive generation prompts and five negative generation prompts, used for evaluation.

characters, as well as variations in the visual appearance of objects and entities. Each entry describes a single edit, and contains the edit prompt (e.g., "The Prince of Wales"), a source prompt (e.g., "Prince Charles"), and a target prompt (e.g., "Prince William").

Moreover, each entry contains five positive prompts and five negative prompts. Positive prompts are meant to evaluate the ability of the editing algorithm to generalize to closely related concepts (e.g., "The Prince of Wales in the park"). Negative prompts are used to ensure that other similar but unrelated concepts remain unchanged (e.g., "Prince Harry"). See Figure 3 for samples from the two dataset, and Appendix B for more details.

4.2 Experimental setup

We implement our method on the publicly available implementations of Stable Diffusion V1-4 [4], and CLIP [34] available on HuggingFace [38]. We compare our method to TIME, another editing method which targets the cross-attention layers [5]. Applying TIME to our dataset RoAD required some modifications, as TIME does not support all edits in RoAD as is. We implemented a few variations, discussed in Appendix G. Moreover, we follow Orgad et al. [5] and compare our method to an oracle model — an unedited model that receives the target prompt as input for the positive examples and the negative prompts for the negative examples — and a baseline model which is an unedited model that receives the source prompts for all generations. We evaluate our model on two datasets, TIME dataset and our newly curated dataset RoAD. For each dataset, we perform a hyper-parameters search over the validation set. This also includes the choice of layer to edit. Further details are in Appendix A.

4.3 Metrics

To measure our methods' utility, we follow Meng et al. [39] and Orgad et al. [5] and focus on efficacy, generalization, and specificity. We use 25 random seeds, editing a clean model in each setting and generating one image per prompt for the given seed. We then compute each of the metrics using CLIP as a zero-shot classifier, [7] as described below, and average over the different seeds.

Efficacy: Quantifies how effective an editing method is on the prompt that was used to perform the edit. For example, when editing "The Prince of Wales" from "Prince Charles" to "Prince William" (see Figure 3), efficacy measures how many of the images generated using the prompt "the Prince of Wales" successfully generate an image of Prince William. For a single edit, efficacy is 1 if $\text{CLIP}(\text{target\_prompt}) > \text{CLIP}(\text{source\_prompt})$, and 0 otherwise.

Generalization: Quantifies how well an editing method generalizes to related prompts. For example, "the prince of Wales in the park". Generalization is calculated as the portion of related

[7] We use Laion's ViT-G/14 [40], which is the best open source CLIP model to date.

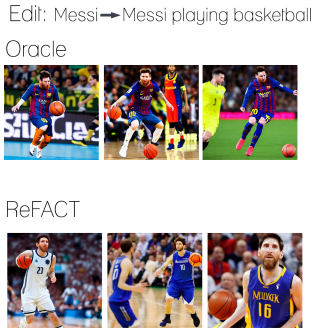


Figure 4: Efficacy of ReFACT. In some cases our method is more effective than the oracle.

Figure 5: Specificty of ReFACT. Our method is able to precisely edit specific concepts without affecting related concepts or other elements in the generated image.

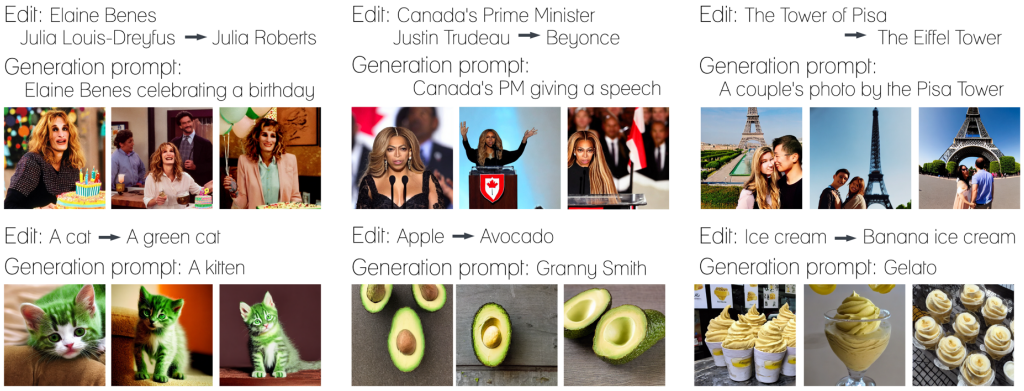


Figure 6: ReFACT is able to generalize to related prompts.

prompts (Positives in Figure 3) for which the editing was successful. As with efficacy, an edit is successful if CLIP(target_prompt) $>$ CLIP(source_prompt).

Specificity: Quantifies how specific an editing method is. Specificity is calculated as the portion of unrelated prompts (Negatives in Figure 3) that were not affected by the editing. A prompt is unaffected if CLIP(target_prompt) $<$ CLIP(source_prompt).

We also compute the geometrical mean of the generalization and specificity scores (denoted **F1**). In addition, to test the effect of ReFACT on the overall quality of the model's image generation process, we measure the FID score [41], as well as the CLIP score [42] over the MS-COCO validation dataset [43], as is standard practice [4, 44, 3, 45].

5 Results

5.1 Qualitative evaluation

As we show in Figure 4, ReFACT is effective in editing the knowledge represented by the edit prompt. Sometimes, it is even more effective than the oracle: When Messi is edited to play basketball, the oracle often adds a basketball to a soccer game. ReFACT, on the other hand, modifies the entire setting, including the court, and sometimes the uniform. Figure 5 demonstrates how ReFACT is able to alter specific knowledge while leaving other unrelated but close prompts unchanged. After editing an apple to appear as an avocado, when the edited model is prompted with "An apple and a lemon", it successfully generates images showcasing both fruits. The generality of ReFACT to other related words and phrasings is demonstrated in Figure 6. For instance, after editing "Canada's Prime

Figure 7: TIME and ReFACT, demonstrated on failure cases of TIME.

Table 1: Evaluation of editing methods on TIME and RoAD test sets. Best results are marked with **bold**. Best results among editing methods (TIME, ReFACT) are marked with underline.

Dataset	Method	Efficacy (↑)	Generalization (↑)	Specificity (↑)	F1 (↑)	FID (↓)	CLIP (↑)
TIME Dataset	Baseline	04.27% ±2.24	06.21% ±0.91	**95.68%** ±1.18	24.37	12.67	26.50
	Oracle	97.04% ±2.35	**93.26%** ±1.47	**95.68%** ±1.18	**94.46**	12.67	26.50
	TIME	83.23% ±3.65	64.08% ±1.66	75.95% ±2.34	69.76	12.10	26.12
	ReFACT	**98.19%** ±1.13	88.02% ±1.15	79.18% ±1.98	83.48	12.48	26.44
RoAD	Baseline	01.15% ±0.91	03.76% ±0.81	**99.36%** ±0.33	19.32	12.67	26.50
	Oracle	**98.13%** ±1.12	**96.68%** ±0.85	**99.36%** ±0.33	**98.01**	12.67	26.50
	TIME	52.18% ±3.86	42.74% ±2.17	75.36% ±1.57	56.75	17.56	26.42
	ReFACT	93.38% ±1.59	86.80% ±0.98	95.44% ±0.53	91.01	12.47	26.48

Minister" to be Beyonce, the model successfully generates images of Beyonce giving a speech in front of the Canadian flag for the prompt "Canada's PM giving a speech". For additional qualitative results, see Appendix E.

We show several comparisons with TIME [5] in Figure 7. ReFACT is able to edit cases where TIME essentially fails and hurts the model's generalization capabilities (editing "Cauliflower" to "Leek"). ReFACT is also able to generalize in cases where TIME does not (editing "a pedestal" to "a wooden pedestal" generalizes also in "a pedestal in the garden"), and keep generations for unrelated prompts unchanged (editing "ice cream" to "strawberry ice cream" does not affect the color of ice).

5.2 Quantitative evaluation

Table 1 presents results on two datasets: the TIME dataset [5] and RoAD. ReFACT achieves better efficacy, generality, and specificity on both datasets, compared to the previous editing method. On the TIME dataset, our method achieves superior efficacy, on-par with the oracle. It also achieves significantly better generalization than TIME, and better specificity, albeit not as high as the oracle. On RoAD, ReFACT achieves significantly better performance across all of the metrics.

Importantly, ReFACT does not hurt the image generation capabilities of the model, as demonstrated by excellent FID and CLIP scores on both datasets (virtually identical to the unedited model's). In contrast, when TIME is used to edit entries from RoAD, we find that it sometimes results in an unwanted outcome where the images generated by the model are not coherent anymore (Figure 7, left). This is also reflected in the higher FID score.

5.3 Failure cases

While ReFACT is very effective at modifying specific attributes and can generalize very well, in some cases it modifies other attributes of an object as well. This is crucial in images of people's faces, where a change in a facial feature changes the identity of the person discussed (see Figure 8a). While

(a) Editing facial features of people. ReFACT might edit unintended features, unlike TIME, which demonstrates a more nuanced change.

(b) Specificity failure cases: Concepts that should not be affected by the edit are changed in an undesirable manner.

Figure 8: Failure cases of ReFACT.

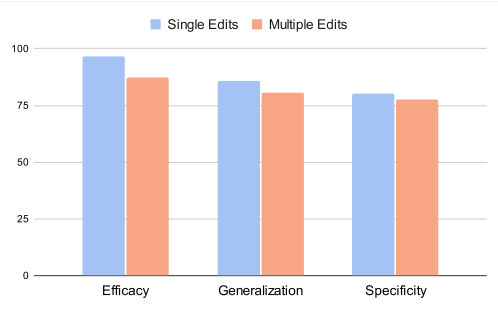


Figure 9: Comparison of multiple edits using ReFACT versus a single method per edit. Metrics presented are mean values computed on 90 samples taken from both datasets.

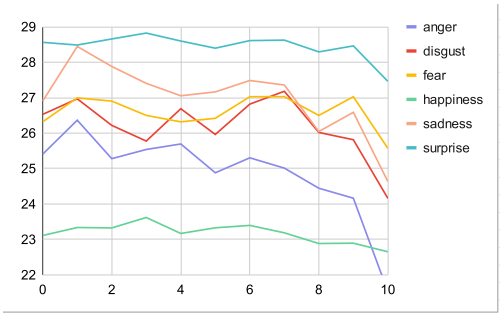


Figure 10: Clip score of different emotions on the generated images after editing to generate images that express this emotion (target for editing is an image). Deeper layers are less effective for editing expressed emotions.

ReFACT performed the desired edit, it excessively changed the person's face, unlike TIME, which better preserved facial features. In addition, ReFACT still incurs some failure cases in specificity, as demonstrated in Figure 8b.

5.4 Multiple edits

Our main experiments with ReFACT edited one piece of information at a time. To assess ReFACT's ability to edit multiple facts, we perform sequential edits. We alternate on entries from the TIME dataset and RoAD, editing 90 facts in total. As Figure 9 shows, sequential edits work just as well as single edits in all three metrics. See Appendix I for additional results. These encouraging results show that ReFACT may be useful in practice. Future work may scale it up by performing simultaneous edits, similar to [27].

6 Per-layer analysis

So far, we edited a particular layer for all facts, which was selected using the validation set. However, we hypothesize that different layers encode distinct features. To investigate differences among

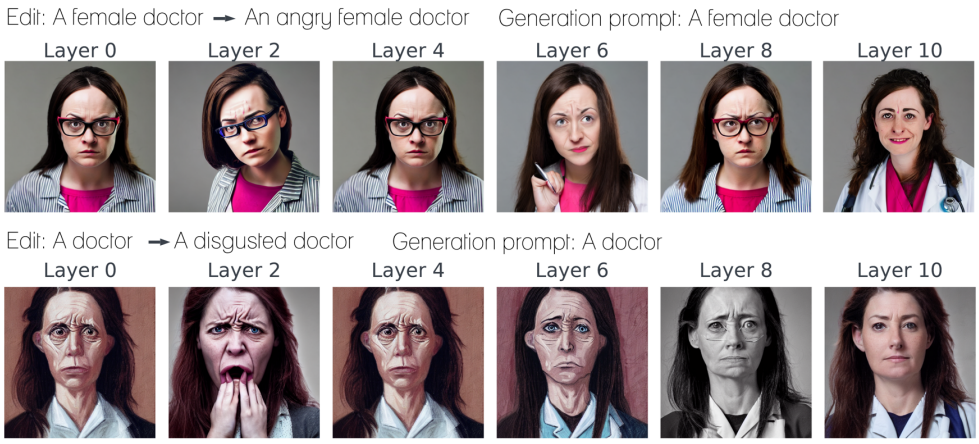


Figure 11: Results of editing various emotions on different layers. Emotions are less visible in the generated image as we edit deeper layers.

different layers in the text encoder, we employ ReFACT as a causal analysis tool, editing individual layers and observing the corresponding outcomes. We focus here on facial expressions, as an initial demonstration for using editing methods to analyze the internal mechanisms of deep models.

Experiment. We use both versions of ReFACT in this section, using either the image encoder or the text encoder to get the target embedding (Section 3.2). We use six "universal" emotions [46] (happiness, sadness, anger, fear, disgust, and surprise), and edit the model using a target image or text of people expressing the different emotions (generated by an unedited model). We edit each layer and generate 50 images for each emotion, 25 of females and 25 of males. For further details, see Appendix H.

Results. Editing lower layers tends to affect the emotions in the generated images more than editing deeper layers, as demonstrated in Figure 11. Moreover, we evaluate the CLIP score of the generated images w.r.t. the edited emotion (e.g., the text "anger"). If an edit is successful in preserving the emotion, the CLIP score should be high. As Figure 10 shows, CLIP scores for the edited emotion decrease as the edited layer is higher. In other words, editing lower layers is generally more effective. These results indicate that emotions are more encoded in the lower layers of the text encoder.

7 Discussion

In this work, we presented ReFACT, an editing method that modifies knowledge embedded in text-to-image models without fine-tuning. ReFACT is effective at editing various types of knowledge, such as implicit model assumptions or the appearance of an entire subject. Its edits are specific, leaving other pieces of knowledge unchanged. We also demonstrated how editing can be used as a causal analysis tool for analyzing which information is stored in different layers.

While ReFACT is a useful tool for updating text-to-image models, it has limitations. Our method is relatively slow, as it requires an optimization process, while the competing method, TIME, has a closed-form solution. ReFACT typically takes up to 2 minutes on a single NVIDIA A40 GPU.

Moreover, ReFACT sometimes fails to preserve unique facial features when editing specific facial attributes of a person. We experimented with editing different layers of the model (see Appendix A) and found that the issue persists. This phenomenon requires further investigation.

The technology presented in this paper is meant to improve human–technology interaction. Nevertheless, it may also be used with unintended consequences, such as planting harmful phrases or incorporating harmful social views. Given the vast research on harmful representations [47, 48, 49, 50, 51], we believe that sharing the editing method in this paper has more benefits than potential harms. We encourage future work to investigate the use of ReFACT for mitigating unwanted social impacts.

References

[1] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

[2] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[3] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[5] Hadas Orgad, Bahjat Kawar, and Yonatan Belinkov. Editing implicit assumptions in text-to-image diffusion models. *arXiv preprint arXiv:2303.08084*, 2023.

[6] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[7] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18187–18197, 2022. doi: 10.1109/CVPR52688.2022.01767.

[8] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022.

[9] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[10] Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. *arXiv preprint arXiv:2211.12446*, 2022.

[11] Chen Henry Wu and Fernando De la Torre. Unifying diffusion models' latent space, with applications to cyclediffusion and guidance. *arXiv preprint arXiv:2210.05559*, 2022.

[12] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. *arXiv preprint arXiv:2212.04489*, 2022.

[13] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

[14] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. *ArXiv*, abs/2103.10951, 2021.

[15] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*, 2022.

[16] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. 2022.

[17] Harsh Agrawal, Eli A. Meirom, Yuval Atzmon, Shie Mannor, and Gal Chechik. Known unknowns: Learning novel concepts using reasoning-by-elimination. In *Conference on Uncertainty in Artificial Intelligence*, 2021.

[18] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *ArXiv*, abs/2208.12242, 2022.

[19] Yoad Tewel, Rinon Gal, Gal Chechik, and Yuval Atzmon. Key-locked rank one editing for text-to-image personalization. *ArXiv*, abs/2305.01644, 2023.

[20] Niv Cohen, Rinon Gal, Eli A. Meirom, Gal Chechik, and Yuval Atzmon. "this is my unicorn, fluffy": Personalizing frozen vision-language representations. *ArXiv*, abs/2204.01694, 2022.

[21] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *ArXiv*, abs/2208.01618, 2022.

[22] Giannis Daras and Alexandros G. Dimakis. Multiresolution textual inversion. *ArXiv*, abs/2211.17115, 2022.

[23] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 351–369. Springer, 2020. doi: 10.1007/978-3-030-58452-8_21. URL https://doi.org/10.1007/978-3-030-58452-8_21.

[24] Amin Heyrani Nobari, Muhammad Fathy Rashad, and Faez Ahmed. Creativegan: Editing generative adversarial networks for creative design synthesis. *ArXiv*, abs/2103.06242, 2021.

[25] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Rewriting geometric rules of a gan. *ACM Transactions on Graphics (TOG)*, 41:1 – 16, 2022.

[26] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. Editing a classifier by rewriting its prediction rules. *ArXiv*, abs/2112.01008, 2021.

[27] Kevin Meng, Arnab Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *ArXiv*, abs/2210.07229, 2022.

[28] Vikas Raunak and Arul Menezes. Rank-one editing of encoder-decoder models. In *NeurIPS 2022 Workshop on Interactive Learning for Natural Language Processing*, November 2022. URL https://www.microsoft.com/en-us/research/publication/rank-one-editing-of-encoder-decoder-models/.

[29] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. *ArXiv*, abs/2110.11309, 2021.

[30] Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix X. Yu, and Sanjiv Kumar. Modifying memories in transformer models. *CoRR*, abs/2012.00363, 2020. URL https://arxiv.org/abs/2012.00363.

[31] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6491–6506. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.522. URL https://doi.org/10.18653/v1/2021.emnlp-main.522.

[32] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Knowledge neurons in pretrained transformers. *ArXiv*, abs/2104.08696, 2021.

[33] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/ec795aeadae0b7d230fa35cbaf04c041-Abstract-Conference.html.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[35] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[36] Teuvo Kohonen. Correlation matrix memories. *IEEE transactions on computers*, 100(4): 353–359, 1972.

[37] James A Anderson. A simple neural network generating an interactive memory. *Mathematical biosciences*, 14(3-4):197–220, 1972.

[38] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[39] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022.

[40] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL `https://openreview.net/forum?id=M3Y74vmsMcY`.

[41] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[42] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, 2021.

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *13th European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[44] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022.

[45] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[46] Paul Ekman. Are there basic emotions? 1992.

[47] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016.

[48] Federico Bianchi, Pratyusha Kalluri, Esin Durmus, Faisal Ladhak, Myra Cheng, Debora Nozza, Tatsunori Hashimoto, Dan Jurafsky, James Zou, and Aylin Caliskan. Easily accessible text-to-image generation amplifies demographic stereotypes at large scale. *arXiv preprint arXiv:2211.03759*, 2022.

[49] Jaemin Cho, Abhay Zala, and Mohit Bansal. Dall-eval: Probing the reasoning skills and social biases of text-to-image generative transformers. *arXiv preprint arXiv:2202.04053*, 2022.

[50] Kathleen C Fraser, Svetlana Kiritchenko, and Isar Nejadgholi. A friendly face: Do text-to-image systems rely on stereotypes when the input is under-specified? In *The AAAI-23 Workshop on Creative AI Across Modalities*, 2023.

[51] Lukas Struppek, Dominik Hintersdorf, and Kristian Kersting. The biased artist: Exploiting cultural biases via homoglyphs in text-guided image generation models. *arXiv preprint arXiv:2209.08891*, 2022.

[52] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[53] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

[54] Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh Jha, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. Torchmetrics - measuring reproducibility in pytorch. *Journal of Open Source Software*, 7(70):4101, 2022. doi: 10.21105/joss.04101. URL `https://doi.org/10.21105/joss.04101`.

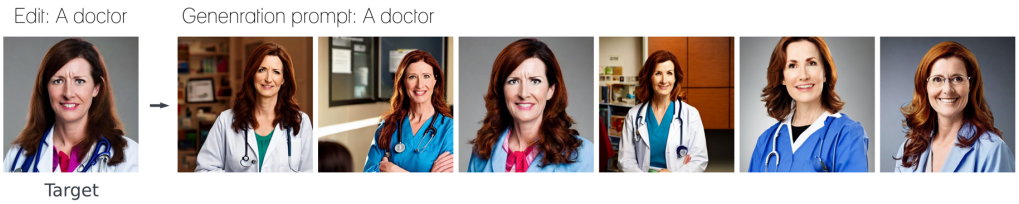


Figure 12: Editing "A doctoc" to "A female doctor" using a image as the target (t^*). Generated images shows that not only the gender was changes, and all photos showcase similar haircut, hair color, skin color, and pose.

Figure 13: Editing from an image versus editing from text. Editing from image allows us to set richer visual traits to be edited.

A Ablations of ReFACT

The modality of t^*. As previously discussed, t^* can either be an image representing the concept that we wish to edit to (a photo of Joe Biden), or a textual prompt representing it (the prompt "Joe Biden"). The image representation enables us to target multiple concepts at once, specifically applicable to changing the appearance of an object or role in a way that is difficult to explain via text. For example, if we want to edit the appearance of a TV character, that is now adapted to be played by a new actor, choosing t^* to be the name of the actor does not capture specific recognizable traits of the new adaptation - see Figure 13. However, the choice of specific image for editing might heavily affect the observed results. It is more difficult to specify the exact property we wish to edit (e.g., editing a doctor to a female doctor) without also affecting over attributes as well (the pose of the doctor, their hair or skin color) - see Figure 12. Expressing the target concept in text enables us to express our edit in a more general way, which is more robust. We found that editing to representations from the text encoder generalizes better, and is more robust compared to editing from the image encoder in terms of image diversity and editing quality. In case of editing appearance of roles, when the diffusion model encodes the edited character well, such as "Joe Biden", editing with text is more effective - see Figure 14.

Direct versus contrastive optimization. The computation of v^* described in Section 3.2 is done using a contrastive objective, maximizing the similarity between the editing prompt (e.g., "The president of the United States") and the target (e.g., "Joe Biden"), while *relatively* minimizing the similarity to other negative examples (e.g., "Donald Trump"). A different approach would be to directly maximize the similarity, without utilizing negative examples. To obtain v^* using direct optimization, we minimize the following loss:

$$v^* = \underset{v}{\operatorname{argmin}}\, d(E(t^*), E_{m^{(l)}:=v}(x_1)) \tag{3}$$

Preliminary experiments showed that contrastive optimization is more effective, and thus we continued with it.

Cosine similarity versus L2 distance. While cosine similarity better reflects CLIP's original training objective, L_2 is more directly related to our goal of editing the embeddings of the input

Figure 14: Editing from text is often more effective, when the CLIP model has a good representation for the target prompt.

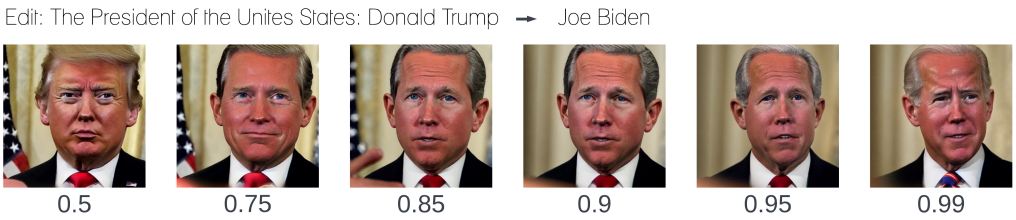


Figure 15: The importance of selecting a high threshold when optimizing v^*. Higher thresholds result in an image that is closer to our target edit.

prompt. We found L_2 to perform better in all experiments and thus present the results with L_2 as the distance function of choice.

Hyper-parameter search. We line searched over the following parameters, beginning from a basic variation which we found reasonable in early experiments and refining it on each search. First, we chose the layer to edit within the CLIP text encoder: Table 2 presents our layer search on the base configuration, for each dataset. We chose layer 9 for editing on TIME dataset, and layer 7 for editing RoAD. Then, we also searched for the learning rate for learning v^* (0.05); the maximum number of steps for optimization (100); and the probability threshold used for early stopping of v^* optimization process (0.99, illustrated in Figure 15);

Table 2: Editing in different layers of the CLIP model.

	TIME dataset				RoAD			
Edit layer	Efficacy	General.	Spec.	F1	Efficacy	General.	Spec.	F1
0	0.925	0.683	0.884	0.777	1.000	0.858	0.935	0.896
1	0.910	0.718	0.807	0.761	1.000	0.890	0.920	0.905
2	0.920	0.755	0.870	0.810	1.000	0.838	0.943	0.889
3	0.955	0.730	0.853	0.789	1.000	0.882	0.931	0.906
4	0.915	0.684	0.876	0.774	1.000	0.838	0.942	0.888
5	0.930	0.708	0.892	0.795	1.000	0.832	0.927	0.878
6	0.930	0.694	0.884	0.783	1.000	0.914	0.900	0.907
7	0.940	0.717	0.870	0.790	1.000	0.970	0.940	**0.955**
8	0.940	0.807	0.803	0.805	1.000	0.941	0.906	0.923
9	0.945	0.771	0.866	**0.817**	1.000	0.919	0.952	0.935
10	0.990	0.801	0.832	0.816	0.996	0.906	0.962	0.934

B RoAD

RoAD consists of two types of editing requests: Roles and appearances. Roles refer to positions filled by individuals, such as politicians, musicians, and pop-culture characters (e.g., “The President of the United States”, “Ross Geller”, “Forrest Gump”). Appearances are editing request that aim to alter the complete visual appearance of an object (e.g., “Apple”, ”Honda Accord”). Although all entries in RoAD share the same structure, there are some conceptual differences in between editing roles and editing appearances. For example, when editing “The President of the United States” to “Joe Biden”, we expect the model to still be able to generate the source prompt, “Donald Trump”. This is not the case when editing “Apple” to “Avocado”, since both the editing prompt and the source prompt are “Apple” are expected to demonstrate the edited fact.

RoAD is split into a test set (90 entries) and a smaller, disjoint, validation set (10 entries), used for hyper-parameter search. Each entry in RoAD consists of an editing prompt, a source, and a target. The editing prompt (e.g., "The Prince of Wales", "A computer") describes a role or entity whose visual appearance can be consistently generated by a text-to-image model. In entries editing roles (46 entries), the source describes the person generated by the model when given the editing prompt (e.g., "Price Charles"). For entries editing appearance (64 entries), the source describe the entity itself and is the same as the editing prompt (e.g., "A computer"). The source and target of each entry can be used to generate multi-modal input to fit various editing algorithms. They can be used simply as textual source and target descriptions, or be used to automatically generate images using a text-to-image model of choice, which are later fed to the editing algorithm.

For each positive prompt, RoAD includes the prompt itself (e.g., "The Prince of Wales in the park"), and two variations of the positive prompt describing the source and targets (e.g., "Prince Charles in the park", "Prince William in the park", respectively). For RoAD entries editing appearance, the positive prompt and source-positive prompts are again identical. For each negative example RoAD includes a negative prompt (e.g., "Prince Harry", "A computer screen") and The negative-target prompt (e.g., "Prince William", "A laptop screen").

C Implementation details

We implemented our code using Pytorch and Huggingface libraries [52, 38, 53], and based our rank-one editing code on the code of Meng et al. [6]. All experiments are averaged over 25 seeds from 0 to 24. We ran the experiments on the following GPUs: Nvidia A40, RTX 6000 Ada Generation, RTX A4000 and GeForce RTX 2080 Ti.

D Metrics

We describe here the measured metrics in a mathematical notation.

Generalization:

$$\frac{\#[\text{CLIP}(\text{target\_prompt}) > \text{CLIP}(\text{source\_prompt})]}{\#\text{positive\_examples}}$$

Specificity:

$$\frac{\#[\text{CLIP}(\text{source\_prompt}) > \text{CLIP}(\text{target\_prompt})]}{\#\text{negative\_examples}}$$

We computed the efficacy, specificity and generality metrics using Laion’s ViT-G/14 [40], which is the best open source CLIP model to date. The general CLIP score used to evaluate generation quality was computed using the standard Torchmetrics [54] CLIPScore class, for which CLIP-vit-large-patch14-336 is the best available CLIP model.

E Additional qualitative results

We present additional qualitative results of ReFACT. Figure 16 demonstrates the generated images for the prompt “a cake” across different edits, using the same seeds. Figure 17 illustrates the generality of ReFACT and Figure 18 illustrates its specificity.

Figure 16: Editing "A cake" to different flavors.

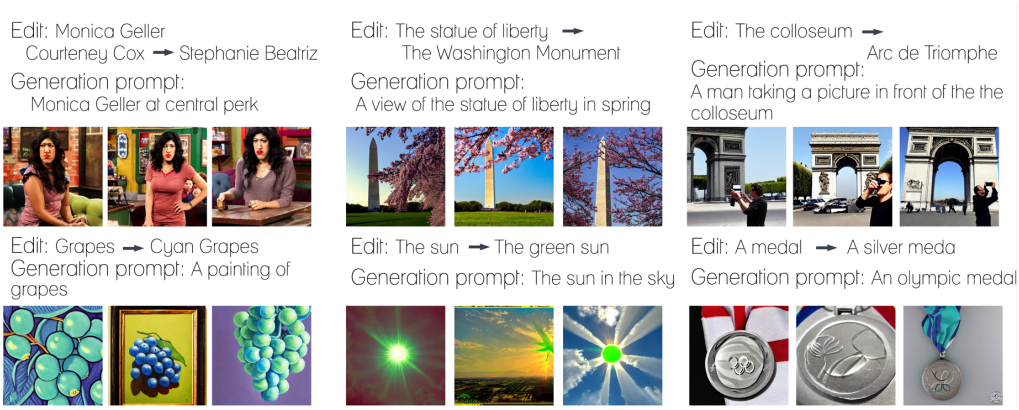


Figure 17: Generality of ReFACT.

F Limitation of ReFACT: facial features

As we discussed in Section 5.3, an edit considering a person can sometimes modify facial features in an undesired way. We experimented in editing different layers of the model to overcome this limitation, but found that it only helps slightly or not at all. This is demonstrated in Figure 19.

G Modifications to TIME

TIME [5] is a method designed to edit implicit assumptions, and as such, it is designed to edit from an under-specified prompt ("a pack of roses") to a specified prompt ("a pack of **blue** roses"). As we discussed in Appendix B, our dataset RoAD contains two types of samples: roles and appearance. We separate their treatment when we run TIME:

Roles. Roles are more similar to the edits preformed by TIME, and can be written as an under-specified prompt ("The President of the United States") and a specified prompt ("Joe Biden as the President of the United States"). We use this formulation to apply TIME to these samples.

Appearance. Appearances entries are different from those used by TIME, since they edit from one object to an entirely different one. For instance, editing "Apple" to "Avocado". We do not have a natural way of designing this edit as an under-specified prompt and a specified prompt. Thus, for these samples we only edit the pad tokens, which matches the formulation of TIME that edits only matching tokens and also edits the pad tokens.

Additionally, we make modifications to TIME that make it more similar to ReFACT, to narrow down the reason that ReFACT is more successful. We experiment with two approaches: editing only the [EOS] token and editing directly to the target prompt ("Joe Biden"), like we do in ReFACT. When we taking the former, we only edit the [EOS] token, as done in ReFACT. We show in Table 3 the results on RoAD with the various modifications. We choose the original setting, that achieves the

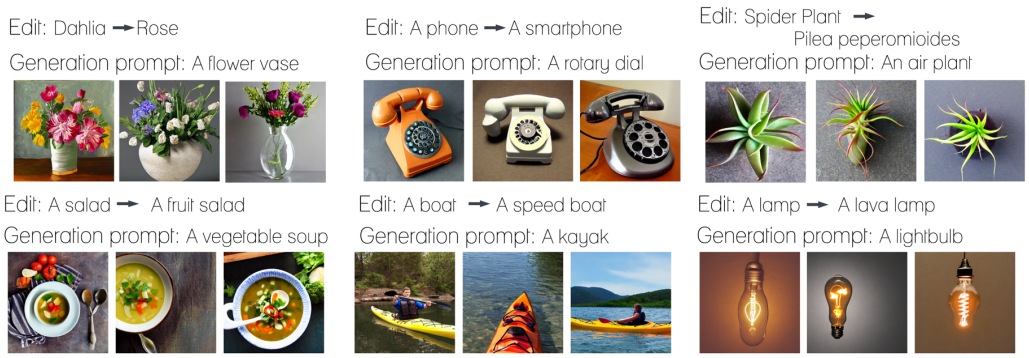


Figure 18: Specificity of ReFACT.

Figure 19: Editing sometimes result in facial features change, even when editing different layers.

highest F1 score. All of the results are relatively poor, which indicates that the difference between the method lies within the component of editing (attention layers versus inner MLP layers).

H Per-layer analysis: facial expressions

H.1 Implementation

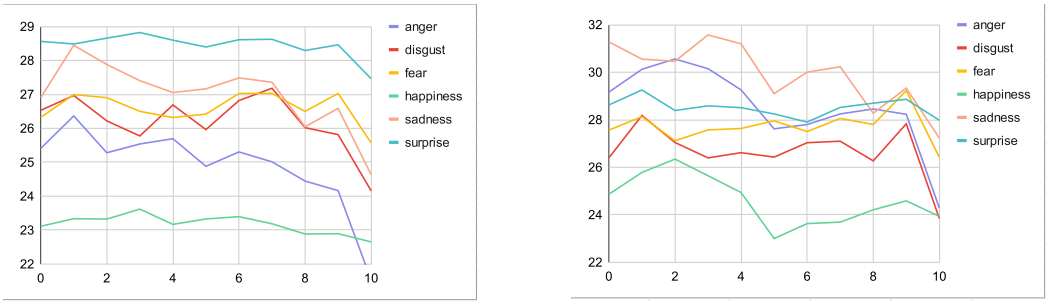
For this experiment, we needed prompts that generate portrait images of people. We found that prompts such as "a portrait of a man" or "a photo of a woman" tend to generate images of very different styles, while the prompt "a doctor", which we borrowed from TIME dataset, tends to generate realistic images of people looking directly at the camera. We thus use it to perform our experiments on facial expressions. Since the generative model is biased [5], it tends to generate male images of doctors and thus we use the prompts "a male doctor" and "a female doctor".

H.2 Additional results

In Figure 10, we present the plots from the image editing and the text editing experiments, on different emotions and layers. The two plots follow the same trend, illustrating that editing in lower layers results in the facial expression more apparent in the generated image by the edited model. Moreover, Figure 20 and 22 present more illustrations of this phenomenon.

Table 3: Modifications to TIME algorithm, tested on RoAD validation set.

Edit to target prompt	Edit [EOS]	Generality	Specificity	F1
False	False	0.42	0.79	0.58
True	True	0.31	0.94	0.54
False	True	0.17	0.96	0.41



(a) Editing with image embedding.

(b) Editing with textual embedding.

Figure 20: CLIP score of different emotions on the generated images after editing each later.

Figure 21: Editing with an image target, across layers.

I Multiple edits

We evaluate multiple edits by preforming the editing requests sequentially on the same CLIP text encoder, using the same hyper-parameters as ReFACT. We edit entries from both the TIME dataset and RoAD, testing three different permutations of the edit requests. We edit up to 90 facts. Figure 23 shows the efficacy, generalization and specificity of the model at every 10 edits interval. Our experiments show that multiple edit result in only a slight drop across all metrics, which can be a result of the high specificity demonstrated by ReFACT.

Figure 24 shows examples of entries that were edited in the first ten sequential edits, along the different steps. The first two rows demonstrate editing "The British Monarch" from "Queen Elizabeth" to "Prince Charles", and editing "Daffodils" to "Blue Daffodils". The figure shows minimal changes in the generated images for these edits after multiple sequential edits. On the other hand, editing "Carnation" to "Foxgloves" shows a drop in efficacy after 20 edits, as the model generated images of different flowers.

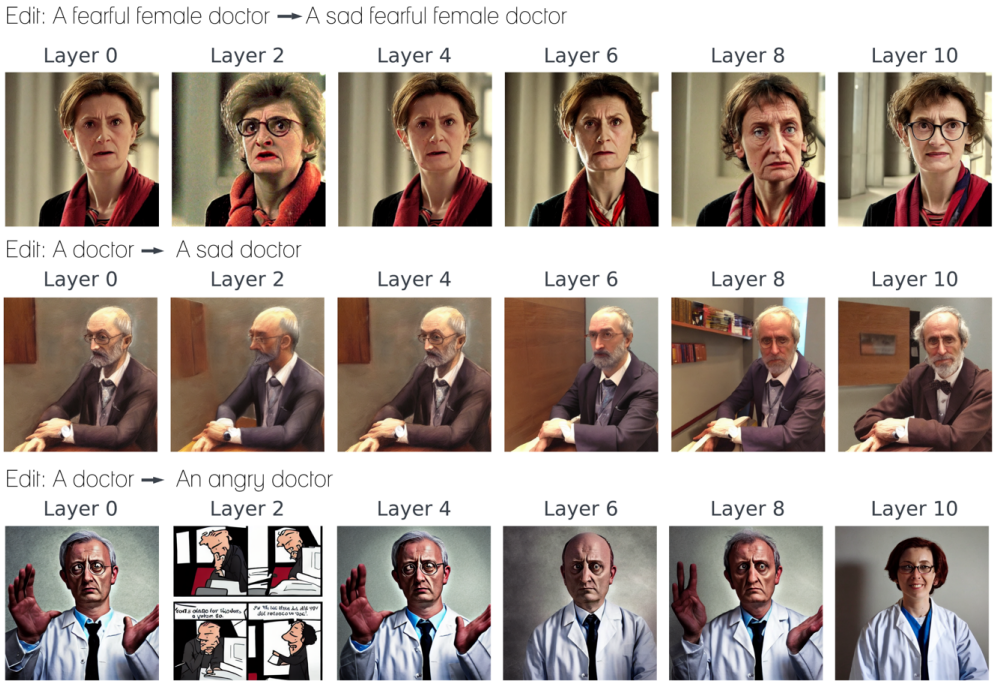


Figure 22: Editing with a textual target, across layers.

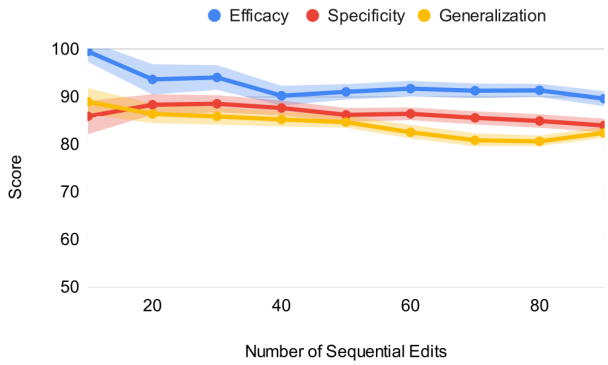


Figure 23: Efficacy, generalization and specificity after multiple sequential edits.

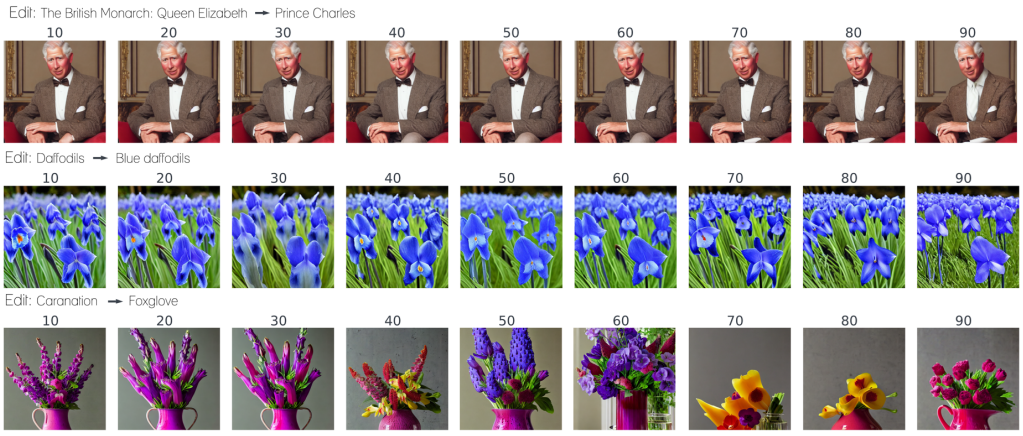


Figure 24: Examples of edited knowledge preservation when preforming multiple sequential edits. Top two rows show examples of edits that are left unaffected by later edits. Bottom row shows and example of an affected edit.