

A. Additional Results

A.1. Additional Ablations

We show quantitative results for requested additional ablations in Tab. A. Specifically, we investigate optimizing a variant of our training objective where the main change is omitting the attribute scale λ_i variation. This performs substantially worse than the full version of our objective. We also evaluate directly taking the CLIP embedding of the target attribute – either its general embedding as represented by the EOS token, or the relevant subject token. Both versions are similarly disentangled as our CLIP difference method, but substantially underperform compared to it in subject-specificity.

	(a) Subject-Specificity	(b) Disentangledness	(c)	(d) Performance
Method	Subject-Specificity \uparrow	$\Delta\text{Id} \downarrow$	LPIPS \downarrow	Continuous Time \downarrow
Ours	<u>3.35</u>	0.40	0.10	✓ 12.0s [4.17it/s]
Ours (w/o Delay)	3.47	<u>0.50</u>	<u>0.22</u>	✓ 12.0s [4.17it/s]
Ours but optimize $\ \tilde{\mathbf{e}}_+ - \hat{\mathbf{e}}_\theta(\mathbf{x}_t \mid \mathbf{e} + \Delta\mathbf{e})\ $ (no λ_i)	2.23	0.55	0.31	✓ 12.0s [4.17it/s]
Our CLIP Difference Method (Sec. 3.2)	2.38	1.20	0.58	✓ 12.0s [4.17it/s]
CLIP Delta without Difference: $\Delta\mathbf{e}_{A_i} = (\mathcal{E}_{\text{CLIP}}(P_+))_{[\text{EOS}]}$	1.98	1.16	0.58	✓ 12.0s [4.17it/s]
CLIP Delta without Difference: $\Delta\mathbf{e}_{A_i} = (\mathcal{E}_{\text{CLIP}}(P_+))_{[S_j]}$	1.83	1.20	0.60	✓ 12.0s [4.17it/s]
Directly modulating $\Delta\tilde{\mathbf{e}}$ (Sec. 3.3) with CFG	3.15	0.73	0.39	✓ 23.0s [2.17it/s]

Table A. Extended version of Tab. 1 with additional ablations/baseline versions of our method.

We also compare with alternative approximations for the noise-space direction $\Delta\tilde{\mathbf{e}}$ we are learning in the tokenwise text embedding space as $\Delta\mathbf{e}_{A_i}$. Generally, other approaches to approximate these attribute- and sample-specific directions will not exhibit subject-specificity, so we perform this investigation in the single-subject case. We compare with two baselines that attempt to directly approximate $\Delta\tilde{\mathbf{e}}$: averaging it over the diffusion timestep t on a per-sample basis and averaging it over samples on a per-timestep basis. We compare them with the actual $\Delta\tilde{\mathbf{e}}$ in Fig. A. We find that both of these approximations, despite still having a dependency on either t or \mathbf{x}_T , only achieve a low similarity to the actual direction they attempt to approximate, while our directions $\Delta\mathbf{e}_{A_i}$ consistently outperform both approximations over all t .

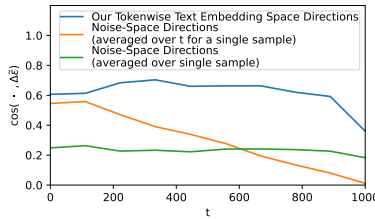


Figure A. Cosine Similarities of approximations of $\Delta\tilde{\mathbf{e}}$ compared to the actual true one over the diffusion timestep t .

A.2. Challenging Attributes

Some attributes are known in the community to be specifically challenging to get to work in practical settings. We show some successful examples of applying Attribute Control to them in Fig. B. Color attributes (Fig. Ba) are known to be prone to leakage across different objects, even in the base model. Our method generally inherits these limitations from the base model and can not address cases where the original prompt already leads to attribute leakage. When *adding* new attributes to the generated image, such as specifying the color for one object, we empirically find our modulations to lead to less (but still not zero) leakage. Intuitively, this makes sense, as we do not add an additional token describing the color change, which could be leaked to later tokens by the CLIP model and which any head of the diffusion model could attend to. Instead, we *exclusively* add the information to the token that describes that object. However, as diffusion cross-attention maps are not fully leakage-free unless applying methods that deliberately enforce this [7, 45, 58], we still observe color leakage with attribute control, although to a lesser extent. This especially occurs when leakage is already present in the base generation or when too much control is exerted (as shown in Fig. Ba). Similarly, cases where the base model is prone to leakage (e.g., trying to affect dogs and cats separately) are less prone to attribute leakage when adding them via our method (see, e.g., Fig. Bc). For attributes where the base model already struggles to apply them at all, our method inherits these limitations. Such attributes

like spatial relations *can* work (see Fig. Bb), but only do so (very) rarely, reflecting the base model’s inability to parse them from normal prompts reliably.

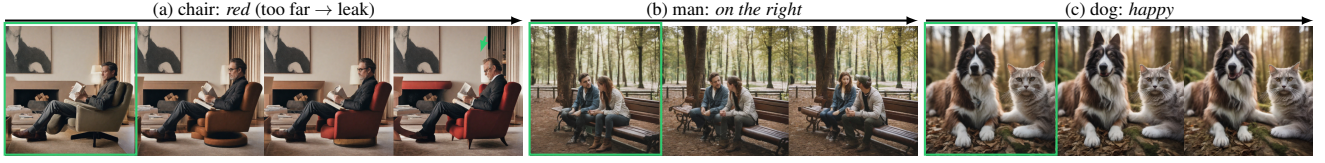


Figure B. Vanilla Attribute Control in challenging settings.

Postfix Attribute Learning Some attributes are not easily expressible as prefixes to the noun. This means that, due to the causal nature of the CLIP text encoder, our optimization-free method for identifying attribute directions (see Sec. 3.2) can not be applied. However, we find that this limitation does not apply to our optimization-based approach (see Sec. 3.3): we can learn directions based on attributes expressed as postfixes (e.g., “*a person wearing sunglasses*”, for which we show a qualitative example in Fig. C).



Figure C. Our learning-based method can also learn to represent attributes represented as postfixes to the target subject noun during training.

A.3. Subject Noun Transferability

We investigate how much our learned attribute modulations can generalize across different nouns that describe the same subject. We generally learn them on a set of different nouns that describe a subject of a specific category (e.g., for people with the words “man”, “woman”, and “person”). However, these words typically do not cover the whole range of possible nouns that can be used to describe subjects of a general category. Ideally, one could learn one modulation for one concept, such as age, on a small set of nouns and generalize across all nouns of a category or even to subjects of other categories.

First, we test the generalization of modulations learned for people on “man”, “woman”, and “person” and apply them to increasingly more specific nouns that describe people. Results are shown in Figs. D and E, and all prompts are “a photo of a beautiful <noun>”. As a baseline, we apply them to “child”, “mother”, and “father”, three words that are previously unseen but still describe very high-level sub-categories of people. We find that the learned modulations still work as expected. Similarly, for categories of jobs such as “doctor”, “barista”, or “firefighter”, which are substantially more specific and also substantially affect their clothing and the rest of the image, we find that they also work well. Finally, applying these learned modulations to very specific nouns such as the names “John” and “Jane” also works as expected. This demonstrates that our learned modulations can generalize well across a wide range of unseen nouns describing instances of a specific category, even if they were only learned on a small set of high-level, potential nouns.



Figure D. **Subject Noun Transferability.** We stress-test applying modulations that have been learned only on the nouns “man”, “woman”, and “person” to various other nouns that describe people. The unmodified image is marked in **green**. All samples are generated using attribute modulations being applied with a linear scale from -2 to 2 across each.

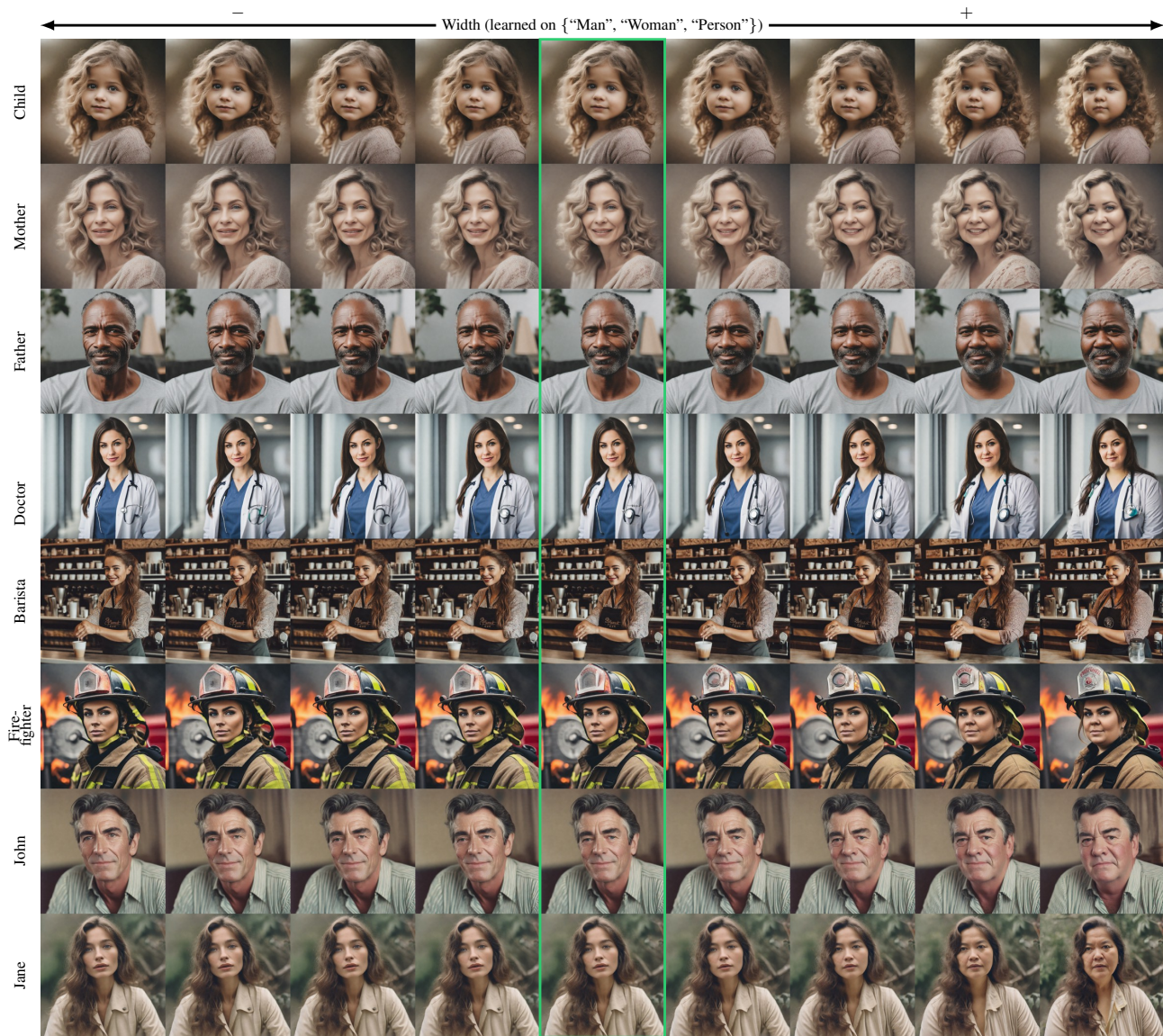


Figure E. **Subject Noun Transferability.** We stress-test applying modulations that have been learned only on the nouns “man”, “woman”, and “person” to various other nouns that describe people. The unmodified image is marked in **green**. All samples are generated using attribute modulations being applied with a linear scale from -2 to 2 across each.

A.4. Multi-Subject Attribute Editing

Figs. **F** and **G** show examples of modulating attributes in a subject-specific manner using our learned modulations. These show that various attributes can be applied to subjects individually, even if both subjects are of the same category (e.g., “people”). A slight correlation between, e.g., the age of the man and the age of the woman in Fig. **F** is visible and expected, as the diffusion model also models these dependencies between different subjects in the generated image. By applying both modulations with different strengths, the whole spectrum of combinations can be achieved, as shown in Fig. **9**.



Figure F. **Multi-Subject Attribute Modifications.** The unmodified image is marked in **green**. All samples are generated using one attribute modulation each being applied to the two subjects mentioned in the prompt with a linear scale from -2 to 2 across each.



Figure G. **Multi-Subject Attribute Modifications.** The unmodified image is marked in **green**. All samples are generated using one attribute modulation each being applied to the two subjects mentioned in the prompt with a linear scale from -2 to 2 across each.

A.5. Compositional Attribute Editing

We show some 2d grids where two attributes are modulated for the same target subject in an additive manner in Figs. H and I. Both attribute modulations interact with each other according to the world knowledge of the diffusion model to produce a realistic image for every combination.



Figure H. **Compositional Attribute Modifications.** The unmodified image is marked in **green**. All samples are generated using two attribute modulations being applied additively with a linear scale from -2 to 2 across each.

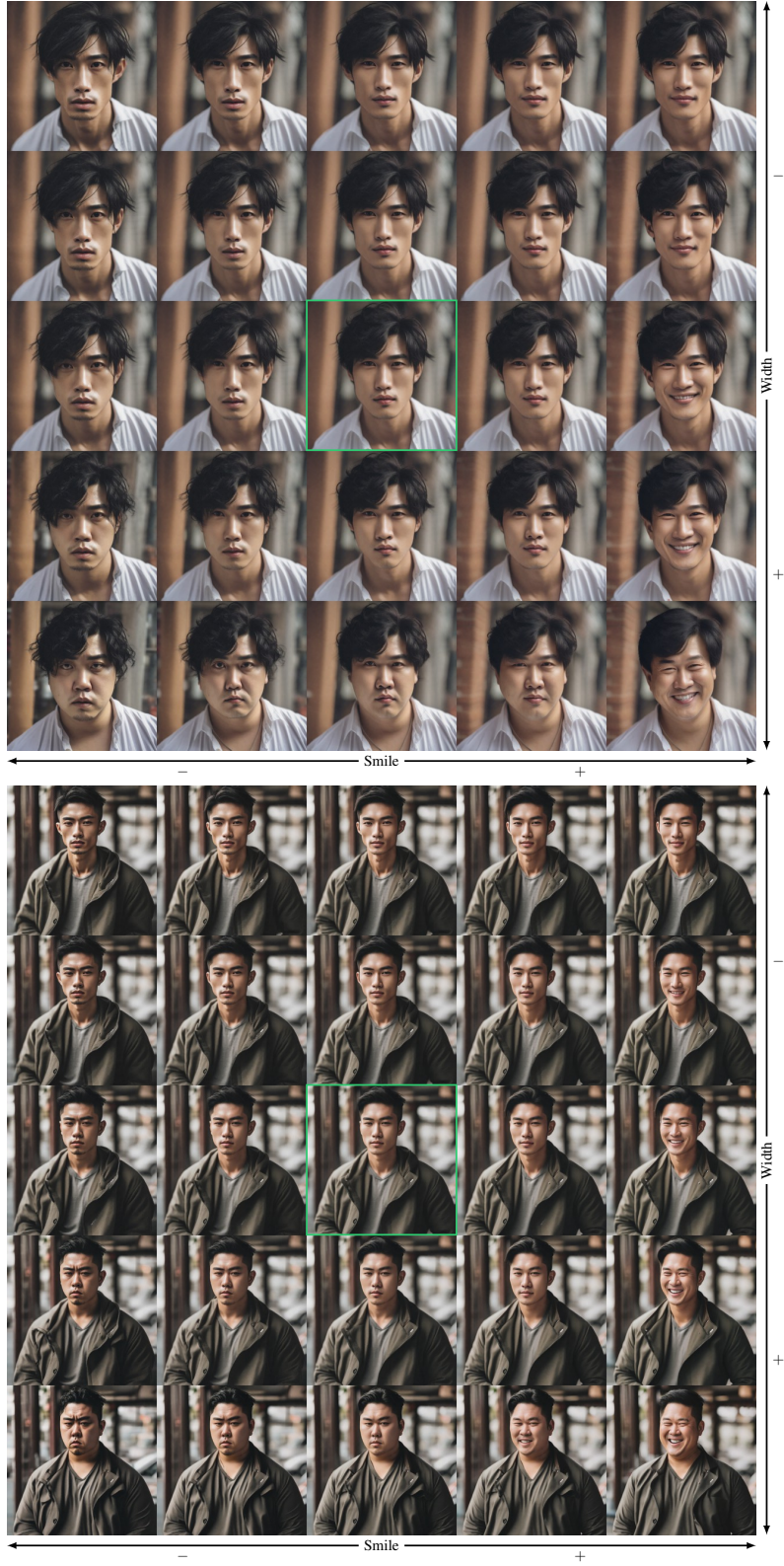


Figure I. **Compositional Attribute Modifications.** The unmodified image is marked in **green**. All samples are generated using two attribute modulations being applied additively with a linear scale from -2 to 2 across each.

A.6. Continuous Attribute Modulation

To illustrate the breadth of attributes that can be modulated and how continuous the attribute changes are, we show a range of attributes being continuously modulated. Figs. J to M show examples where attribute modulations are applied with our delayed sampling, Fig. N shows attribute modulations applied for the full sampling time. For every category, we re-use the same sample instances as a starting point.

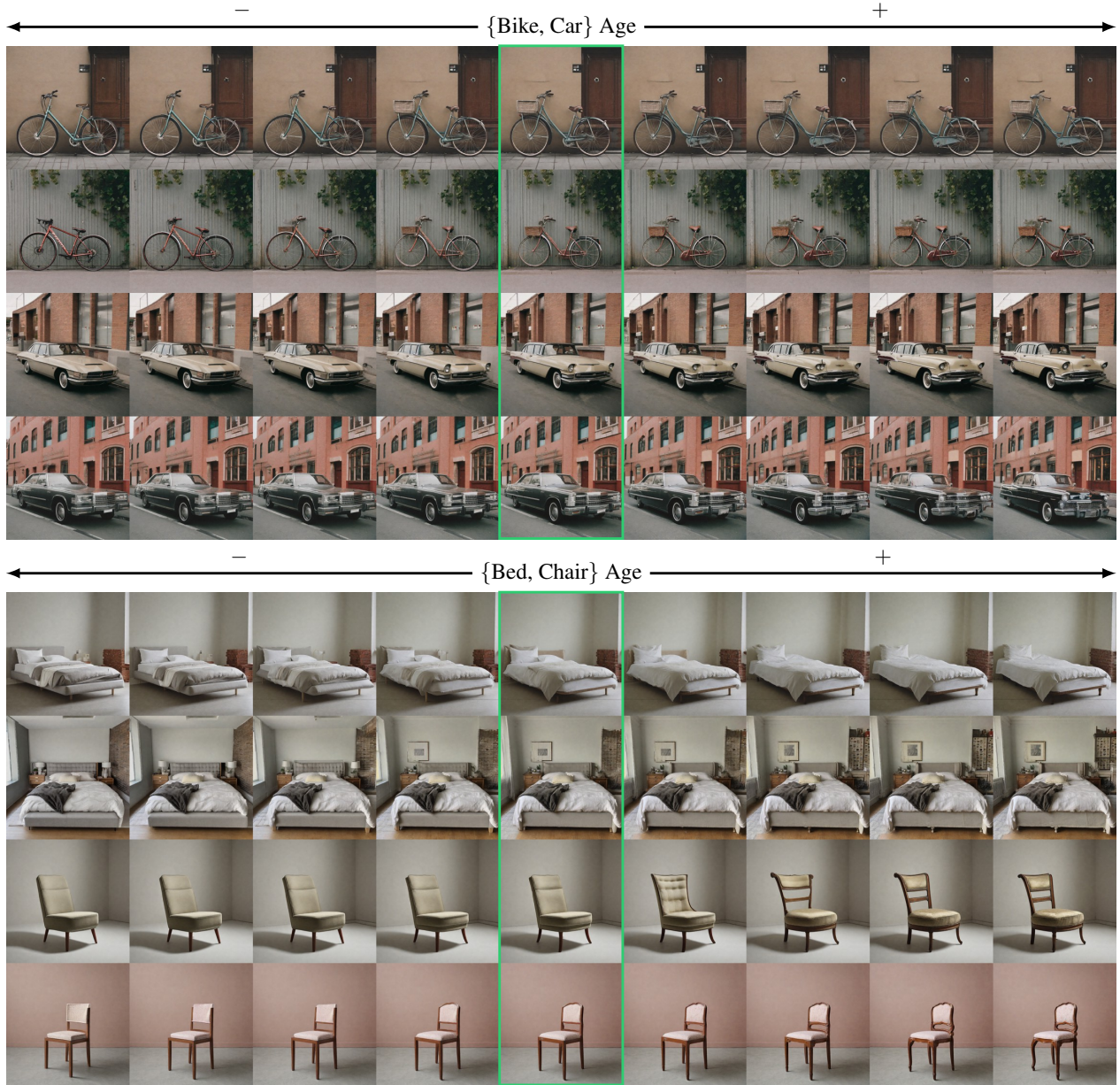


Figure J. **Continuous Attribute Modifications.** Unmodified images are marked in **green**. All samples are generated using a linear scale from -2 to 2.

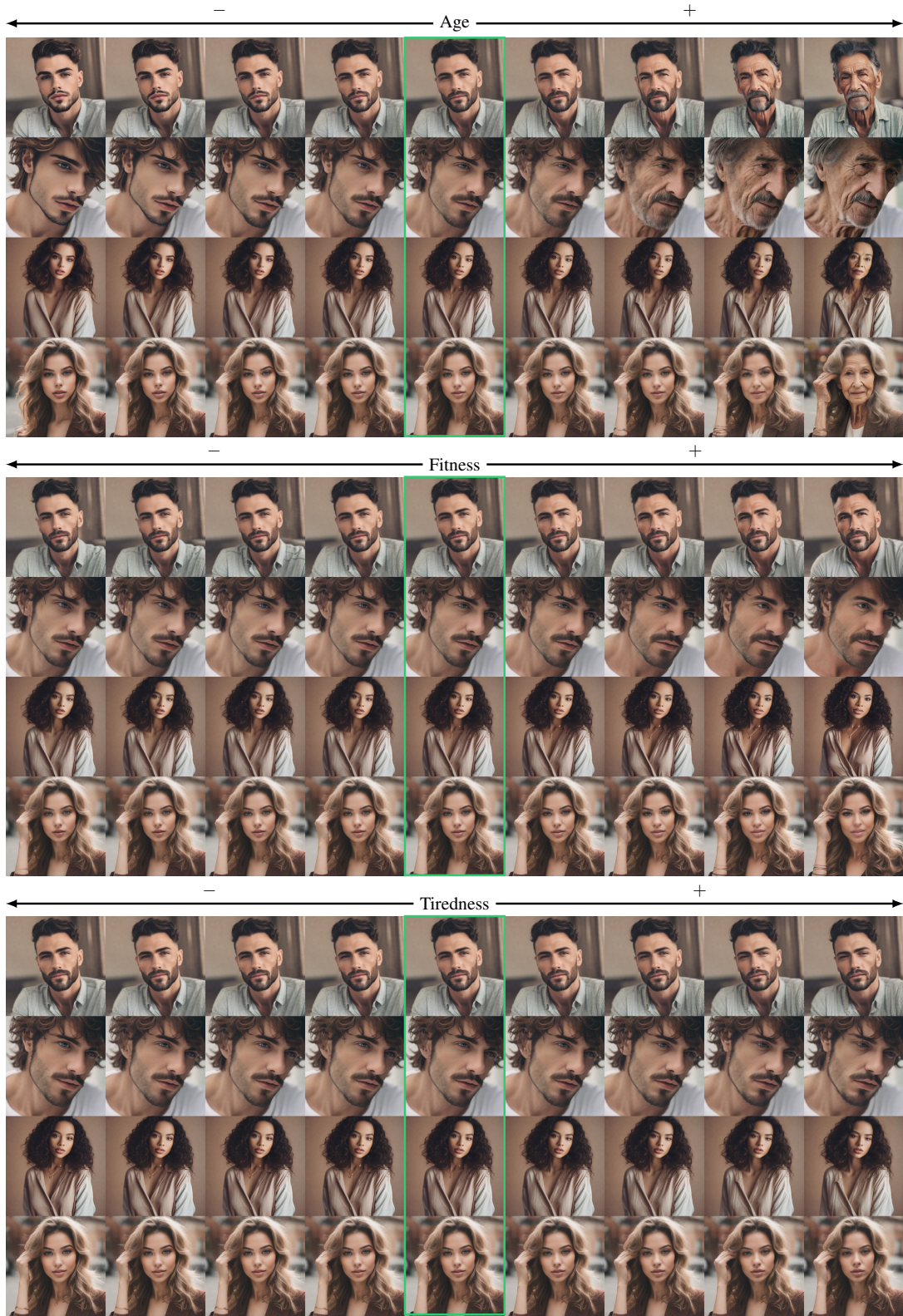


Figure K. **Continuous Attribute Modifications.** Unmodified images are marked in **green**. All samples are generated using a linear scale from -2 to 2.

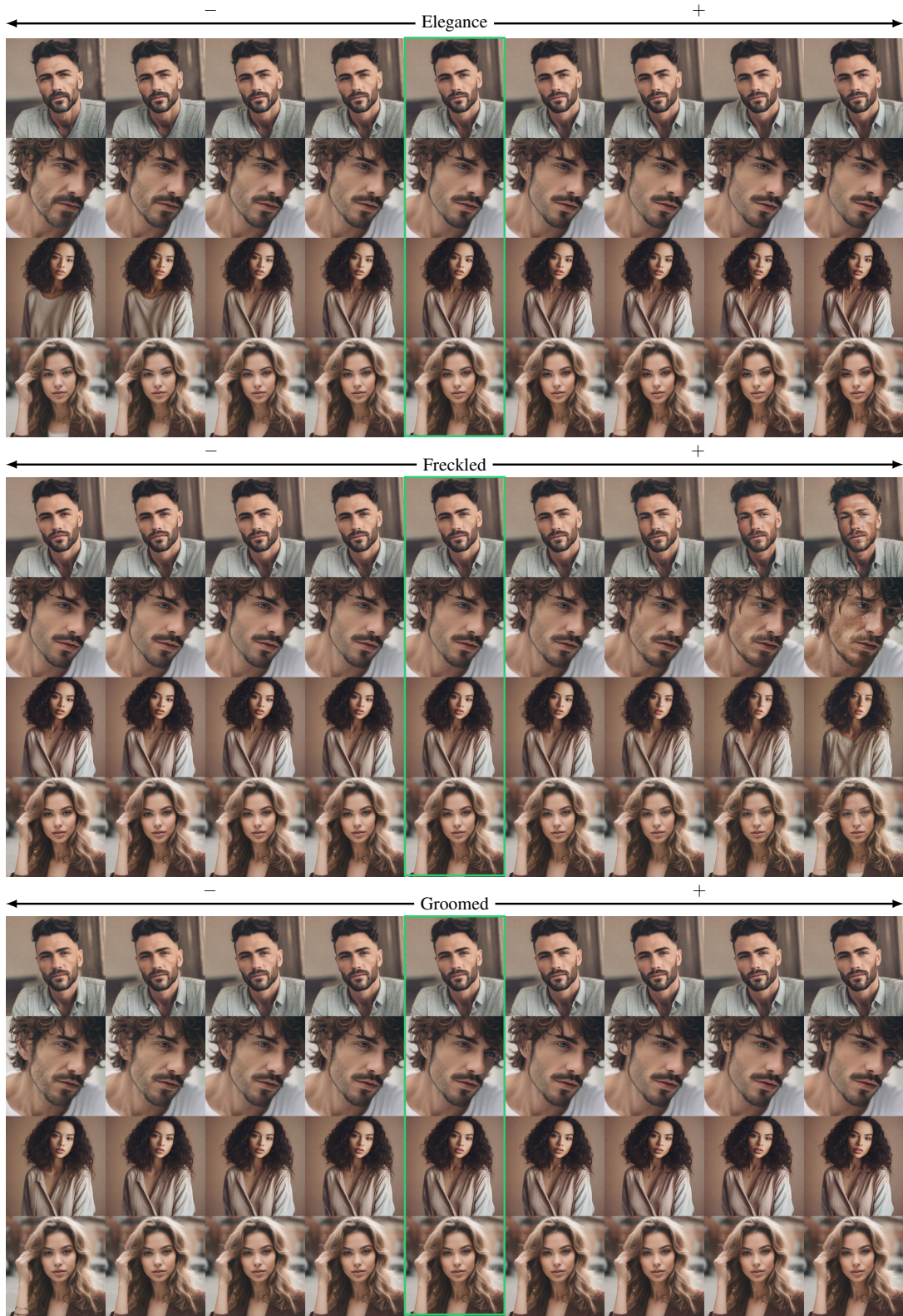


Figure L. **Continuous Attribute Modifications.** Unmodified images are marked in **green**. All samples are generated using a linear scale from -2 to 2.



Figure M. **Continuous Attribute Modifications.** Unmodified images are marked in **green**. All samples are generated using a linear scale from -2 to 2.

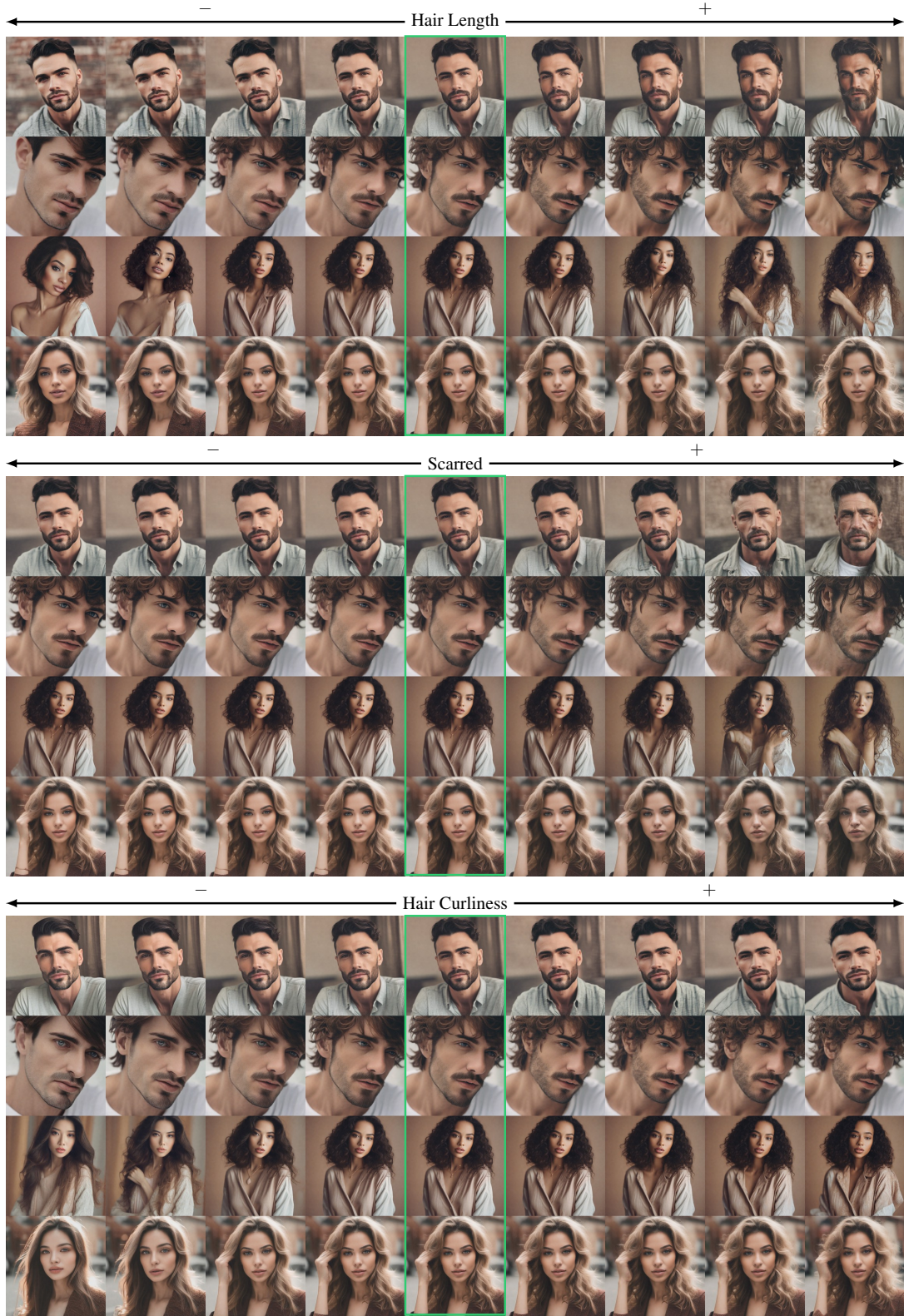


Figure N. **Continuous Attribute Modifications.** Unmodified images are marked in **green**. All samples are generated using a linear scale from -2 to 2, with the modulations being applied for all steps (w/o Delay).

B. Implementation Details

This section gives details about the implementation of our method. We generally use the default settings as set in `diffusers`²-v0.25.0 with a classifier-free guidance [18] scale of 7.5 and 50-step DDIM [50] sampling unless specified otherwise.

B.1. Semantic Direction Training

Algorithm 1 Algorithm for Learning the Semantic Directions

```

1: Input:
   Pre-trained diffusion model  $\hat{\epsilon}_\theta$ 
   CLIP embedding dimension  $d_{\text{CLIP}}$ 
   Learning rate  $\eta$ , number of steps  $S$ , batch size  $B$ 
2: Output:
   Learned semantic direction  $\Delta \mathbf{e}_{A_i}$ 
3: Initialize  $\Delta \mathbf{e}_{A_i} = \mathbf{0}$  ▷ Initialization
4: for  $s = 1$  to  $S$  do ▷ Training loop
5:    $\mathcal{L}_{\text{batch}} \leftarrow 0$  ▷ Initialize batch loss
6:   for each entry in batch of size  $B$  do
7:     Sample random subject  $S_j$  and neutral prompt  $P$ 
8:     Generate image  $\mathbf{x}_0$  from neutral prompt  $P$ 
9:      $t \sim \mathcal{U}[0, T]$  ▷ Sample random timestep
10:     $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I})$  ▷ Add noise
11:     $\tilde{\epsilon} = \hat{\epsilon}_\theta(\mathbf{x}_t | P)$  ▷ Predict noise for  $P$ 
12:     $\tilde{\epsilon}_+ = \hat{\epsilon}_\theta(\mathbf{x}_t | P_+)$  ▷ Predict noise for  $P_+$ 
13:     $\Delta \tilde{\epsilon} = \tilde{\epsilon}_+ - \tilde{\epsilon}$  ▷ Compute noise direction
14:     $\lambda_i \sim \mathcal{U}([-5, 5] \setminus (-0.1, 0.1))$  ▷ Sample scale factor
15:     $\mathcal{L}_i = w(t) \|(\epsilon + \lambda_i \Delta \tilde{\epsilon}) - \hat{\epsilon}_\theta(\mathbf{x}_t | \mathbf{e}', (\mathbf{e}, \lambda_i \Delta \mathbf{e}_{A_i}), t)\|_2^2$  ▷ Compute loss for this entry
16:     $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}} + \mathcal{L}_i$  ▷ Accumulate batch loss
17:   end for
18:   Compute mean loss for the batch:  $\mathcal{L}_{\text{mean}} \leftarrow \frac{1}{B} \mathcal{L}_{\text{batch}}$ 
19:   Update  $\Delta \mathbf{e}_{A_i}$  using AdamW optimizer with learning rate  $\eta$  based on  $\mathcal{L}_{\text{mean}}$ 
20: end for
21: Return:  $\Delta \mathbf{e}_{A_i}$ 

```

The semantic directions $\Delta \mathbf{e}_{A_i}$ for target attribute A_i are implemented as learnable parameters of shape $1 \times d_{\text{CLIP}}$, with d_{CLIP} being the embedding dimension of the CLIP text encoder. For SDXL [40], this is 2048, resulting from the channelwise concatenation of embeddings from the OpenAI CLIP ViT-L [42] and OpenCLIP ViT-bigG [23]. This direction is applied additively with scaling according to Eq. (3) to the target subject tokens (e.g., “person” in the case of “a photo of a person”) in the original text embedding \mathbf{e} . If the target subject consists of multiple tokens, we broadcast $\Delta \mathbf{e}_{A_i}$ across those tokens, although this is only very rarely the case in practice. Similarly, if one subject is mentioned in the prompt multiple times, we apply the same modulation to all instances.

We train our semantic directions $\Delta \mathbf{e}_{A_i}$ for 1000 steps³ at a batch size of 10. We use AdamW [31] with a learning rate of 0.1, $(\beta_1, \beta_2) = (0.5, 0.8)$, and weight decay of 0.333. All directions are trained on a single A100 with 40GB of VRAM using a bfloat16 version of SDXL [40].

For every entry in the batch, we use a random combination of prefix prompt (e.g. “an photo of”, optionally with attributes such as ethnicity (e.g., {asian, african-american, caucasian, arab, african, south-american, indian, ...})), to focus the implied direction on one that is invariant to these attributes) and prompt tuple (e.g “a woman”) and sample an image with the neutral prompt (e.g. “a photo of a woman”) and a random seed, stopping at a random timestep. We then compute the prediction starting from that step for all two/three prompts, resulting in $\tilde{\epsilon}$, $\tilde{\epsilon}_+$, and optionally $\tilde{\epsilon}_-$. In contrast to [14], we explicitly distill the full direction implied by $\Delta \tilde{\epsilon}$ by using multiple scales λ_i sampled from a continuous scale distribution. Preliminary

²<https://github.com/huggingface/diffusers>

³The directions tend to be mostly converged after 10 steps, but we train for a unified training time across all attributes for consistency.

experiments showed that this helps obtain substantially more robust directions. Additionally, we sample our starting samples using standard sampling instead of a modified generation process.

We then sample four values for $\lambda_i \sim \mathcal{U}([-5, 5] \setminus (-0.1, 0.1))$ and compute our training loss (Eq. (4)) over them. We found that sampling multiple values for λ_i substantially boosts the quality of our learned directions at little overhead cost (as the online sampling of the original images is the most costly part) and that values for λ_i very close to zero were not particularly useful for the training process. Empirically, we find that most of our learned directions are already close to convergence after five optimization steps, but we keep training for the full time for simplicity.

B.2. Combination of Attribute Control with other Methods

In Sec. 4, we combine our attribute control method with other off-the-shelf controlled generation methods.

Combination with Prompt-to-Prompt [17] To combine our method with Prompt-to-Prompt, we apply the standard Prompt-to-Prompt method. We use the same adaptation mode and hyperparameters as used for adding adjectives in the text prompt, but add our modulations on the text prompt embedding instead. To modulate the change, we scale our directions as usual.

Combination with AdapEdit [32] AdapEdit uses the same general external interface as Prompt-to-Prompt. Here, we apply our modulations in the exact same way as previously described for Prompt-to-Prompt. As AdapEdit is not available for SDXL [40], we use zero-shot adaptation of our semantic directions obtained on SDXL to SD1.5, as described in Sec. 4.2.

Combination with ReNoise [15] To apply our controlled generation approach to editing, we combine it with ReNoise, a standard inversion approach. We use their official reference implementation based on SDXL Turbo [47] and apply our modulations learned on SDXL there. We perform inversion purely with ReNoise with default settings and an image description prompt to obtain a starting latent \mathbf{x}_T , and then perform controlled generation purely with our method with standard settings. This could optionally be combined further with other methods during inference, such as Prompt-to-Prompt [17] and AdapEdit [32].

B.3. Experiment Evaluation Details

To compute perceptual image differences, we use LPIPS [60] as implemented in the `lpips`⁴ package with default settings at a resolution of 256^2 (interpolated bi-linearly). For CLIP scores, we use the standard implementation in `torchmetrics`⁵ (which outputs cosine similarities scaled to $[0, 100]$) with default settings, including the default CLIP choice of the CLIP-ViT-L/14 trained by OpenAI [42]. For image-image similarity evaluations with DINOv2 [37], we use the ViT-L/14 variant with registers [10] and bi-linearly resize to 224^2 before passing them to the model and comparing the cosine similarity of the CLS token outputs. Finally, for ReID evaluations, we use the ArcFace [11] implementation provided by the `insightface`⁶ python package with the default `buffalo_l` model, where we compute the cosine similarity of the embeddings of the detected faces.

Implementations of other Methods For Concept Sliders [14], we use the official public implementation⁷. For Prompt-to-Prompt [17], we use RoyiRa’s unofficial port of the method to Stable Diffusion XL⁸. This implementation also served as the basis for integrating our method with Prompt-to-Prompt in our codebase. As this implementation is partially incomplete, we referred to the official implementation Prompt-to-Prompt⁹ for the implementation of reweighting of added words. For AdapEdit¹⁰, MasaCtrl¹¹, and ReNoise¹², we also used the respective official implementations. When comparing attribute modulation capabilities across different methods, we compare using the target attribute age on people, as this attribute is i) unambiguous in what exactly it describes, ii) fully continuous, and iii) the attribute supported by Concept Sliders¹³ that can be evaluated most objectively while being one that SD(XL) can readily interpret when given as text (unlike, e.g., eye size).

⁴<https://github.com/richzhang/PerceptualSimilarity>

⁵<https://github.com/Lightning-AI/torchmetrics>

⁶<https://github.com/deepinsight/insightface>

⁷<https://github.com/rohitgandikota/sliders>

⁸<https://github.com/RoyiRa/prompt-to-prompt-with-sdxl>

⁹<https://github.com/google/prompt-to-prompt>

¹⁰<https://github.com/AnonymousPony/adap-edit>

¹¹<https://github.com/TencentARC/MasaCtrl>

¹²<https://github.com/garibida/ReNoise-Inversion>

¹³https://sliders.baulab.info/weights/xl_sliders/

Attribute Distribution Shifts (Figure 6) For each value of $\lambda_i \in \{0, 1, 2, 3\}$, 20 samples (with fixed seeds across scales) were drawn. We compute the delta CLIP score as specified in the experiments section of the paper and use scipy’s Gaussian KDE method¹⁴ to compute the kernel density estimate for the resulting distributions with Scott’s rule and default settings.

Qualitative Continuous Modulation (Figure 7) We continuously modulate the age of the person described in the prompt with both our method and Concept Sliders [14], choosing coefficients such that a wide range is covered and both methods show similar scales per column. For Prompt-to-Prompt [17] and MasaCtrl [6], we add “old” or “young” to the prompt to coarsely modulate the target attribute. Prompt-to-Prompt further enables some fine-grained control *around the already offset attribute expression point from the added adjective* by re-weighting the added adjective. This does, at least for Stable Diffusion XL [40], not allow continuous modulation back to the original image, causing a discontinuity. This can intuitively be explained by the fact that attributes are aggregated in the subject noun, a fact that our method exploits to directly enable fine-grained, subject-specific target attribute modulation: as the attribute modulation for P2P is already partially contained in the subject noun, modulating just the added adjective’s cross-attention map can not fully recover the original generated image. At the same time, when combined with our method, where we just modulate the target subject noun’s embedding instead of adding new adjectives, this problem immediately subsides.

Quantitative Subject Specificity Evaluation (Table 1a) With each method, we generate variations across a set of 50 images with individual prompts describing two people, where we modulate the target attribute of one of the two subjects. We detect each subject in the unmodified image as previously described with the standard pipeline from *insightface*, and then compute the target metric for each bounding box. We aggregate the specificity metric as described in Eq. (6) by computing the fraction individually per sample and then aggregating the overall mean. As there are some cases where this effectively results in a division by zero, we clamp the resulting individual values to $[0, 10]$. We chose 10 as a threshold, as it prevents these outlier samples from having an extraordinarily strong effect on the overall mean.

Attribute Coverage Evaluation (Figure 9) To evaluate the set of attribute combinations reachable by each method, we start from the same setup as previously described for Table 1a, but continuously modulate the age for both subjects visible in the image, covering all combinations of modulation scales for each method. We evaluate 20 values per subject, producing 400 generated samples per method for methods that allow independent continuous modulation of both subjects. We then measure the attribute expression for each subject bounding box (obtained as previously in Table 1a) using Eq. (8) and plot the distribution for one representative sample in Fig. 9.

Quantitative Disentangledness Evaluation (Figure 10, Table 1b) We generate 50 base samples showing people with different prompts of the format “a close-up portrait of a {modifiers} {woman, man}”, where {modifiers} describes a set of prefixes (e.g., “{ \emptyset , beautiful, elegant} asian”, “{ \emptyset , beautiful, elegant} african-american”, etc) to cover a wide variety of different images. Then, we modulate the target attribute continuously using each method. We then measure the attribute expression change with Eq. (8), the image change with LPIPS, and the identity change as in Eq. (7). We aggregate these values over all 50 images per combination of method & hyperparameters and then plot them in Fig. 10. For Table 1b, we compute the slope of these graphs (using the absolute value of ΔCLIP_{B_i} for the denominator, to account for the fact that the changes increase for positive values and one for negative values of ΔCLIP_{B_i}) to quantify the disentangledness of the edits both from overall visual changes (LPIPS) and person identity changes (ΔId).

Inference Performance Evaluation (Table 1d) For each method, we use the released implementations of each respective method with default settings and replicate the original environments as closely as possible, given the information documented by the authors. We measure inference times on the same Nvidia A100 SXM with 80GB of VRAM and document both the total time and (average) step time, as some methods use different step counts for sampling. For the main paper, we consolidate inversion and generation time if applicable. We exclude the time spent obtaining attribute deltas, as it is done once ahead of time and causes no overhead during inference/amortizes quickly when needing to train deltas for new attributes, similar to Concept Sliders [14], where we also exclude slider training time due to the same reason.

¹⁴https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.gaussian_kde.html

C. Visualization Details & Prompts

Generally, all examples in the paper use Stable Diffusion XL as introduced by Podell et al. [40] unless noted otherwise. In the following, we provide the prompts and, in the case of editing examples, image sources incl. licenses, used to generate the various qualitative examples presented in the paper.

Figure 1 Prompt: *“A close-up photo of a man and a woman sitting on a bench.”*

Figure 2 Prompts: *“a portrait of a beautiful car”, “a portrait of a beautiful frog”, and “a portrait of a beautiful suv”.*

Figure 3 Prompt: *“a portrait of a beautiful woman with her beautiful dog”.*

Figure 4 Prompt: *“a photo of a car”.*

Figure 6 Prompt: *“a photo of a car”.*

Figure 7 Base prompt: *“a close-up portrait of a indian woman”.*

Figure 8 Image 1 is a photo with the title *“a red rolls royce parked in front of a building”* by Rico Reynaldi, obtained from Unsplash¹⁵. The image is licensed under the Unsplash license¹⁶ and has been center-cropped for inversion.

Inversion Prompt: *“a photo of a beautiful red car on the top deck of a parking garage with large buildings in the background, hazy weather with sunshine”.*

Image 2 is a photo by The Royal Society, obtained from Wikimedia¹⁷. The image is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license¹⁸ and has been cropped to primarily show the person’s head.

Inversion Prompt: *“a photo of a man wearing glasses and a suit”.*

Figure 11 Prompt: *“a photo of a beautiful asian man”.*

Figure 12 Prompt: *“a portrait of an indian woman standing next to an african-american man”.*

Figure 13 Prompt 1: *“a portrait of a beautiful chair”.*

Prompt 2: *“photo of an old car”.*

Prompt 3: *“a portrait of a beautiful truck”.*

Prompt 4: *“a photo of a beautiful man”.*

Figure 14 aMUSEd: *“a photo of a beautiful man”.*

SD 1.5: *“a headshot of a relaxed woman and a friendly man”.*

Figure 15a Prompt: *“a photo of a beautiful man”*

Figure 15b Prompt: *“a photo of a beautiful woman”*

Figure 15c Prompt: *“a close-up photo of a real beautiful man with his beautiful cat sitting in the forest, high detail, wide angle lens.”*

Figure Ba Prompt: *“A close-up photo of a man sitting in a chair. He is leaning back and reading a book. A sofa is seen in the background. modern aesthetic, architectural digest.”*

¹⁵<https://unsplash.com/photos/a-red-rolls-royce-parked-in-front-of-a-building-sAN11DGnjqk>

¹⁶<https://unsplash.com/license>

¹⁷https://commons.wikimedia.org/wiki/File:Demis_Hassabis_Royal_Society.jpg

¹⁸<https://creativecommons.org/licenses/by-sa/3.0/deed.en>

Figure Bb Prompt: *“A close-up photo of a man and a woman sitting on a bench. The setting is in the forest, high detail, wide angle lens”*

Figure Bc Prompt: *“A close-up photo of a dog sitting next to a cat. The setting is in the forest, high detail, wide angle lens”*

Figure C Prompt: *“A photo of a beautiful asian man”*

Figures D and E Prompt Template: *“a photo of a beautiful [...]”*

Figure F Prompt 1: *“a photo of a bearded man in a beanie enjoying a concert with a bohemian woman in flowing attire”*

Prompt 2: *“a portrait of an indian woman standing next to an african-american man”*

Figure G Prompt 1: *“a photo of a tech-savvy man with a laptop engaged in conversation with a creative woman with colorful tattoos”*

Prompt 2: *“a portrait of an indian woman dressed in traditional clothing next to an african-american man wearing a hat standing in a library”*

Figure H Prompt 1: *“a photo of a car”*

Prompt 2: *“a photo of a compact red car”*

Figure I Prompt 1 & 2: *“a photo of a beautiful asian man”*

Figure J Prompt 1 & 2: *“a photo of a bike”*

Prompt 3 & 4: *“a photo of a car”*

Prompt 5 & 6: *“a photo of a bed”*

Prompt 7 & 8: *“a photo of a chair”*

Figures K to N Prompt 1 & 3: *“a photo of a beautiful man”*

Prompt 2 & 4: *“a photo of a beautiful woman”*