Appendices

In this document, we present additional material to support the main paper. Firstly, in Section A for reproducibility purposes we provide implementation details and detailed algorithms for our method. Lastly, in Section B we provide additional experimental results, comparisons and an ablation study designed to supplement and further validate our proposed method.

A Algorithms & Implementation Details

A.1 Algorithms

In this Section, we provide an algorithm that describes our method as presented in Section 4.2 in the main paper. Additionally, we offer detailed algorithms for the Subspace Iteration method used to approximate the SVD of the Jacobian of the denoising network and the JIVE algorithm used to obtain a solution to the minimization problem in Eq. (5) of the main paper.
Computing Joint and Individual Components in the Latent Space of DMs: Algorithm 1 summarizes our method for decomposing the Jacobian of each region into a joint and an individual component. We start with a real image $\mathbf{x}_0$, a denoising network $\mathbf{e}_\theta$ and a set of regions $M = \{\mathbf{m}_1, \cdots, \mathbf{m}_N\}$, along with the denoising timestep t and joint and individual ranks r_C and r_A. Starting with $\mathbf{x}_0$, we obtain its corresponding noise $\mathbf{x}_T$ via DDIM Inversion [4]. We denoise $\mathbf{x}_T$ up to timestep t with the standard DDIM reverse process (see Eq. (3) in text). Then we obtain the SVD of the Jacobian of each region with the Subspace Iteration (Algorithm 2). Finally, by utilizing the JIVE (Algorithm 3) we obtain the joint and individual components $\mathbf{C}$ and $\mathbf{A}_i$ for each region i. The rows of $\mathbf{A}_i$ are latent directions that result in meaningful local edits within region i.

Algorithm 1 Computing Joint and Individual Components in the Latent Space of DMs

procedure ComputeIndivJoint($\mathbf{e}_\theta, \mathbf{x}_0, t, \mathrm{M}, r_C, r_A$)
  $\mathbf{x}_T \leftarrow \mathrm{DDIMInversion}(\mathbf{x}_0)$ ▷ DDIM Inversion [4]
  $\mathbf{x}_t \leftarrow \mathrm{DDIMReverse}(\mathbf{x}_T, t)$ ▷ DDIM Reverse process until timestep t
  for $i, \mathbf{m}$ in enumerate(M) **do**
    $\mathbf{U}^{(i)}, \mathbf{S}^{(i)}, \mathbf{V}^{(i)} \leftarrow \mathrm{SubspaceIteration}\left(\{\mathbf{e}^t_\theta(\mathbf{x}_t)\}_m\right)$ ▷ Algorithm 2
    $\mathbf{J}^{(i)}_\perp \leftarrow \mathbf{S}^{(i)}\mathbf{V}^{(i)T}$ ▷ Dimension-reducing transformation
  end for
  $\mathbf{J}_\perp = \left[\mathbf{J}^{(1)^T}, \cdots, \mathbf{J}^{(N)^T}\right]^T$
  $\mathbf{C}, \{\mathbf{A}_i\}_{i=1}^N \leftarrow \mathrm{JIVE}(\mathbf{J}_\perp, r_C, r_A)$ ▷ Algorithm 3
  return $\mathbf{C}, \{\mathbf{A}_i\}_{i=1}^N$
end procedure

Jacobian Subspace Iteration: As discussed in Section 4.2 the dimension of the latent space and the output image, result in a Jacobian matrix of approximately 6B parameters. To efficiently calculate the SVD of the Jacobian of the denoising network without storing it in

memory we rely on Jacobian Subspace Iteration as proposed by Haas et al. [1]. Note that differently from [1] we calculate the SVD of the Jacobian of the denoising network explicitly within a region of interest $\mathbf{m}$.

Algorithm 2 Jacobian Subspace Iteration

procedure SUBSPACEITERATION($\mathbf{e}_\theta, \mathbf{x}_t, \mathbf{m}$)
  $\mathbf{h}_t \leftarrow \mathbf{e}^t_\theta(\mathbf{x}_t)_{:\frac{l}{2}}$ ▷ Apply only the encoder to get bottleneck featuremaps
  $\mathbf{y}_m \leftarrow \{\mathbf{e}^t_\theta(\mathbf{x}_t)\}_m$ ▷ Output of denoising network in region $\mathbf{m}$
  $\mathbf{V} \leftarrow$ i.i.d standard Gaussian Samples
  $\mathbf{Q}, \mathbf{R} \leftarrow \mathrm{QR}(\mathbf{V})$ ▷ Reduced QR decomposition
  $\mathbf{V} \leftarrow \mathbf{Q}$ ▷ s.t. $\mathbf{V}\mathbf{V}^T = \mathbf{I}$
  while termination criterion **do**
    $\mathbf{U} \leftarrow \frac{\partial \{\mathbf{e}^t_\theta(\mathbf{x}_t|\ \mathbf{h}_t + a\mathbf{V})\}_m}{\partial a}\Big|_{a=0}$ ▷ Forward differentiation
    $\hat{\mathbf{V}} \leftarrow \frac{\partial \mathbf{U}^T \mathbf{y}_m}{\partial \mathbf{h}_t}$
    $\mathbf{V}, \mathbf{S}, \mathbf{R} \leftarrow \mathrm{SVD}(\hat{\mathbf{V}})$ ▷ Reduced SVD
  end while
  return $\mathbf{U}, \mathbf{S}, \mathbf{V},$
end procedure

Joint and Individual Variation Explained (JIVE) Algorithm: For completeness we provide the iterative JIVE algorithm [3]. Given a set of N matrices $\mathbf{X} = [\mathbf{X}^{(1)} \cdots \mathbf{X}^{(N)}]$, JIVE iteratively approximates their joint and individual components.

Algorithm 3 Joint and Individual Variation Explained

procedure JIVE($\mathbf{X}, r_C, r_A$)
  $\mathbf{X}_{joint} \leftarrow \left[\mathbf{X}^{(1)^T} \cdots \mathbf{X}^{(N)^T}\right]^T$
  while termination criterion **do**
    $\mathbf{C} = \left[\mathbf{C}^{(1)^T}, \cdots, \mathbf{C}^{(N)^T}\right] \leftarrow$ rank r_C SVD of $\mathbf{X}_{joint}$
    for $i = (1, \cdots, N)$ **do**
      $\mathbf{X}^{(i)}_{indiv} \leftarrow \mathbf{X}^{(i)} - \mathbf{C}^{(i)}$
      $\mathbf{A}^{(i)} \leftarrow$ rank r_A SVD of $\mathbf{X}^{(i)}_{indiv}(\mathbf{I} - \mathbf{V}\mathbf{V}^T)$ ▷ Ensures orthogonality constraint
      $\mathbf{X}^{(i)}_{joint} \leftarrow \mathbf{X}^{(i)} - \mathbf{A}^{(i)}$
    end for
    $\mathbf{X}_{joint} = \left[\mathbf{X}^{(1)^T}_{joint} \cdots \mathbf{X}^{(N)^T}_{joint}\right]^T$
  end while
  return $\mathbf{C}, \{\mathbf{A}\}_{i=1}^N$
end procedure

A.2 Impementation Details

Here we describe more thoroughly the experimental setup we use to produce our results in the quantitative comparisons in the main paper, in order to ensure reproducibility. For all methods compared we use an editing strength $a = 50$ along the identified semantic direction to obtain the edits. The alternative methods we benchmark our proposed method against are detailed below.

Asyrp For Asyrp [2] we use the author's official code[1]. Since Asyrp is a supervised approach we train each attribute with 1k samples following the recipe detailed in their paper. Asyrp uses CLIP to obtain the editing directions and requires a source caption y^{source} and a target caption y^{target}. The source and target captions we use to obtain the editing directions for each attribute are:

- Smile: $y^{\text{source}} =$ "*face*", $y^{\text{target}} =$ "*face with a smile*"
- Gaze: $y^{\text{source}} =$ "*face*", $y^{\text{target}} =$ "*face looking left*"
- Red Lips: $y^{\text{source}} =$ "*face*", $y^{\text{target}} =$ "*face with red lips*"
- Close Eyes: $y^{\text{source}} =$ "*face*", $y^{\text{target}} =$ "*face with closed eyes*"

Haas et al. We implement the method of Haas et al. [1] following their paper [2]. The editing directions for Smile, Close Eyes, Red Lips, and Gaze edits were obtained from CelebA-HQ samples with index numbers `00009`, `00000`, `00042`, `00003` respectively.

Ours For our method we obtain all editing directions from CelebA-HQ sample with index number `00020`.

B Additional Experimental Resuls

B.1 Ablation Study: Choice of joint and individual ranks

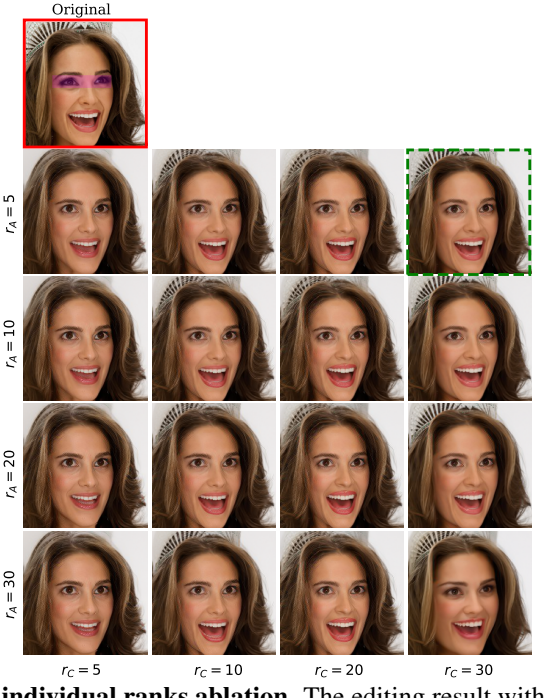

Figure 1: **Joint and individual ranks ablation**. The editing result with joint and individual ranks used for all our experiments i.e. $r_C = 30$, $r_A = 5$ is highlighted with a green rectangle.

Here we turn our focus on the effect of joint and individual rank selection on local editing. In Figure 1 we present the editing results for a localized attribute manipulation (open eyes) under various joint and individual rank choices. We highlight with a green rectangle the editing

[1] https://github.com/kwonminki/Asyrp_official

[2] At the time of writing the paper the official codebase of Haas et al. was not public yet.

result obtained $r_C = 30$ and $r_A = 5$ which are the ranks used for all experiments. We observe that a low joint rank results in undesirable edits outside the region of interest, like changing the mouth and the hairband. Similar effects are observed under a high individual rank. Conversely, a low individual rank and a high joint rank produce a localized manipulation that minimally affects the rest of the image.

B.2 Edits from different timesteps

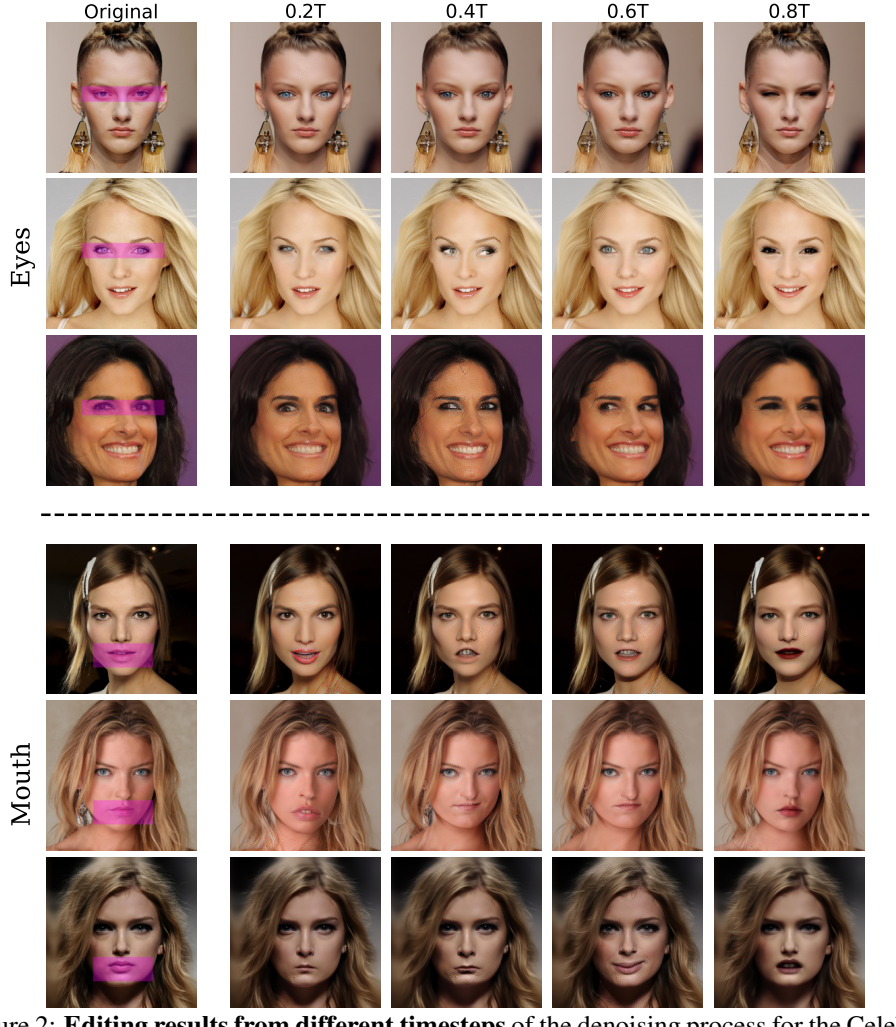

Figure 2: **Editing results from different timesteps** of the denoising process for the CelebA-HQ datasets Regions of interest are denoted by a pink rectangle. All edits presented correspond to the 1st principal component for each timestep.

In the main paper, we present editing results obtained at timestep $t = 0.6$T of the denoising process. In Figure 2 and Figure 3 we present editing results obtained from various timesteps on CelebA-HQ, LSUN-Churches, and METFACES. All edits presented correspond to the

first principal component for each timestep. We observe that our method is robust with respect to timestep selection, identifying meaningful semantic directions at various timesteps. For CelebA-HQ, these directions produce edits like changing the eye color or the gaze for the eyes region and changing the lip color and the expression for the mouth region. For LSUN-Churches, editing the window region produces different window variations, while for METFACES, editing the mouth region changes the subject's expression and facial hair.

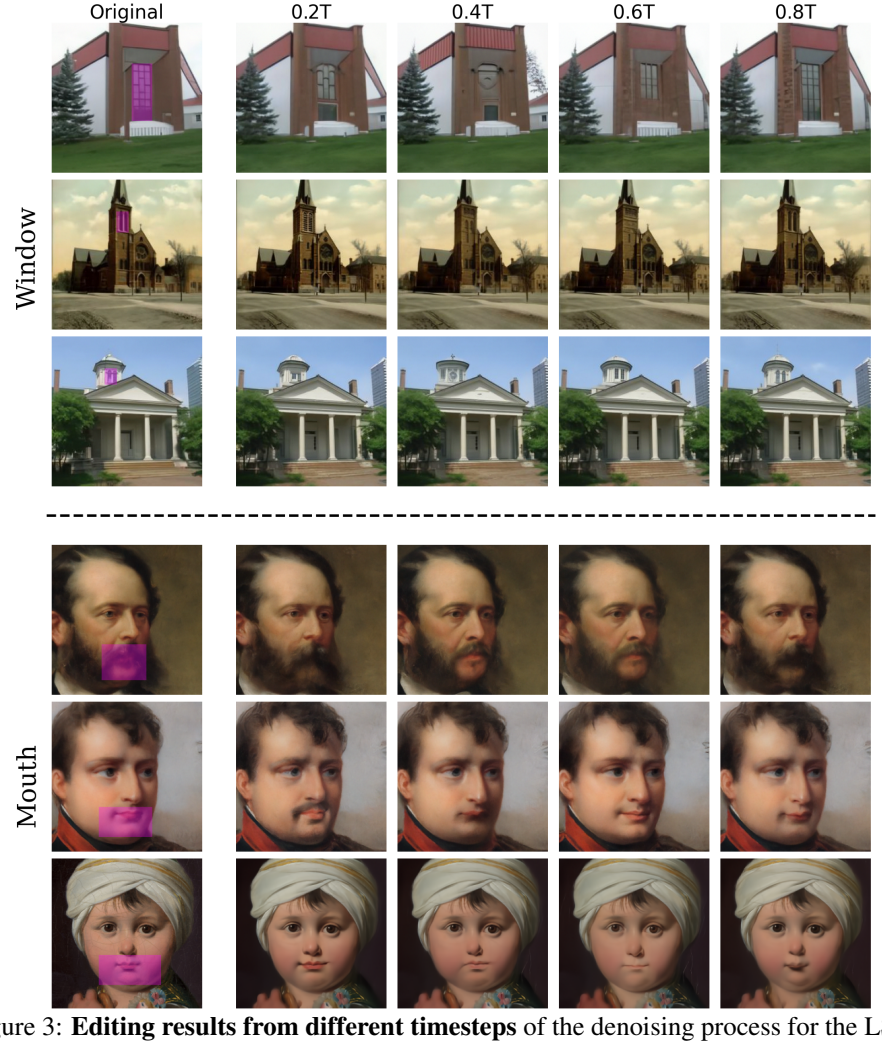

Figure 3: **Editing results from different timesteps** of the denoising process for the LSUN-Churches and METFACES datasets. Regions of interest are denoted by a pink rectangle. All edits presented correspond to the 1st principal component for each timestep.

B.3 Linear Interpolation between directions

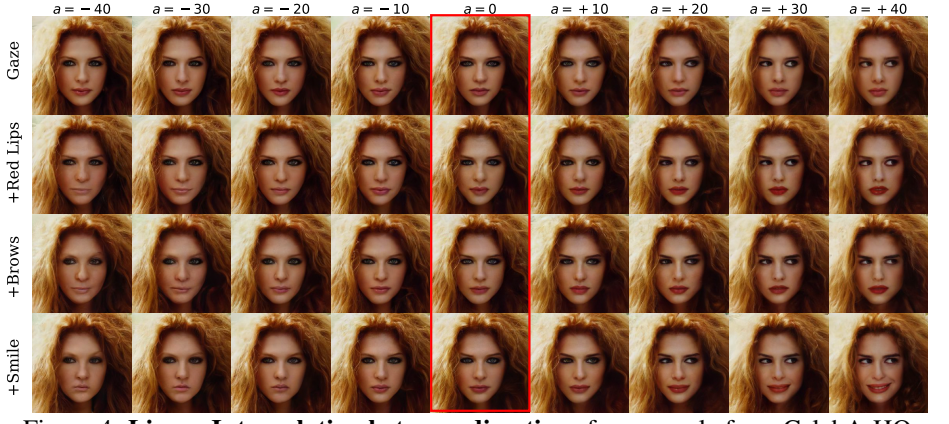

Figure 4: **Linear Interpolation between directions** for a sample from CelebA-HQ.

In Figure 4 we showcase how latent directions identified by our method can be composed to simultaneously edit different attributes. Let $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$ be the latent directions corresponding to *Gaze*, *Red Lips*, *Brows*, and *Smile* edits respectively. Each image in position (i, j) in Figure 4 is produced with editing vector $\mathbf{v}_{i,j} = a_i \sum_{k=1}^{j} \mathbf{v}_k$ with $a_i \in [-40, \ldots, 40]$ and $\mathbf{v}_k \in [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4]$, where a_i is the editing strength. We observe that scaling the strength of the edit controls the magnitude of attribute change and that the negative scales result in semantically opposite edits.

B.4 Additional Comparisons

Figure 5: **Additional qualitative comparisons** between our method and existing alternatives for Red Lips and Gaze edits.

In Figure 5 we present additional qualitative comparisons for two semantic directions not shown in the main paper. Asyrp [2] produces considerable distortions and artifacts in the edited regions and in some instances significantly alters the subject's identity, as in the second and third images of the Red Lips edit. Haas et al. [1] are better at preserving the subject's identity but the edits fail to be localized to the region of interest, changing the subject's overall expression. Our method succeeds in producing the desired semantic edits in the region of interest with almost imperceptible changes to the other parts of the image.

References

[1] René Haas, Inbar Huberman-Spiegelglas, Rotem Mulayoff, and Tomer Michaeli. Discovering interpretable directions in the semantic latent space of diffusion models. *arXiv preprint arXiv:2303.11073*, 2023.

[2] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. In *The Eleventh International Conference on Learning Representations*, 2022.

[3] Eric F Lock, Katherine A Hoadley, James Stephen Marron, and Andrew B Nobel. Joint and individual variation explained (jive) for integrated analysis of multiple data types. *The annals of applied statistics*, 7(1):523, 2013.

[4] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.