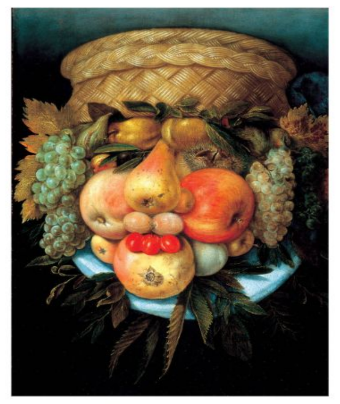
Making Images from Images:

Tightly Constrained Parallel Denoising

Shumeet Baluja, Google Deepmind
David Marwood, Google DeepMind
Ashwin Baluja, Northwestern University

Classic Optical Illusions: A Fascination for Centuries

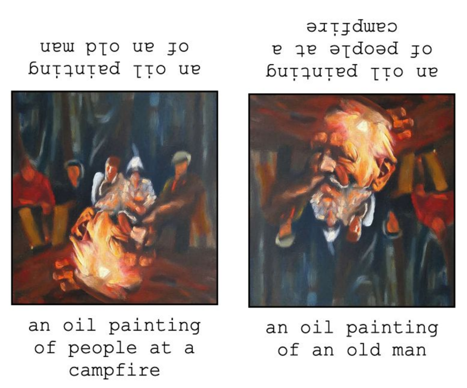
- G. Arcimboldo's "Fruit Basket"(1590) that shows a face when upright, and a fruit basket when upside-down.
- Duck or Rabbit? 1892 issue of *Fliegende Blätter*

A.


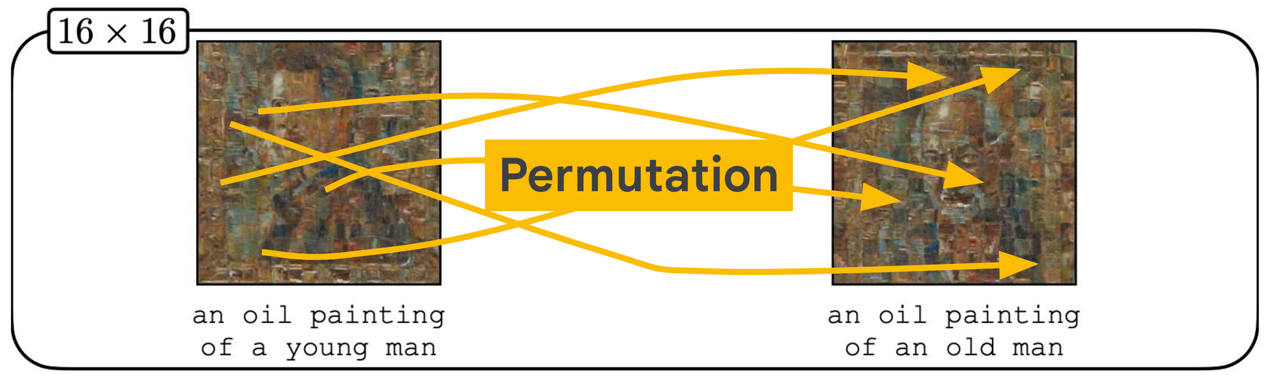
B.

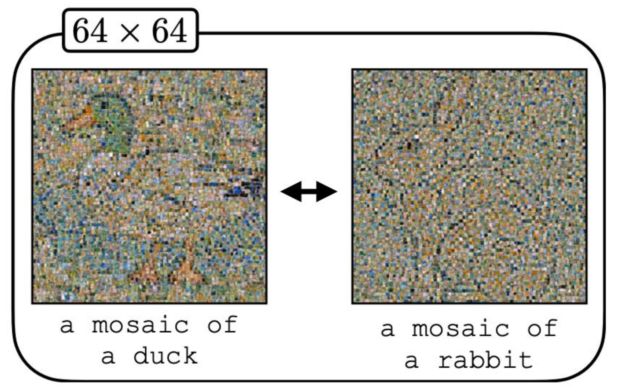

Computational Optical Illusions: In 2023 & 2024 - Photo Realism

- Used the "standard" text-to-image diffusion process to handle multiple objectives (e.g. multiple prompts).
- Used *multiple* prompts ("people at a campfire" & "old man") to create a *single* image that through a pre-specified transformation, represented the multiple prompts.
- Geng et. al (2023) showed any rearrangement of pixels is a suitable transformation for simultaneous denoising processes.



Examples Images taken from:
Geng, D., Park, I., Owens, A.: Visual anagrams: Generating multi-view optical illusions with diffusion models, 2023.



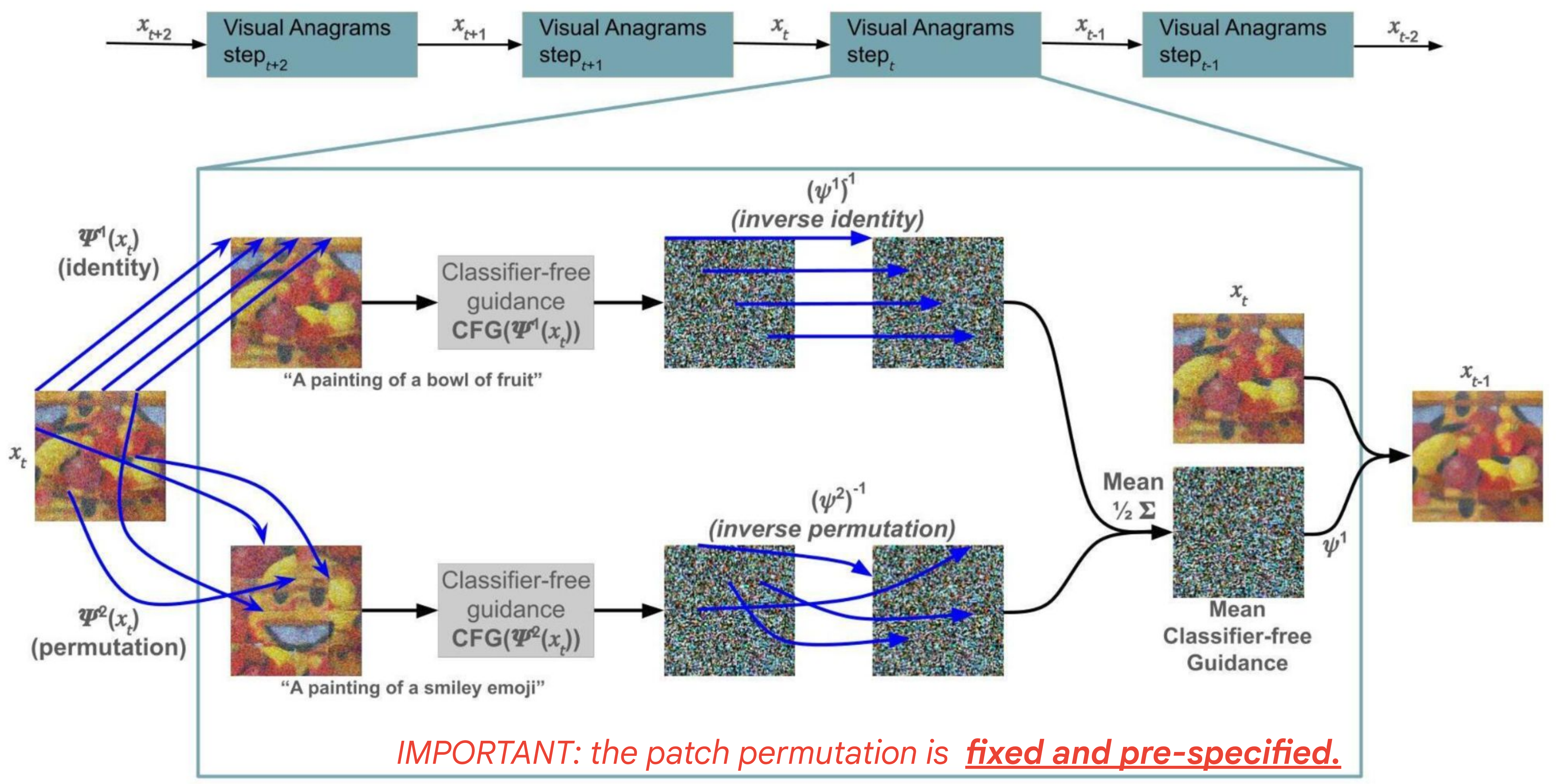


Also See:
1. Tancik, M.: Illusion-diffusion.
2. Geng, D., Park, I., Owens, A.: Visual anagrams: Generating multi-view optical illusions with diffusion models
3. Geng, D., Park, I., Owens, A.: Factorized diffusion: Perceptual illusions by noise decomposition
4. Chen, Z., Geng, D., Owens, A.: Images that sound: Composing images and sounds on a single canvas
5. Bar-Tal, O., Yariv, L., Lipman, Y., Dekel, T.: Multidiffusion: Fusing diffusion paths for controlled image generation

Synthesis of Images via Patch Permutations *"A Bowl of Fruit" ⟶ "A Smiley Emoji"*

Modify Standard Diffusion Process:

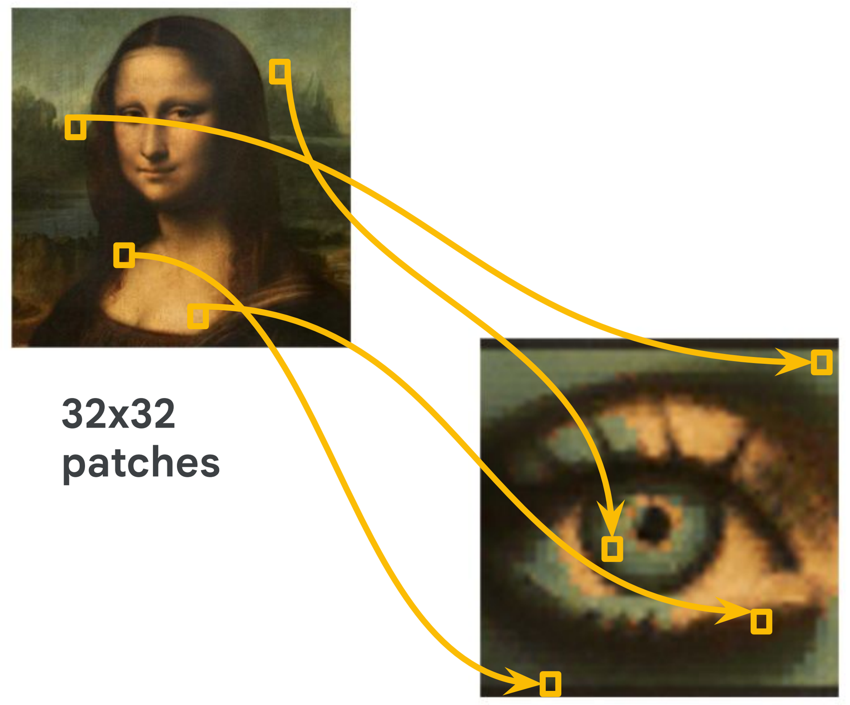
1. Apply the patch permutation to the noisy-image to obtain the two views of the same image.
2. Estimate the noise in each image according to the respective prompts.
3. Re-align the noise estimates by the inverse patch permutation.
4. Average the noise estimates.
5. De-noise the image with the mean-noise.



THE PROBLEM: HOW DO WE TRANSFORM A SPECIFIC PIECE OF ART?

Task:

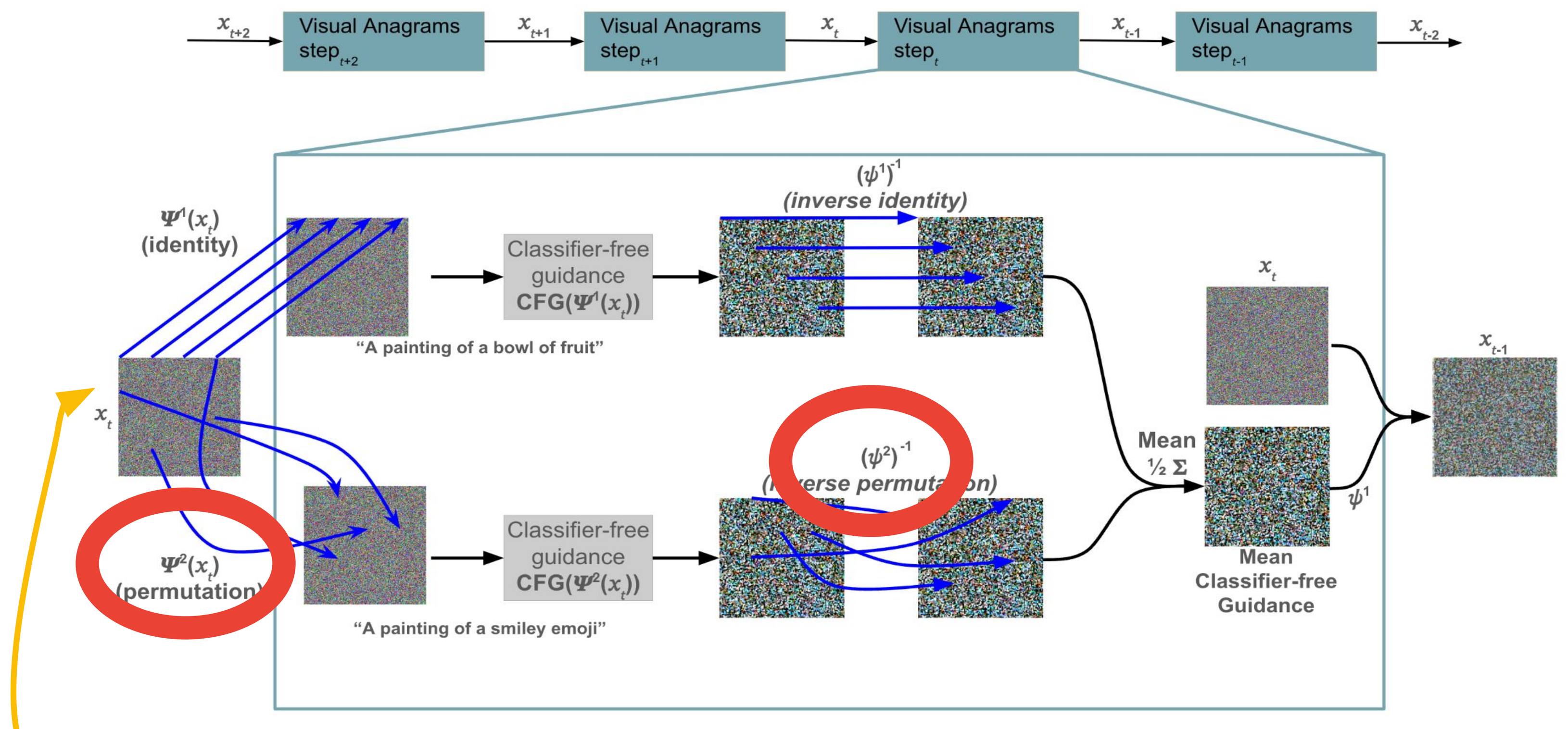
Create a close-up of an eye with image patches from the Mona Lisa?



- If we knew the final eye image we wanted, the problem is simple: finding the best match.
  - Compute **N**x**N** patch distances & use the ***Hungarian Method / Kuhn-Munkres*** for optimal matching.
- Difficulties:
  - The end-image is <u>not</u> known.
- This is a very tightly constrained problem.
  - **We no longer change pixel RGB values.**
  - We only move tiles around.

Any prespecified tile rearrangement from a known image would simply yield a fixed, nonsensical image.

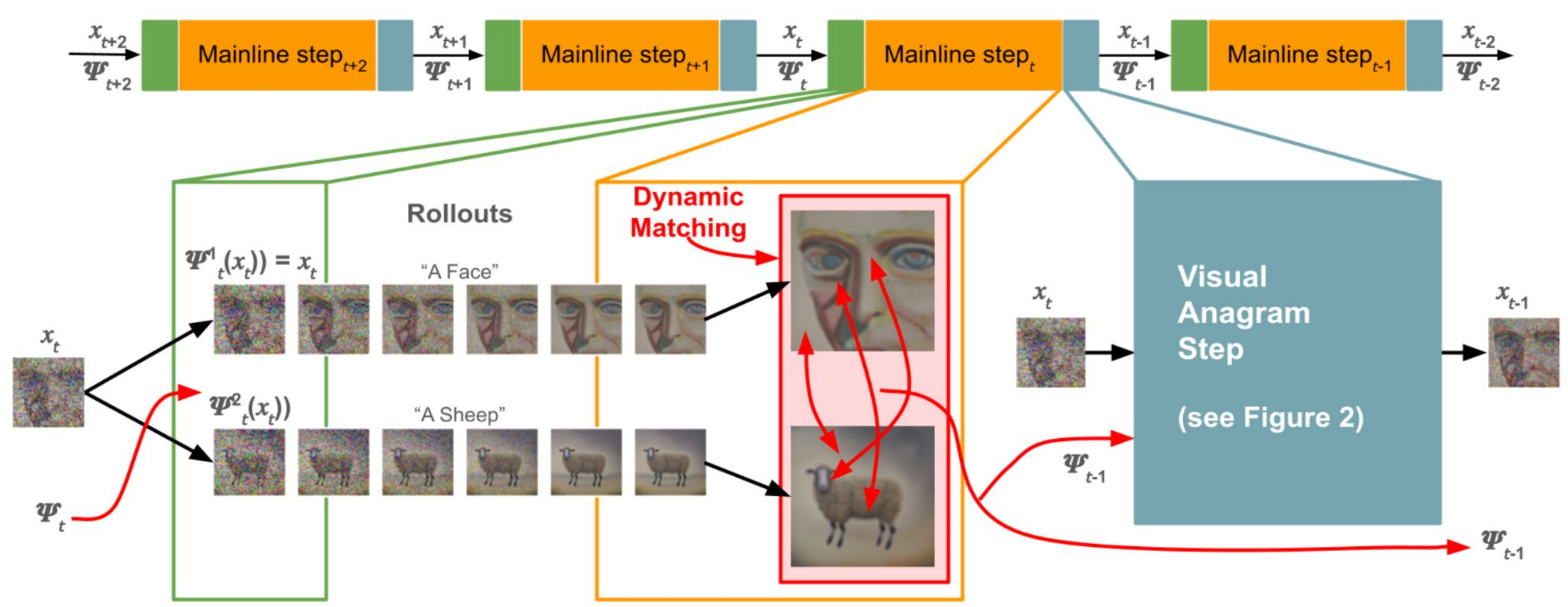
Therefore:
Tile arrangement is learned.



Problem: Hungarian Matching is matching noise!

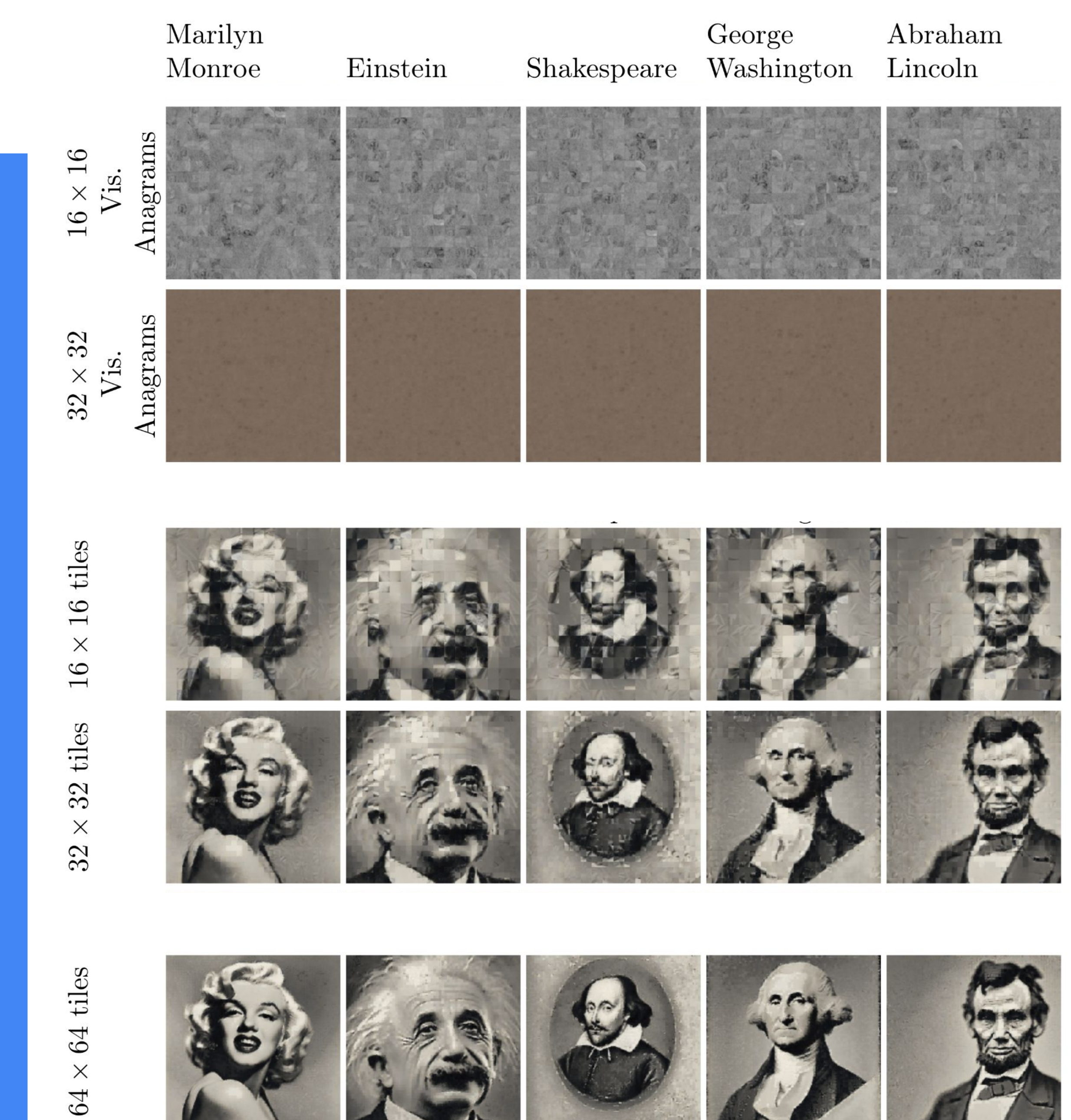
KEY INSIGHT: WE APPROXIMATE WHAT THE FINAL IMAGE WILL BE VIA ***<u>THE ROLLOUT</u>***

- **At every time-step t:**
  - Permute the tiles from the permutation found in the last step.
  - Simulate a diffusion process forward for both prompts. (a few steps or to the end).
  - Calculate the lowest-cost tile permutations (Hungarian Method) to match the tiles in the rollouts.
  - Use this **new permutation** as input to the procedure shown earlier.
- We are no longer trying to match noise, but an approximation (guess?) of what the end image will look like.



Fixed permutations
- As the number of tiles increases, the results become worse.
- The constraints make it difficult to find suitable images.

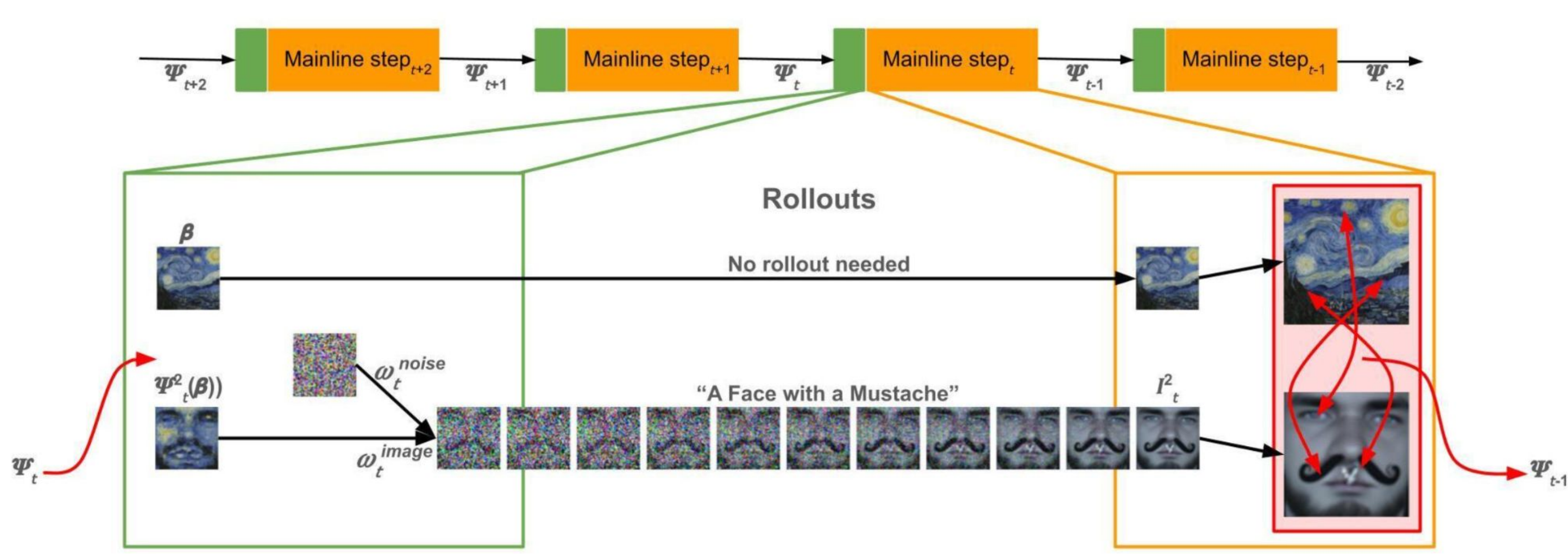
Dynamic permutations:
- The more tiles there are – the more degrees of freedom, and ***the better the results.***



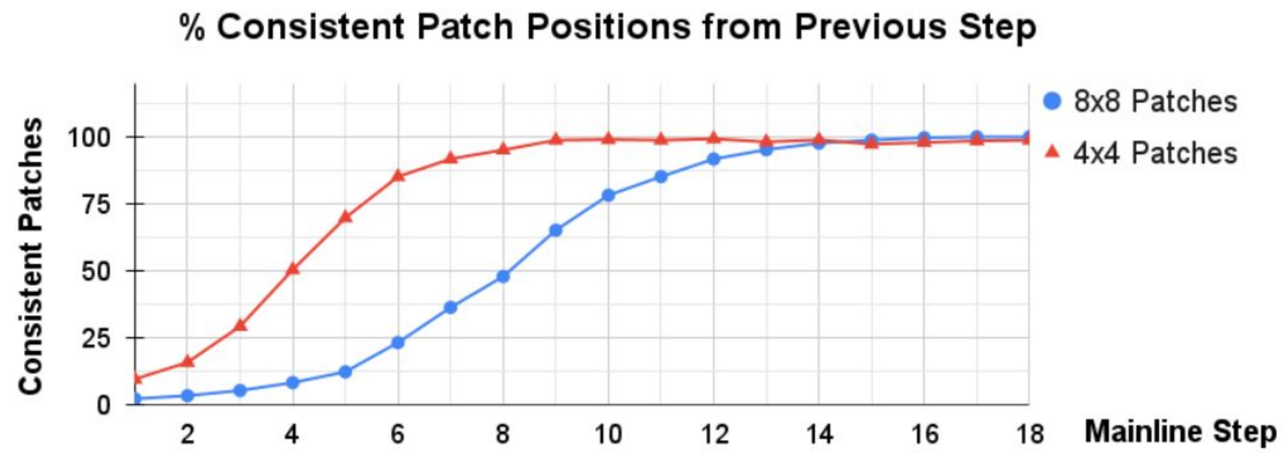
SOLUTION: WITH DYNAMIC PERMUTATIONS, SPECIFY THE BASIS ARTWORK, β

- **We only rollout the image that we are creating.**
  The basis image, β, does not need to be rolled-out.

- Permute β with the last determined permutation.
- We add noise to the permuted image (as right now the image is a noise free permutation of β).
  - The noise is added in the amount specified by the diffusion schedule.
- Perform **The Rollout.**
- A new optimal-patch-permutation is computed between the results of the Rollout and the basis image, β, using Hungarian Method.



PERMUTATION CHANGES OVER TIME

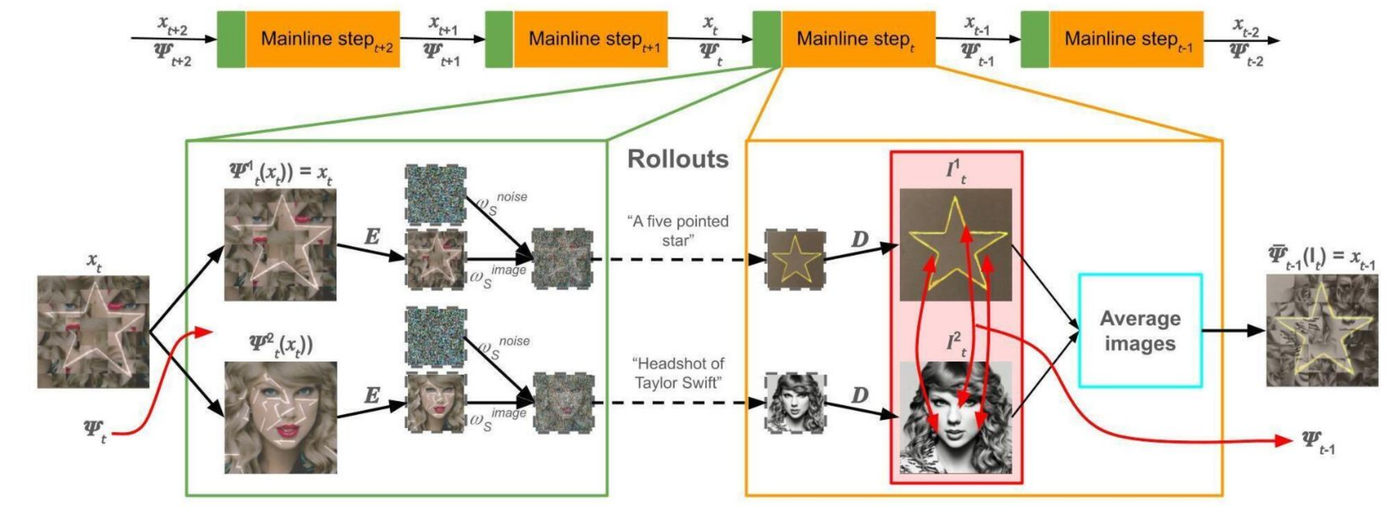


As the diffusion process continues, changes between successive permutations is reduced. As the images become more consistent (as $t \to 0$) each ψ_{t-1} looks more like ψ_t. By iteration 15, the 8×8 tiles have ceased movement. Movement ends by iteration 9 for the 4×4 tiles. Average of 10 runs shown.

MOVING TO LATENT DIFFUSION

```
Changes for Latent Diffusion

1. Transformation are matched
   in pixel space.
2. Diffusion occurs after
   encoding in Latent space.
3. We do a full rollout from
   step 50 to step 0.
4. The noise introduced is
   set to 98%, with 2% of the
   previous rollout image.
```

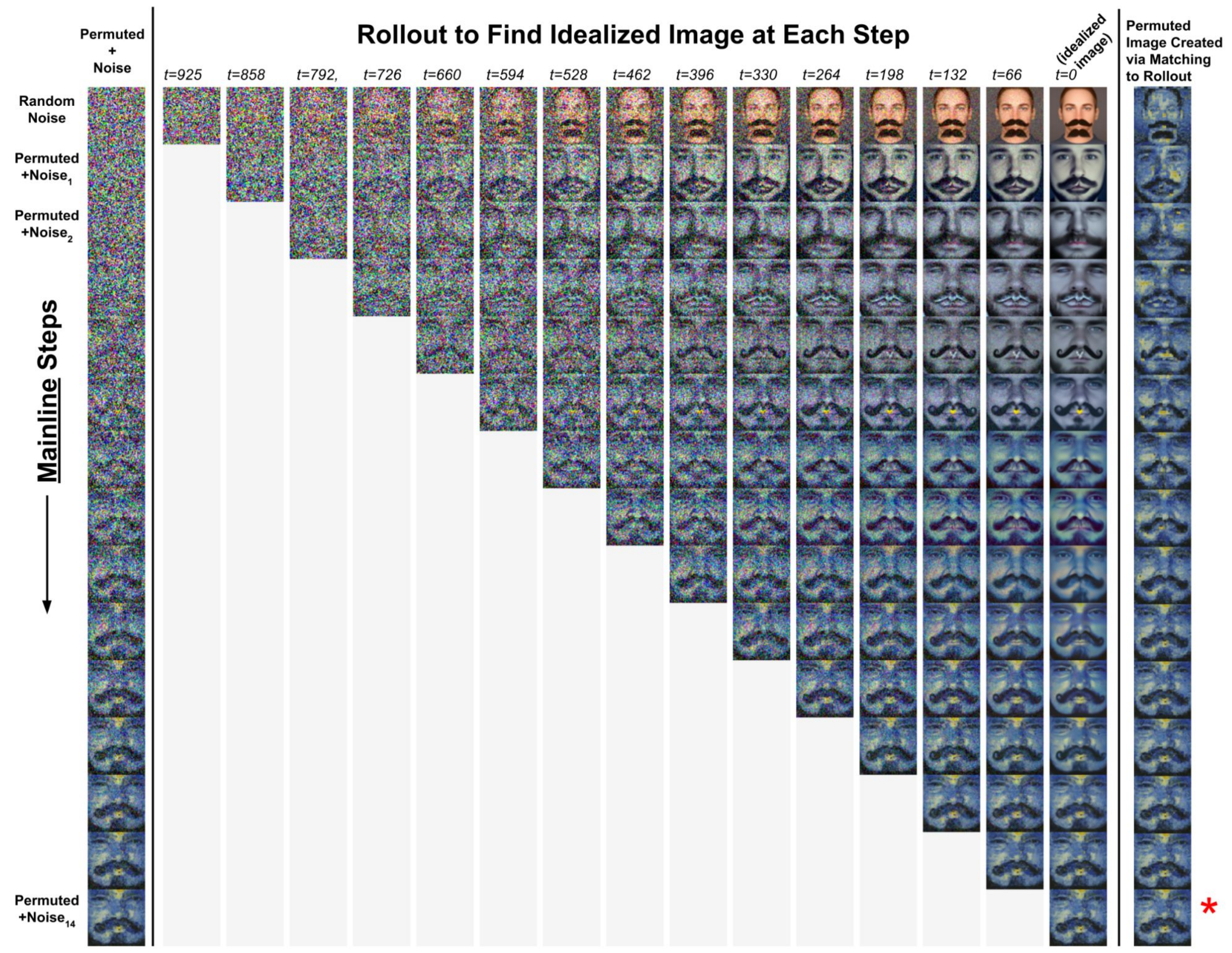


Changes for latent diffusion. Latent representations, with a dashed outline, are used for diffusion and pixel images are used for dynamic matching and calculating Ψ. Latents are decoded with D for dynamic matching then re-encoded with E; D&E are pretrained components of the latent diffusion system. Rollout mixing uses fixed w_S^{image} and w_S^{noise}. Rollouts span a full diffusion run from $S = 50$ to 0.

ROLLOUTS VISUALIZED

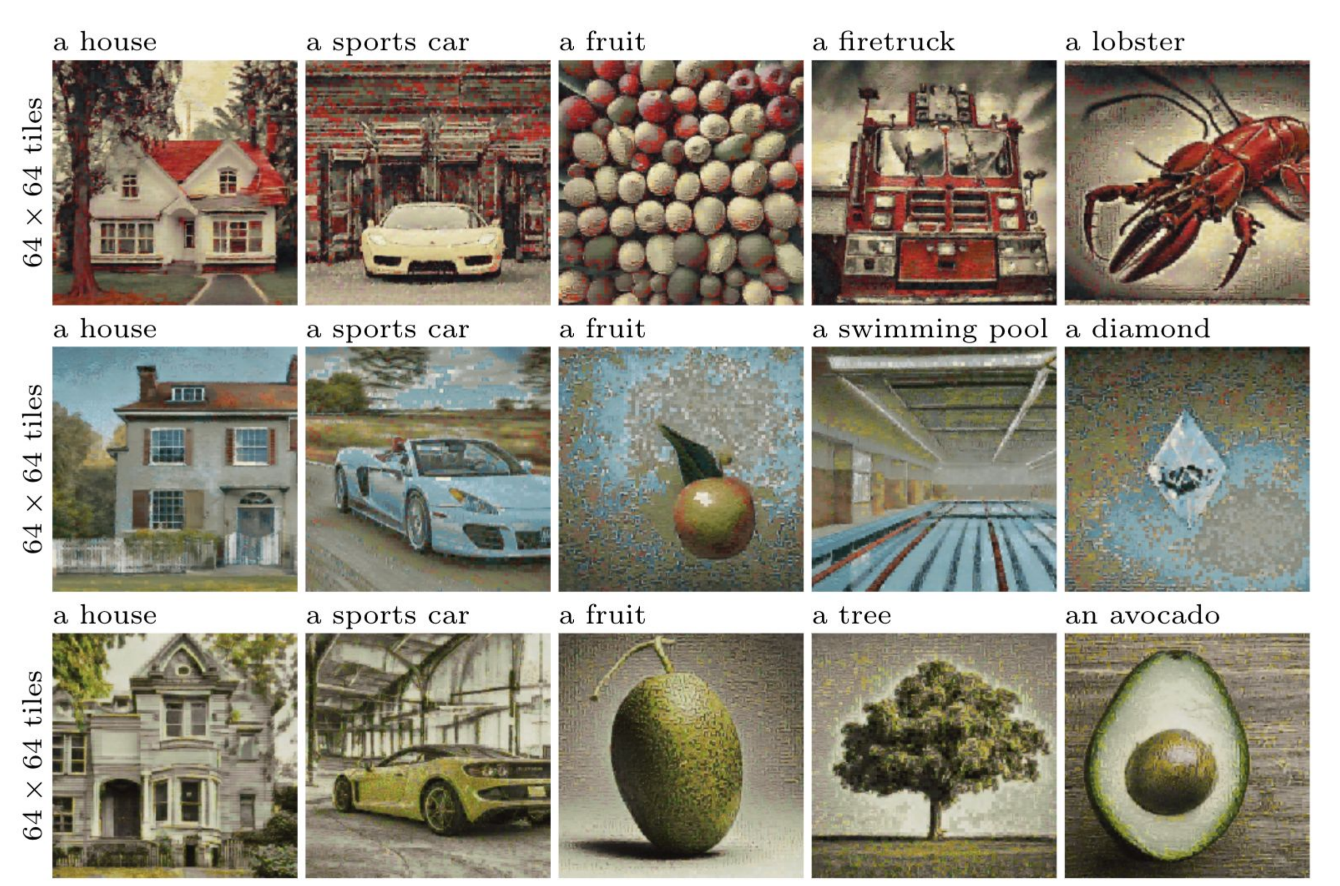
Basis Image, β

$\longrightarrow$ man with a mustache



CONSTRAINT PROPAGATION IN PARALLEL DIFFUSION

- This provides an intuition of how constraints are propagated through the parallel diffusion processes.
- In these three examples with 5 prompts, 3 prompts are kept constant, 2 are changed.
- The last 2 prompts have strong colors associated with them.
- Notice how the typical colors of the last 2 prompts are visible in all 5 images – an effect of constraint propagation.



Constraint propagation across all images when no-basis images are used.

RESULTS

The images created reflect what is representable with the patches in the original source image.