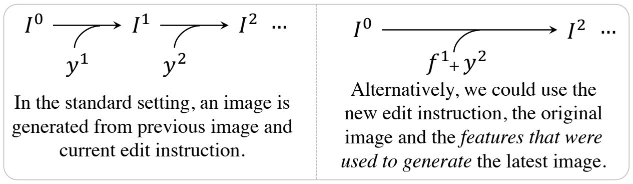
A. Alternate Iterative Editing Approaches

We would like to briefly discuss some alternate approaches that we had explored towards addressing iterative editing. Though intuitive, these approaches was found to be less effective than our proposed *latent iteration* approach.

A.1. Change Isolation and Feature Injection



In the standard setting, an image is generated from previous image and current edit instruction.

Alternatively, we could use the new edit instruction, the original image and the *features that were used to generate* the latest image.

Figure 10. Ideally, it would be effective if we could isolate the changes intended by each edit instruction to the original image $\boldsymbol{I}_0$, and compose the final image by adding them successively to $\boldsymbol{I}_0$. We propose to do this in the feature space.

Iterative editing involves making semantic changes corresponding to a set of edit instructions $\{\boldsymbol{y}_0, \cdots, \boldsymbol{y}_n\}$ to the original image $\boldsymbol{I}_0$. If we could isolate the changes each $\boldsymbol{y}_i$ cause to $\boldsymbol{I}_{i-1}$, we can cumulate these changes and apply them directly to $\boldsymbol{I}_0$. This would completely side-step the noisy artifact addition issue that surfaces when an image is recursively passed though the model for editing.

We explore an approach that does the change isolation in the image space and applies it back to the image in its feature space. Let $\boldsymbol{I}_i$ be the image generated following the edit instruction $\boldsymbol{y}_i$. We identify the changes caused by $\boldsymbol{y}_i$ by taking a difference $\boldsymbol{I}_i^{\triangle} = \boldsymbol{I}_i - \boldsymbol{I}_{i-1}$. Next, we isolate the features $\boldsymbol{f}_i^{\triangle}$ corresponding to $\boldsymbol{I}_i^{\triangle}$ (by forwarding passing $\boldsymbol{I}_i^{\triangle}$ through the model), and inject them into the model while consuming $\boldsymbol{y}_{i+1}$ and $\boldsymbol{I}_0$. As we want to preserve the overall image statistics of the original image, we choose to inject the self-attention features into the decoder layers of the UNet [23], following Tumanyan *et al.* [28] and Ceylan *et al.* [3]. Though this approach helps to preserve the background of $\boldsymbol{I}_0$, the newer edits were not well represented in the images. The high-frequency elements like edges were well carried over, but finer details are mostly ignored.

A.2. Iterative Noise Removal

Another approach is to explicitly reduce noise that gets accumulated while we iteratively pass the edited image through the model for successive edits. We used Gaussian blur towards this effort. Though this approach indeed reduces the noise accumulation, it significantly blurs the subsequent edit images. Feature injection to the self-attention layers were able to partially reduce the effect of blurring, but the images were not better than doing *latent iteration*.

B. Latent Iteration vs. Image Iteration

A key finding of our analysis is quantifying the accumulation of noisy artifacts, while iteratively passing an image through the latent diffusion model. Here, in Fig. 12, we showcase a few more examples where we see degradation in the image quality while being iteratively processed. We choose photos, painting and landscape pictures for this study. We iteratively pass these images though the LDM (introduced in Sec. 4.2) for 20 steps. We use a null string as the edit instruction to neutralize its contribution. As is evident from the figure, when we iterate in the image space, we see more severe image degradation when compared to iterating in the latent space. In Fig. 16, we compare with a new semantic edit instruction in each step.

C. More Qualitative Comparisons

Figs. 13 to 15 showcases more qualitative evaluation of EMILIE. We compare with two top performing baselines: 1) concatenating all the edit instructions that we have seen so far and applying them to the original image, and 2) iteratively passing the edited image through Instruct Pix2Pix [2] to be updated by the newer edit instruction. We see that EMILIE is able to create images with lower noise levels and is more visually appealing and semantically consistent.

D. Failure Cases

EMILIE has shortcomings too. Fig. 11 shows one such setting where EMILIE fails to undo edit instructions. It is natural for the edit instructions to be conflicting to each other. For example, we might add a Christmas tree at step 1, and later plan to remove it. This is hard for the method to do. Also, in some cases, the model makes inconsistent changes. For instance, in Fig. 15, the mustache that was added in step 2 should have had ideally been gray. Though the edit instruction was to change the existing bird to the crow, the model indeed added a new bird too. This is largely an artifact of the base editing framework [2] that we build on. These are interesting future research directions.

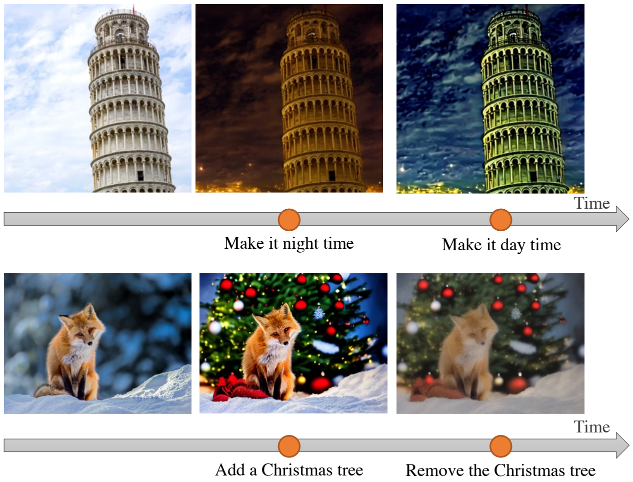


Figure 11. EMILIE is not able to do identity preservation across edits that reverse concepts and conflict each other.

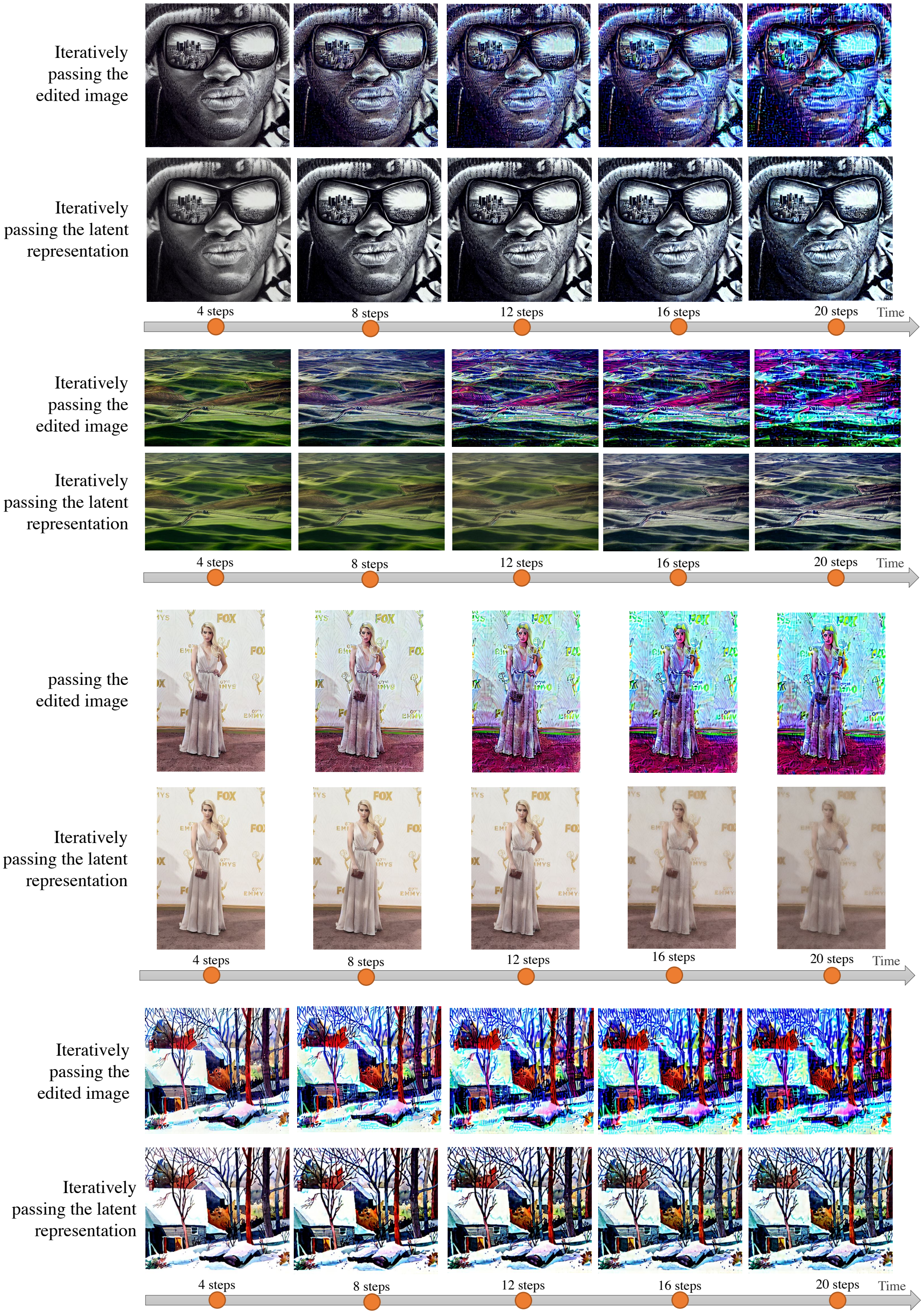

Iteratively passing the edited image
Iteratively passing the latent representation
4 steps
8 steps
12 steps
16 steps
20 steps
Time
Iteratively passing the edited image
Iteratively passing the latent representation
4 steps
8 steps
12 steps
16 steps
20 steps
Time
passing the edited image
Iteratively passing the latent representation
4 steps
8 steps
12 steps
16 steps
20 steps
Time
Iteratively passing the edited image
Iteratively passing the latent representation
4 steps
8 steps
12 steps
16 steps
20 steps
Time

Figure 12. While iteratively passing an image through the diffusion model, we see noisy artifacts being accumulated (first row in each pair). Iterating over the latent representations helps minimise the artifact accumulation (second row in each pair).

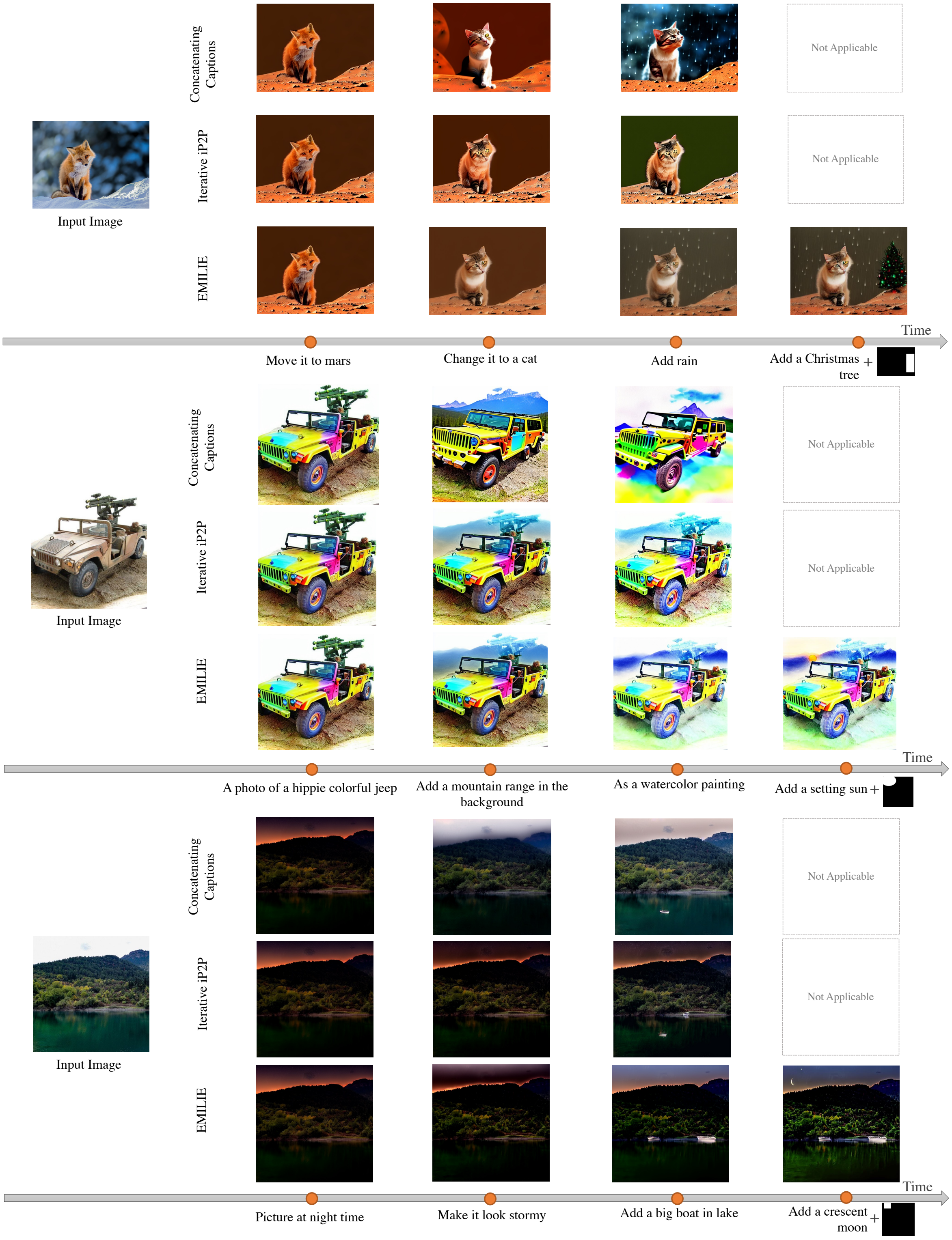


Figure 13. We showcase more qualitative comparisons with baselines. We see that EMILIE is able to produce more realistic and meaningful edits when compared to concatenating all the instructions seen so far, and iteratively using Instruct Pix2Pix [2].

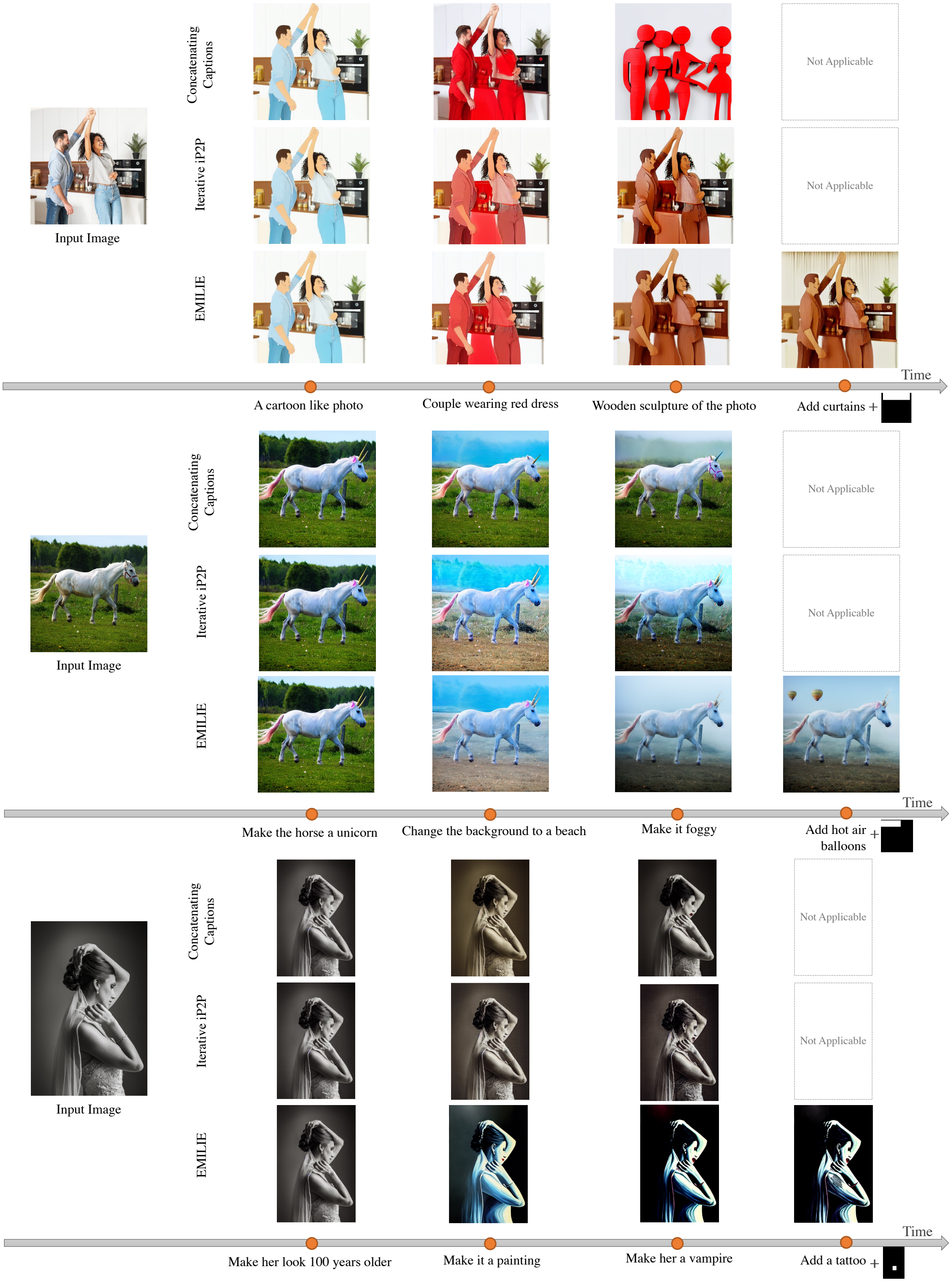
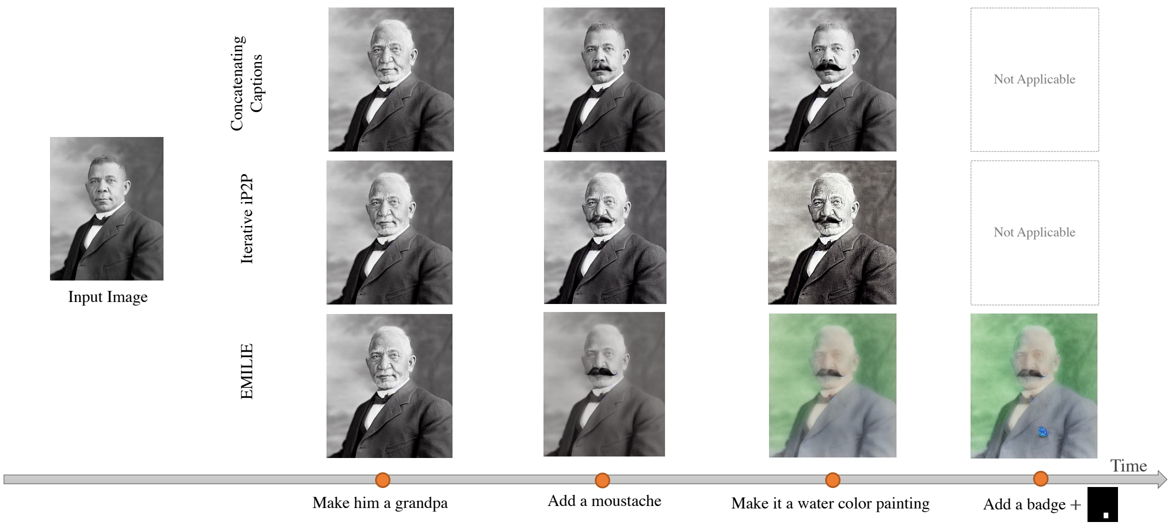


Figure 14. We showcase more qualitative comparisons with baselines. We see that EMILIE is able to produce more realistic and meaningful edits when compared to concatenating all the instructions seen so far, and iteratively using Instruct Pix2Pix [2].



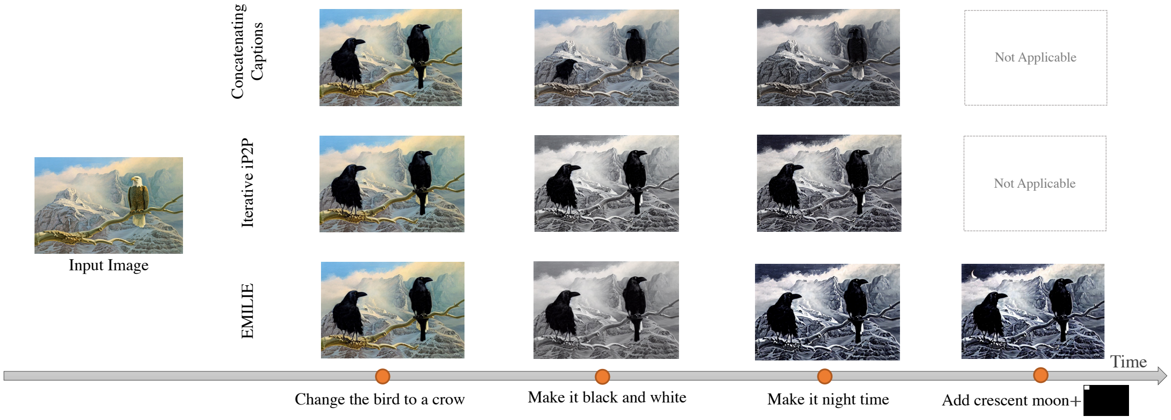


Figure 15. We showcase more qualitative comparisons with baselines. We see that EMILIE is able to produce more realistic and meaningful edits when compared to concatenating all the instructions seen so far, and iteratively using Instruct Pix2Pix [2].

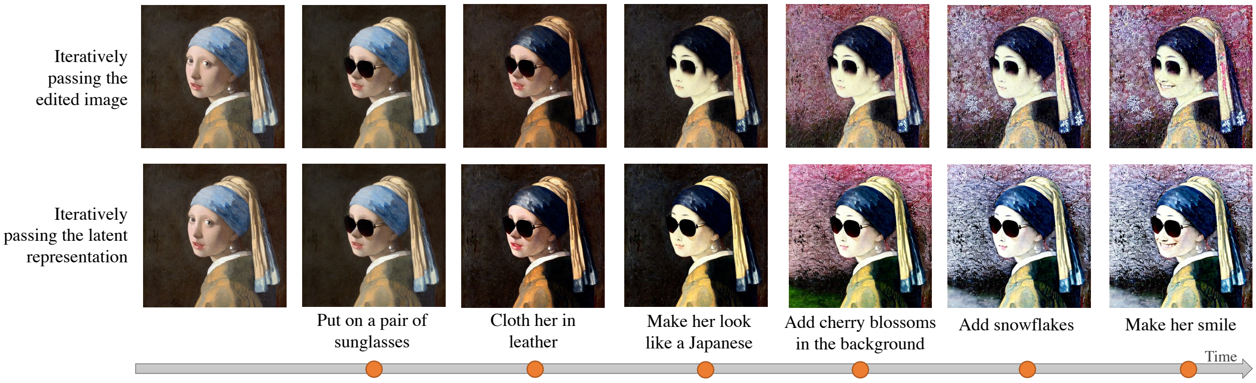

Iteratively passing the edited image
Iteratively passing the latent representation
Put on a pair of sunglasses
Cloth her in leather
Make her look like a Japanese
Add cherry blossoms in the background
Add snowflakes
Make her smile
Time

Figure 16. While iteratively editing images with a new edit instruction in each step, we observe that EMILIE is able to add new semantic information with minimal artifact accumulation.