Patronus: INTERPRETABLE DIFFUSION MODELS WITH PROTOTYPES

Nina Weng, Aasa Feragen, Siavash Bigdeli
Technical University of Denmark
{ninwe, afhar, sarbi}@dtu.dk

Code is available at nina-weng.github.io/patronus.github.io.

APPENDIX

A Additional Implementation Details

A.1 Interpretability: Locating Most Activated Patches

Given an input image x_0 and a chosen prototype J, the most activated patch is determined as follows:

1. Compute the similarity score between each latent feature z_i, where $i \in \{1, 2, \ldots, N\}$, and the prototype p_J. Identify the closest z_i as:
$$i' = \arg\min_i D(z_i, p_J)$$
where $D(\cdot, \cdot)$ represents the chosen distance metric; in our case, it's the log transformed square $L2$ distance (Sec. 3.1).
2. Map the index i' back to the spatial coordinates of the feature map. Using the receptive field of the encoder, locate the corresponding patch in the original image, as shown in Fig. 5.

A.2 Interpretability: Emergence Analysis

Retrieve prototype activation vector along generation process for one image. For a given guidance s and starting noise x_T, we can sample a generated image x_0, where x_t denotes the intermediate state at each timestep. At each timestep t, the estimated denoised image $\hat{x}_0^t$ is computed as:

$$\hat{x}_0^t = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t} \cdot \epsilon_\theta(x_t, t, s)). \tag{8}$$

Here, the superscript t indicates that this estimate originates from timestep t. The prototype activation vector at timestep t is then obtained as: $s^t = Enc(\hat{x}_{0,t})$. By repeating this process across all timesteps t, we obtain the sequence $\{s^t\}_{t=0}^T$. Extracting the j-th index rom each activation vector yields $\{s_j^t\}_{t=0}^T$, which tracks how the j-th prototype is activated throughout the generation process for sample x_0.

Examples of how prototypes emerge differently over time during diffusion. Building on the illustrative example in Fig.1a, where specific semantic features are enhanced or suppressed using prototype activation vectors, we examine how the five selected prototypes emerge over time during the generation process. This is shown for the original image (Fig.6a), with the enhancement of the prototype "White collar" (Fig.6b), and with the enhancement of the prototype "Curly hair" (Fig.6c). Interestingly, when the "White collar" prototype is enhanced, its similarity score increases notably around timestep 700/1000 (Fig.6b). In contrast, when enhancing the "Curly hair" prototype, its similarity score begins to rise around timestep 200/1000 (Fig.6c). Note that larger t corresponds to an earlier stage of the diffusion process, which means "Curly hair" as a semantic information emerged later in diffusion generation process, compared to "White collar".

This observation is further confirmed by the estimated denoised $\hat{x}_0^t$ in Fig.6b and Fig.6c. White clothing appears at an early stage of the denoised images, whereas although the hair becomes fluffier, the fine-grained curly texture only emerges around timestep 200 (bottom of Fig.6c). This difference

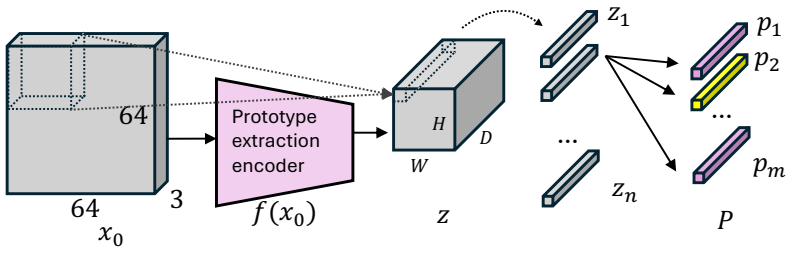


Figure 5: **Illustration on how to find the most activated patch given x_0 regarding prototype J.** Here the shape of x_0 ($64 \times 64 \times 3$) serves as an example and should be generalizable to other scenarios.

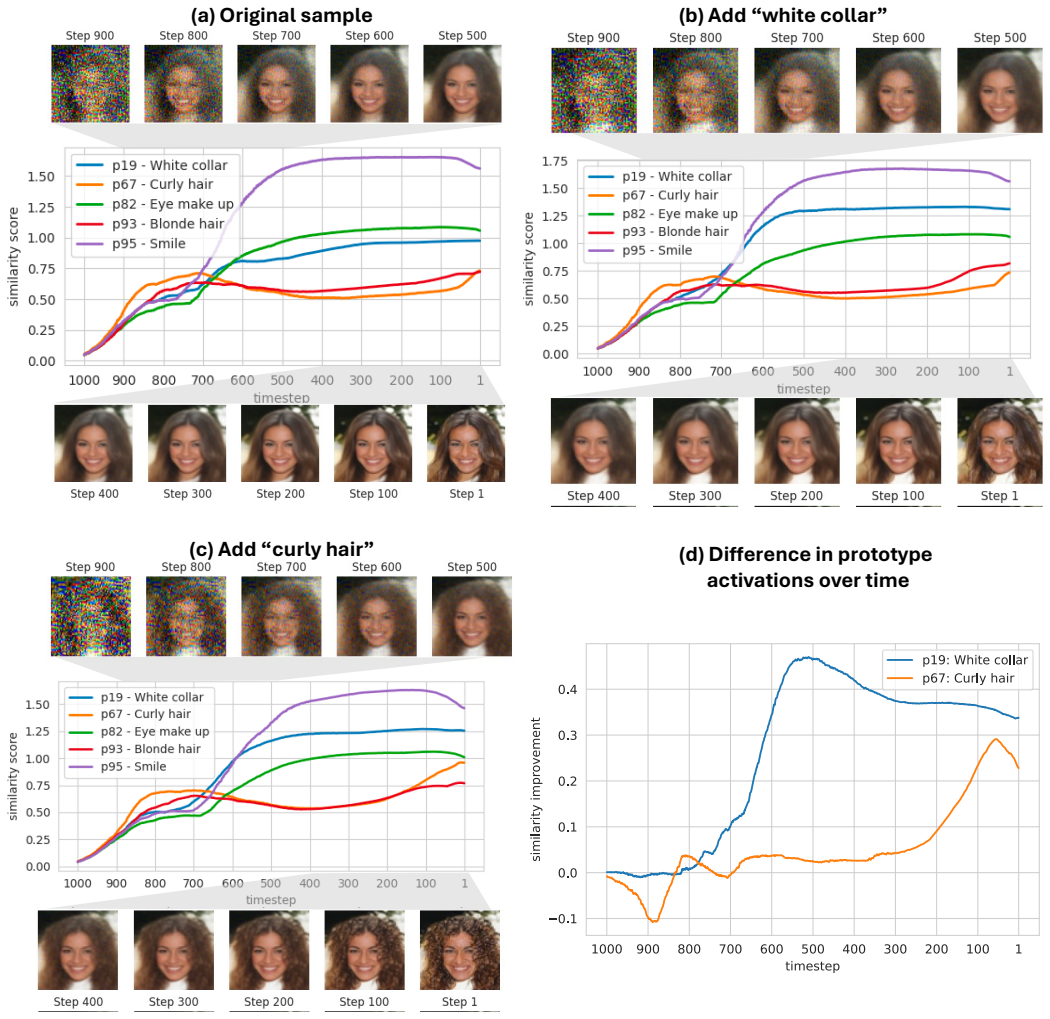


Figure 6: **Prototype emergence details in:** (a) **the original generated sample;** (b) generated sample with **”white collar”** added; (c) generated sample with **”curly hair”** added; (d) **differences in prototype activations over time**. Sharp increases indicate the timesteps when prototypes begin to emerge prominently in the enhanced sample compared to the original.

is summarized in Fig.6d, where we show the difference in prototype activation vectors between the enhanced and original samples. The sharp increases in the plotted curves indicate the timesteps at which the respective prototypes begin to emerge prominently.

Trend analysis on a larger sample set. To analyze the trend more comprehensively, we randomly select 100 samples from the test set and enhance all 100 prototypes for each sample. We then compute the average difference between the enhanced activation scores and their original values, visualizing the results as an averaged curve, just as shown in Fig. 4b.

A.3 Hyper-parameters for Training

More hyper-parameter for Patronus training could be found in Table 3. We trained all experiments with Adam optimizer, learning rate of $1e^{-4}$.

For the latent diffusion model, we trained with an 1D UNet, with base channels of 64, and channel multipliers as (1,2,4). We set the dropout rate as 0.2 and learning rate as $1e^{-4}$.

Table 3: **Hyperparameters used for *Patronus* training.**

	Input size	Num. p	shape of p	patch size	Num. patches	Num. Channels	Num. Channel. Mult
FashionMNIST	(1,32,32)	30	(1,1,128)	14×14	100	64	1,2,4,4
Cifar10	(3,32,32)	100	(1,1,128)	14×14	100	64	1,2,4,4
FFHQ	(3,64,64)	100	(1,1,128)	14×14	672	64	1,2,4,4
CelebA	(3,64,64)	100	(1,1,128)	14×14	672	64	1,2,4,4
CheXpert	(1,224,224)	100	(1,1,128)	60×60	1849	64	1,2,4,4

Table 4: **Ablation study on CelebA**, regarding the number of prototypes and the prototype vector size.

# p	**shape of p**	TAD ↑	Attrs ↑	Latent AUROC↑	FID ↓
32	(1,1,128)	$\mathit{0.8291 \pm 0.0146}$	9.0000 ± 0.0000	0.8288 ± 0.0022	6.0126 ± 0.0740
64		$\mathbf{0.8515 \pm 0.0545}$	9.2000 ± 0.4000	0.8527 ± 0.0017	6.5395 ± 0.0676
100		0.5395 ± 0.1205	$\mathit{12.0000 \pm 1.0954}$	$\mathit{0.8646 \pm 0.0010}$	5.4871 ± 0.0151
128		0.4700 ± 0.0758	$\mathbf{12.0000 \pm 0.8944}$	$\mathbf{0.8713 \pm 0.0018}$	$\mathbf{5.1264 \pm 0.0488}$
64	(1,1,64)	0.5860 ± 0.0535	8.4000 ± 0.4899	0.8491 ± 0.0021	$\mathit{5.2017 \pm 0.0429}$
	(1,1,128) *	$\mathbf{0.8515 \pm 0.0545}$	9.2000 ± 0.4000	0.8527 ± 0.0017	6.5395 ± 0.0676
	(1,1,256)	0.4779 ± 0.0045	8.0000 ± 0.0000	0.8492 ± 0.0008	6.2918 ± 0.1279

* The result is the same as the second row since the hyper-parameters are identical. We listed it as another row for easier comparison.

For prototype encoder, in this work we apply a 4-layer convolutional network with ReLU activations. The channel progression is $1 \to 32 \to 64 \to 64 \to 128$. All convolutional layers use a 3×3 kernel, with strides of $[2,\ 1,\ 1,\ 1]$ for the four layers and paddings of $[1,\ 0,\ 0,\ 0]$, respectively.

A.4 CheXpert Dataset

In this work, we use a subset of the CheXpert dataset (Irvin et al., 2019), retaining only frontal chest X-ray scans. To mitigate potential information leakage and reduce memorization effects due to patient-specific variations, we sample a single scan per patient[2]. This preprocessing step yields a total of 28,878 chest X-rays, of which 90% are allocated for training and the remaining 10% for testing.

B Ablation Study

As shown in Tab. 4, we present an ablation study of *Patronus* with respect to the number of prototypes and the dimensionality of the prototype vectors. Experiments are conducted on the CelebA dataset using an input resolution of $(3, 64, 64)$, a training duration of 200 epochs, and a learning rate of $1\mathrm{e}{-4}$. Note that FID is computed in the context of conditional generation.

B.1 Number of Prototypes

We evaluate the impact of varying the number of prototypes, setting $\#p = \{32, 64, 100, 128\}$, while fixing the prototype vector size to $(1, 1, 128)$. As the number of prototypes increases, we observe consistent improvements in latent quality (measured by AUROC), the number of attributes captured, and the FID score. However, the TAD score—which reflects the disentanglement quality—tends to decline. This trade-off is expected: while a larger prototype pool allows the model to capture more fine-grained visual patterns, it also introduces redundancy, reducing the distinctiveness and interpretability of individual prototypes. This suggests the existence of an optimal prototype budget for balancing generation quality and disentanglement.

[2]The number of recordings per patient in CheXpert is highly imbalanced, ranging from 1 to 89 (Weng et al., 2023). Notably, disease severity is correlated with scan frequency—fewer than 25% of control subjects have more than five scans, while this proportion exceeds 50% among patients.

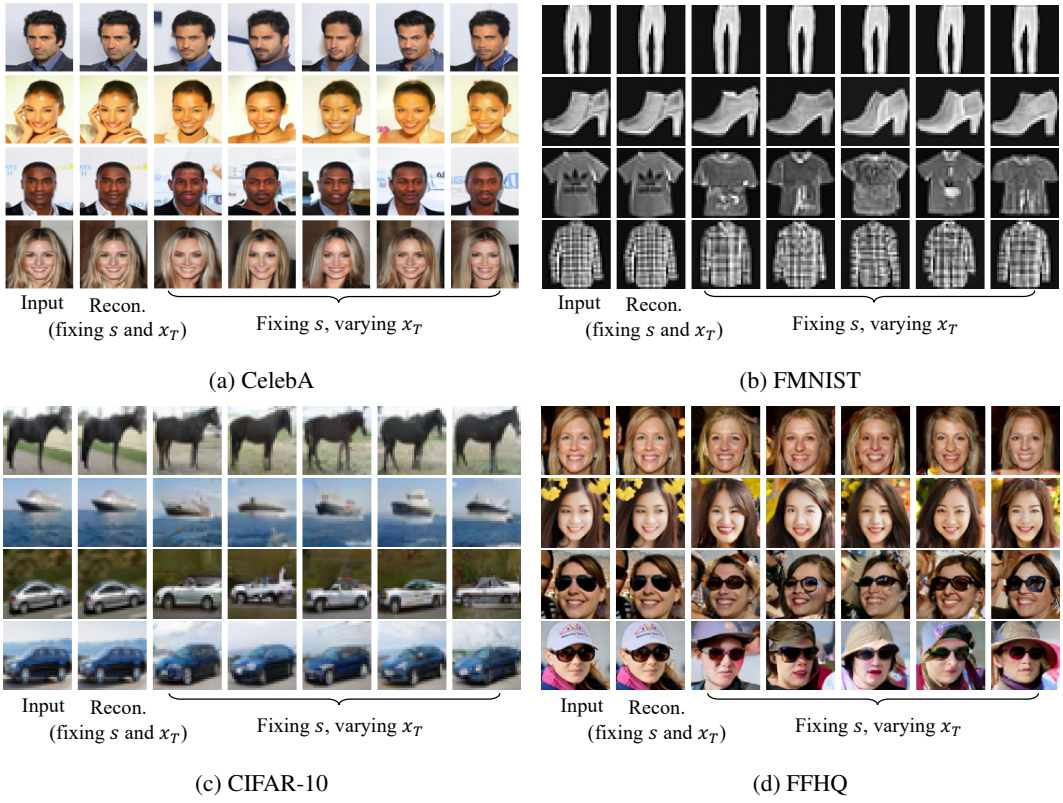


Figure 7: **Additional examples of reconstruction and variations with fixed s and random x_T.**

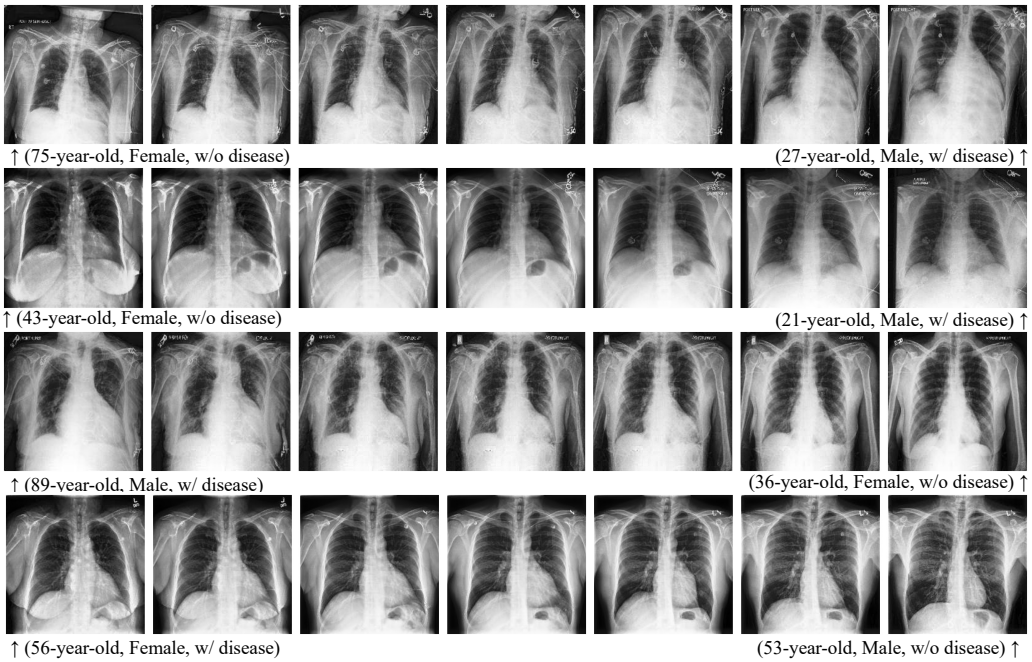


Figure 8: **Additional Image Interpolation Examples on CheXpert.** Interpolations are shown between two real chest X-rays, with patient age, gender, and disease condition (Cardiomegaly).

B.2 PROTOTYPE VECTOR SIZE

A larger prototype vector size allows each prototype to encode more detailed semantic information within a fixed spatial region. Conversely, when the prototype vector size is too small, the model may struggle to capture sufficient semantic richness. However, excessively large prototype vectors may introduce optimization challenges, such as slower convergence and increased redundancy. This trade-off is reflected in the results shown in Tab. 4, where a prototype vector size of $(1, 1, 128)$ yields the best overall performance.

C ADDITIONAL EXPERIMENTAL RESULTS

C.1 ADDITIONAL VISUAL EXAMPLES

Reconstruction and Variation with Random x_T We provide additional visual examples across multiple datasets in Fig. 7. These results demonstrate *Patronus*' ability to capture semantic features across different datasets.

Notably, while *Patronus* effectively preserves fine details and patterns, it struggles with rare patterns. For instance, in the third row of Fig.7b, the model fails to reconstruct the Adidas logo accurately. Another interesting case appears in the last row of Fig.7d, where *Patronus* successfully identifies the presence of a hat but generates variations of different hat styles instead of an exact reproduction.

Interpolation We provide additional visual examples of interpolation across various datasets, including CheXpert (Fig.8), as well as FMNIST, CIFAR-10, CelebA, and FFHQ (Fig.9a). In datasets with well-defined class clusters, such as Fashion-MNIST and CIFAR-10, interpolation tends to be less effective when transitioning between images belonging to different classes (Fig. 9b).

More Visual Samples for Diagnosis Ability of Patronus In Fig.10, we present additional samples and hair-color-related prototypes for the diagnosis task, revealing a more pronounced bias—enhancing hair color also affects the presence of a smile. More specifically, we showcase eight cases, including four female and four male subjects. Within each gender group, we include two individuals with black hair and no smile, alongside one individual with brown or blonde hair and a smile. When enhancing black hair-related prototypes, all images transition to a non-smiling expression (as seen in the fourth and sixth columns of Fig.10).

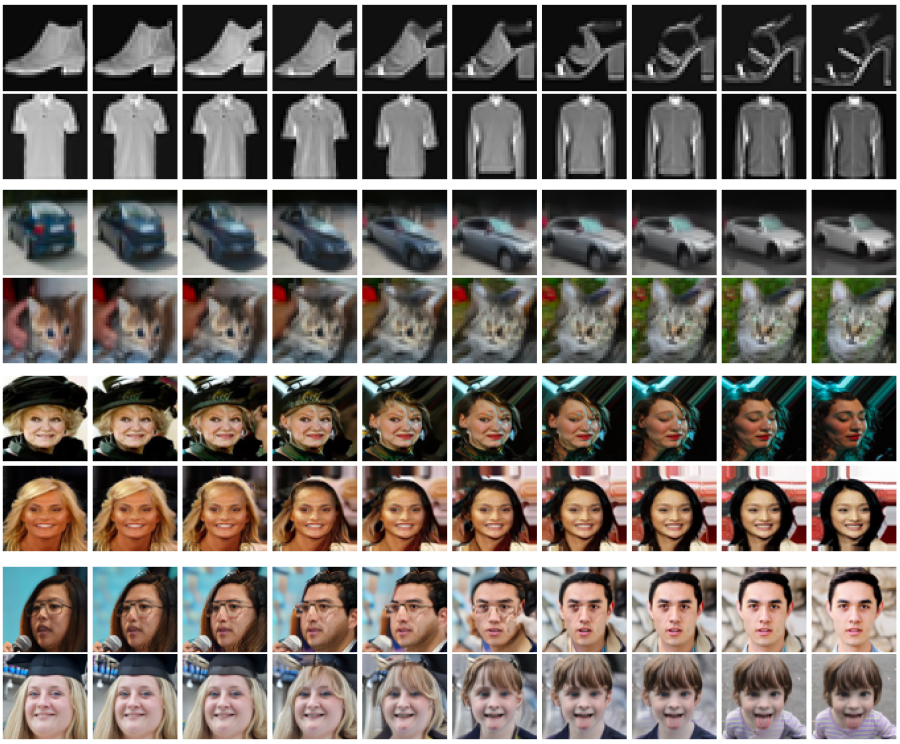
C.2 VISUAL REPRESENTATION OF PROTOTYPES

We present a complete visual representation of the learned prototypes from the CelebA dataset in Fig. 11 and 12. Below each prototype, we provide a summary of its semantic meaning based on human observation without explicit annotation. Consequently, these interpretations may contain inaccuracies. For prototypes where a clear semantic meaning could not be determined, we leave the description blank. Notably, these blank descriptions highlight the inherent limitations of language in capturing visual concepts. The visualization process follows the steps outlined in Sec. 3.3.

C.3 CAPTURED ATTRIBUTES BY A SINGLE PROTOTYPE

As shown in Tab. 1, we applied TAD and the number of attributes captured to estimate the prototype disentanglement ability, where *Patronus* remarkably outperformed the SOTA by nine captured attributes, while prior methods capture at most three. Tab. 5 details which prototypes capture these attributes, with visualizations in Fig. 13.

As shown in Tab. 5, a single prototype can capture multiple attributes. E.g. prototype 82 captures both "Eyeglasses" and "Rosy_Cheeks". This finding is particularly interesting, as prior visual inspection suggested that prototype 82 represents the concept of "Heavy Eye Make-up" (Fig. 3), which is semantically related to both attributes: heavy eye make-up often co-occurs with rosy cheeks and tends to be negatively correlated with eyeglasses, possibly because individuals wearing heavy makeup are more likely to use contact lenses instead. This observation is further supported by the intervention results shown in Fig. 13, where suppressing the activation of p_{82} leads to the appearance of eyeglasses, while enhancing the activation induces both rosy cheeks and prominent eye make-up.



(a) Within-dataset, same-class.

(b) Across classes.

Figure 9: **Additional interpolation examples.** (a) Interpolation across datasets: Fashion-MNIST (first two rows), CIFAR-10 (third and fourth), CelebA (fifth and sixth), FFHQ (seventh and eighth) (b) Cross-class interpolation (Fashion-MNIST, CIFAR-10) showing incoherent transitions.

Table 5: **Captured Attributes in CelebA.**

Captured Attributes	Captured Attributes AUROC	Captured Prototype Index
Bald	0.8276 ± 0.0041	30
Bangs	0.8411 ± 0.0022	9
Black_Hair	0.8104 ± 0.0034	78
Blond_Hair	0.8917 ± 0.0022	93
Blurry	0.8717 ± 0.0058	35
Eyeglasses	0.7967 ± 0.0026	82
Pale_Skin	0.8549 ± 0.0044	58
Rosy_Cheeks	0.8226 ± 0.0040	82
Wearing_Hat	0.9002 ± 0.0025	9

C.4 Latent Quality for CheXpert Dataset

We analyze the latent quality for CheXpert dataset by measuring TAD, number of attributes being captured and latent AUROC in Tab. 6. A total of 23 attributes are evaluated, comprising four demographic attributes, four indicators related to patients' socioeconomic or health status, one shortcut

Figure 10: **More visual samples and more hair-color related prototypes for diagnosing unwanted correlations with *Patronus***. By selecting the top two hair color prototypes (highest AUROC for single-dimension prediction), their correlation with "smile" prototype reveals dataset bias. $p_j(a/b)$ define as: (a) Spearman correlation with the "smile" prototype (green: positive, red: negative), and (b) AUROC for predicting hair color when using this prototype (**bold** if ≥ 0.75).

feature pacemaker (annotated by Weng et al. (2024)), and 14 disease-related labels. All attributes are binarized as detailed below[3]: Age ($\geq$60 or $<$60), Sex (Male or Female), Race (White or Non-white), Ethnicity (Hispanic/Latino or Other), Insurance (Enrolled in Medicare or Not), Interpreter Need (Yes or No), Deceased (Yes or No), and BMI (within the normal range: 18.5–25.0, or outside).

Evaluating latent quality. As shown in Tab. 6, the learned latent representations demonstrate strong predictive capabilities for most demographic attributes, such as age, sex and BMI, achieving high AUROC scores using a simple logistic regression model. Notably, the latent space also encodes information relevant to the presence of a pacemaker. Furthermore, it supports the prediction of several cardiopulmonary conditions—such as Cardiomegaly, Edema, and Pleural Effusion—with AUROC values exceeding 0.75 (bold font in the table). These results indicate that the latent representations capture semantically meaningful and clinically relevant information. For comparison, we include the performance of a ResNet-50 baseline in the Tab. 6. It is worth noting that this baseline was trained on 320×320 resolution images (Bressem et al., 2020), and that the original CheXpert validation set includes only samples from five disease categories. As a result, performance metrics for the remaining categories are unavailable.

Interpreting Captured Attributes and Their Corresponding Prototypes. Among the 23 attributes, three are captured by a single prototype, as listed in Tab. 6, with visualizations provided in Fig. 14. Notably, the **Sex** attribute is captured by a prototype that focuses on the edge of the chest wall (Fig. 14a). In the extrapolation experiment on prototype p_1 (Fig. 14b), we observe a decrease in rib cage size as the prototype similarity score increases. This observation aligns with clinical find-

[3]We acknowledge that the binarization is not ideal, as it may be white-centralized and introduce bias; however, it is adopted here for the sake of simplification.

Figure 11: **Visualization of learned prototypes on CelebA (prototypes 0 - 49).** Each row shows 10 prototypes with three views per prototype (top to bottom): original image, image enhanced with prototype j, and most activated patch (serves as the prototype visualization). *BG* means background.

ings that females tend to have a disproportionately smaller rib cage compared to males (Bellemare et al., 2003; 2001). For the **Age** attribute, the corresponding prototype is most activated in the region around the upper thoracic vertebrae (Fig. 14a). This may relate to the age-associated ossification

Figure 12: **Visualization of learned prototypes on CelebA (prototypes 50 - 99).** Each row shows 10 prototypes with three views per prototype (top to bottom): original image, image enhanced with prototype j, and most activated patch (serves as the prototype visualization). BG means background.

of the costochondral cartilage of the first rib (McCormick & Stewart, 1988; Radiology Masterclass, n.d.). Regarding the **Pacemaker** attribute, the most activated patch is located near the upper region of the heart, potentially reflecting the correlation between pacemaker presence and underlying car-

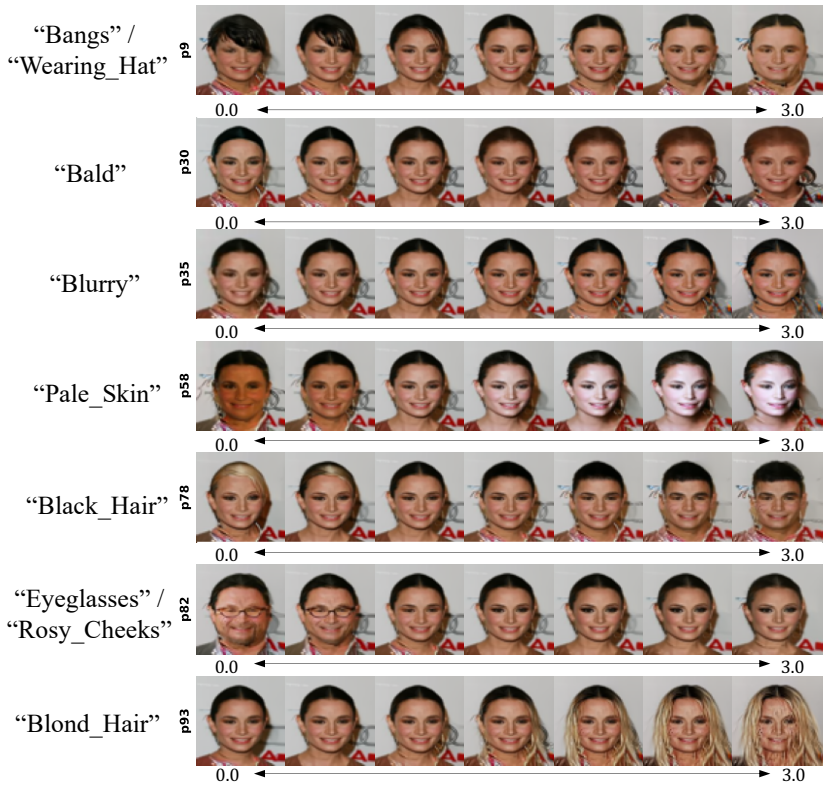
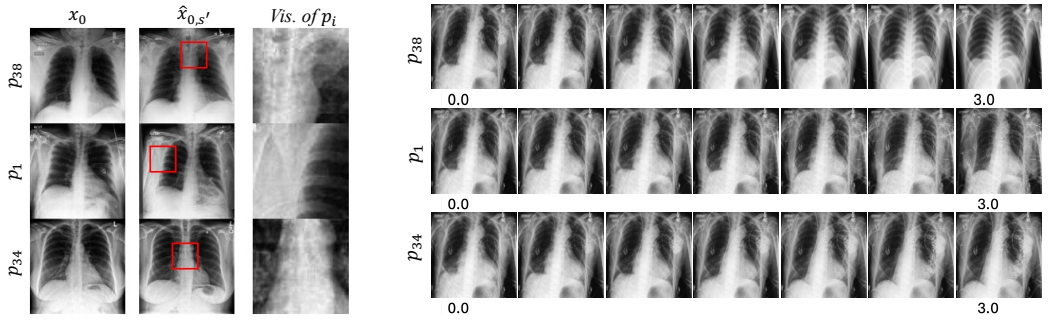


Figure 13: **Visualization of Captured Attributes in CelebA via Explorations.** The text on the left indicates the attributes captured in the CelebA dataset. Note that a single prototype can potentially capture multiple attributes.



(a) Visualization of prototypes capturing attributes.

(b) Extrapolation over the same prototypes.

Figure 14: **Visualization and extrapolation of prototypes capturing attributes from CheXpert.** Three attributes are captured: Age (by p_{38}), Sex (by p_1), and presence of a pacemaker (by p_{34}).

diac conditions. In Fig. 14b, increasing the activation of prototype p_{34} leads to a more pronounced appearance of a pacemaker.

We acknowledge the limitations of our medical expertise and do not intend to draw definitive clinical conclusions from these observations. We welcome researchers with medical backgrounds to further evaluate and interpret these findings.

Table 6: **Latent Quality of the CheXpert Dataset.** The *latent AUROC* denotes the average AUROC across all considered attributes. A total of 23 attributes are evaluated, comprising four demographic attributes, four indicators related to patients' socioeconomic or health status, one shortcut feature (pacemaker), and 14 disease-related labels. All attributes are binarized. **Bolded** values indicate AUROC ≥ 0.75.

TAD↑	Attrs ↑	Latent AUROC↑	**Demographic Attributes***				**Other Attributes***				**Shortcut**
			Age	Sex	Race	Ethnicity	Insurance	Interpreter	Deceased	BMI	PM †
0.12±0.01	3.0±0.0	0.74±0.01	**0.88±0.01**	**0.98±0.00**	0.70±0.01	0.72±0.01	0.74±0.00	0.74±0.00	0.66±0.01	**0.77±0.01**	**0.92±0.01**

	Disease Labels													
	No Finding	Enlarged CM‡	Cardio-megaly	Lung Opacity	Lung Lesion	Edema	Consoli-dation	Pneu-monia	Atelec-tasis	Pneumo-thorax	Pleural Effusion	Pleural Other	Fracture	Support Devices
LR using p	**0.86**	0.61	**0.75**	0.70	0.65	**0.77**	0.64	0.61	0.60	0.68	**0.80**	0.70	0.68	**0.76**
ResNet50	-	-	0.80	-	-	0.88	0.90	-	0.80	-	0.91	-	-	-

Captured Attributes	**Captured Attributes AUROC**	**Captured Prototype Index**
Age Group	0.7891 ± 0.0031	38
Sex	0.8806 ± 0.0005	1
PM†	0.7823 ± 0.0028	34

* AUROC Performance from Latent Representations. The reported AUROC values reflect the performance of a logistic regression classifier trained on the latent representations using 5-fold cross-validation. Demographic and other attributes are binarized, details are in text.

† Pacemaker, annotations from Weng et al. (2024).

‡ Enlarged Cardiomediastinum.

C.5 Prototype Consistency

Prototype visualization consistency for one run. We quantitatively evaluated the consistency of prototype visualizations in one run. As the visual concepts are very small patches, consistency is measured directly in pixel space, which faithfully captures their low-level structural differences.

Using Euclidean (L2) distance between visualization pairs, we observe a clear separation: within-prototype = 2.31 vs. between-prototype = 3.95. A two-sample t-test confirms this separation is highly significant (t = 19.55, p = 1.6×10^{-37}). This confirms that each prototype forms a coherent and distinguishable visual concept.

Learned prototype consistency cross different run. This part evaluates the consistency of learned prototypes across different random initializations. We'd like to emphasis that our goal is not to enforce identical explanations across models, but to explain each model's behaviour as it is trained. If two models use different internal concepts during generation, then different prototypes are expected and even desirable. Nevertheless, under almost identical training conditions (same architecture, dataset, and objective), it is still meaningful to assess whether the learned prototypes remain consistent.

To do so, we compare prototype activation patterns on the same input batches rather than directly comparing prototype vectors, since the latter are not aligned across runs due to arbitrary rotations in the feature space. For each batch of B images (here $B = 512$), we obtain prototype activations from both models:

$$A^1, A^2 \in \mathbb{R}^{N \times B},$$

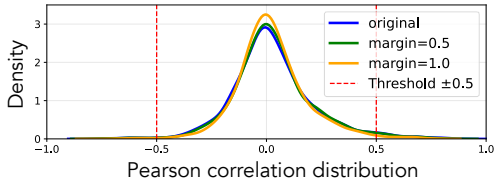
where each row corresponds to the activation pattern of one prototype across the batch.

We then compute the pairwise cosine similarity matrix:

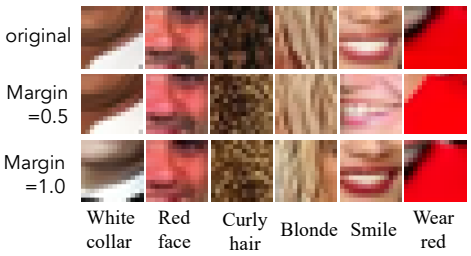
$$S_{i,j} = \frac{\langle A_i^1, A_j^2 \rangle}{\|A_i^1\| \cdot \|A_j^2\|}.$$

To resolve the permutation ambiguity between prototype indices, we determine the optimal one-to-one alignment using the Hungarian algorithm :

$$\pi = \arg \max_{\text{1-1 mapping}} \sum_i S_{i,\pi(i)}.$$



(a) **Distribution of Pearson correlation values for prototype similarity scores**, computed from 1,024 randomly selected training samples. Results compare the original training and training with an additional loss term. The difference is minimal, with only a small fraction of pairs exceeding an absolute correlation of 0.5 (red dashed lines) for all cases.



(b) Prototype visualizations and their semantic interpretations remain consistent regardless of the additional loss.

Figure 15: **Effect of additional loss on prototype diversity.** (Top) Pearson correlation distribution of prototype similarity scores shows minimal differences regardless of additional loss. (Bottom) Prototype visualizations remain consistent, indicating negligible impact on semantic representations.

We then assess how consistently this alignment appears across all test batches. The resulting permutation consistency score is 0.882, indicating that the prototype-to-prototype mapping across seeds is largely stable, with most prototypes consistently aligned between runs. After applying the optimal alignment, the matched prototypes achieve extremely high similarity (mean ≈ 0.99), indicating that the semantic behaviour of prototypes is almost identical across seeds.

D Extended Discussion

D.1 Prototype Correlation and Collapse

As discussed in Sec. 5, we further provide additional experimental results in Fig. 15. These results confirm that the new models do not show substantial changes in the learned prototypes, suggesting that prototypes optimized via the denoising objective are already sufficiently decorrelated without explicit regularization.

D.2 Limitations of Cross-Modal Interpretability

We emphasize interpreting diffusion models *without* cross-modality by design for two key reasons:

Language's limitation in representing complexity: While language, as a discursive symbolism, serves as a powerful medium for interpretation, it alone cannot fully represent non-symbolic sensory and semantics complexity (Langer, 2009). This is evident in the superior performance of multi-modal learning over single-modality; prior work (Gal et al., 2022; Goyal et al., 2017) further shows the restricted perceptual capacity of language-only generation by introducing visual cues.

Inherited bias from the text embedding: (1) Incomplete text representations: if certain concepts are not explicitly named (e.g., medical devices in radiology reports), the model cannot learn their visual counterparts. (2) Spurious correlations: text data may encode unintended biases, such as differing report detail levels by patient demographics, which may propagate into generated images.

D.3 Do We Capture All Relevant Attributes?

We further illustrate the challenge of capturing global features with sample results in Fig. 16. Following the experiment described in Sec. 4.1, we reconstruct images and their variations using a fixed s with either fixed or random x_T. While fine-grained details, such as cheekbone structure in the second row and shirt details in the first row, are well preserved, the generated images with varying x_T fail to capture age or gender consistently on harder scenarios. For instance, the middle image in the first row appears noticeably younger than the original, while most variations in the second row

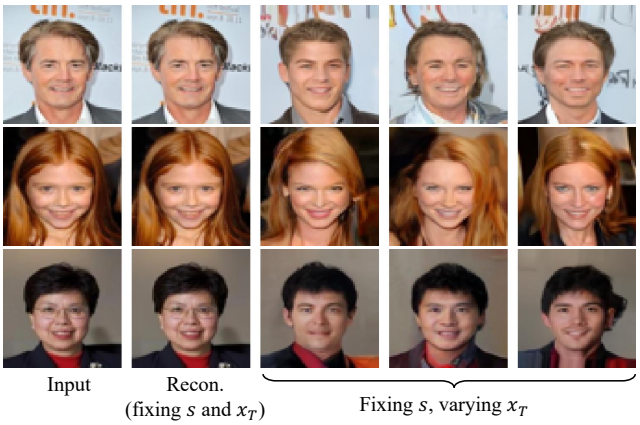


Figure 16: ***Patronus* may struggle to capture global features such as age and gender**. To illustrate this, we present three samples generated using the fixed s with random x_T. While the generated images successfully preserve semantic attributes such as hair color, cheekbones, smile, shirt, and background color, they fail to accurately reconstruct age or gender.

depict a more mature appearance. Additionally, in the third sample, a gender shift from female to male is observed, possibly influenced by the presence of short hair.

D.4 Justification for Excluding ProtoPNet's Additional Loss Terms

In this work, we optimize the model using only the denoiser loss. A key question arises: *Is this loss sufficient?* Besides the attempt in Sec. 5 (Fig. 15) to add an extra disentanglement loss, we further compare with ProtoPNet, which, alongside cross-entropy loss for classification, introduces two additional loss terms: **(1) Cluster loss (Clst)**: encourages each training sample to have at least one patch close to a prototype; **(2) Separation loss (Sep)**: pushes latent patches away from prototypes of other classes.

Neither loss applies in our setting: (i) prototype representation is supposed to capture the underlying data distribution, *not* to enforce proximity to specific training samples; (ii) our method is generative and does not rely on class labels, making class-dependent separation constraints irrelevant. Thus, the denoiser loss is sufficient, as ProtoPNet's additional terms do not align with the objectives.

E The Use of LLMs in this Work

In this work, we used large language models (LLMs) solely to assist with the presentation of the paper, including grammar correction, wording refinement, and minor sentence shortening (at the level of one or two words, not entire paragraphs). LLMs were not used for any other purpose. We always check the content after using LLMs, and we are responsible for the content that we submit.

F Quantitative evaluation over interpretability

F.1 Evaluating Prototype–Language Alignment via Image Captioning

An additional experiment was conducted to examine whether language-based analysis can support the alignment between visual concepts and human understanding. Specifically, we aim to assess whether our prototypes encode visual features that are both visible and interpretable through human concepts. To this end, we used the BLIP model (Salesforce/blip-image-captioning-large on HuggingFace) to caption the prototype-enhanced image for prototype j and the corresponding original image, and then compared the words with the most significant increases. For example, for the prototype corresponding to 'curly hair' in the visualization (Fig. 3a), the top three words with the highest increases after applying the prototype are 'curly': 409, 'hair': 307, 'long': 66, where each value

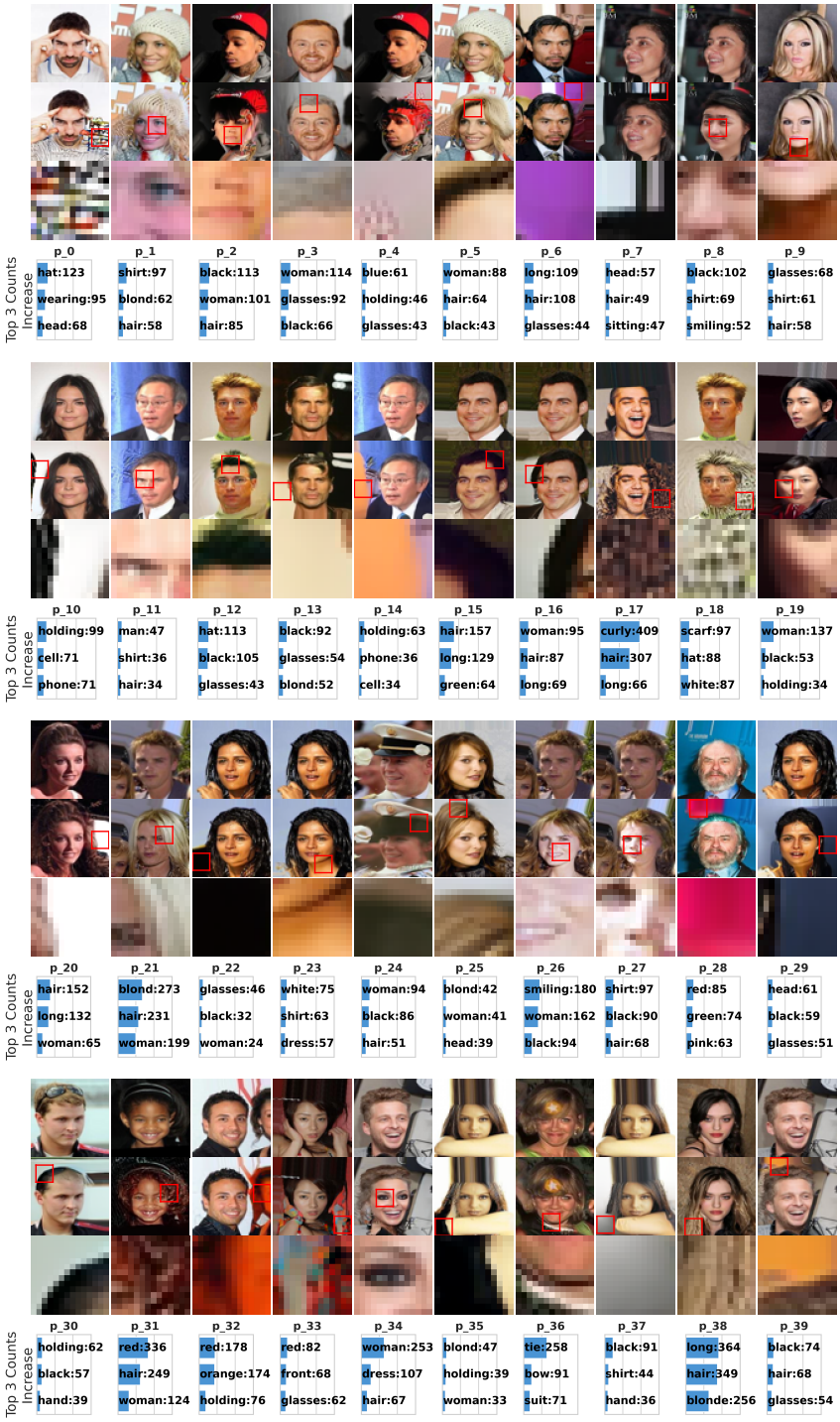


Figure 17: **Visualization of learned prototypes on CelebA with the top three increased words using BLIP for captioning (p0-p39).** Each row shows 10 prototypes with three views per prototype (top to bottom): original image, image enhanced with prototype j, and most activated patch (serves as the prototype visualization), and the top three increased words.

reflects the difference in word frequency between the enhanced and original images across 672 samples. We provide the full list of the top three increased words, along with prototype visualizations from one run, in Fig. 17-18. Note that this evaluation depends strongly on the captioning model's

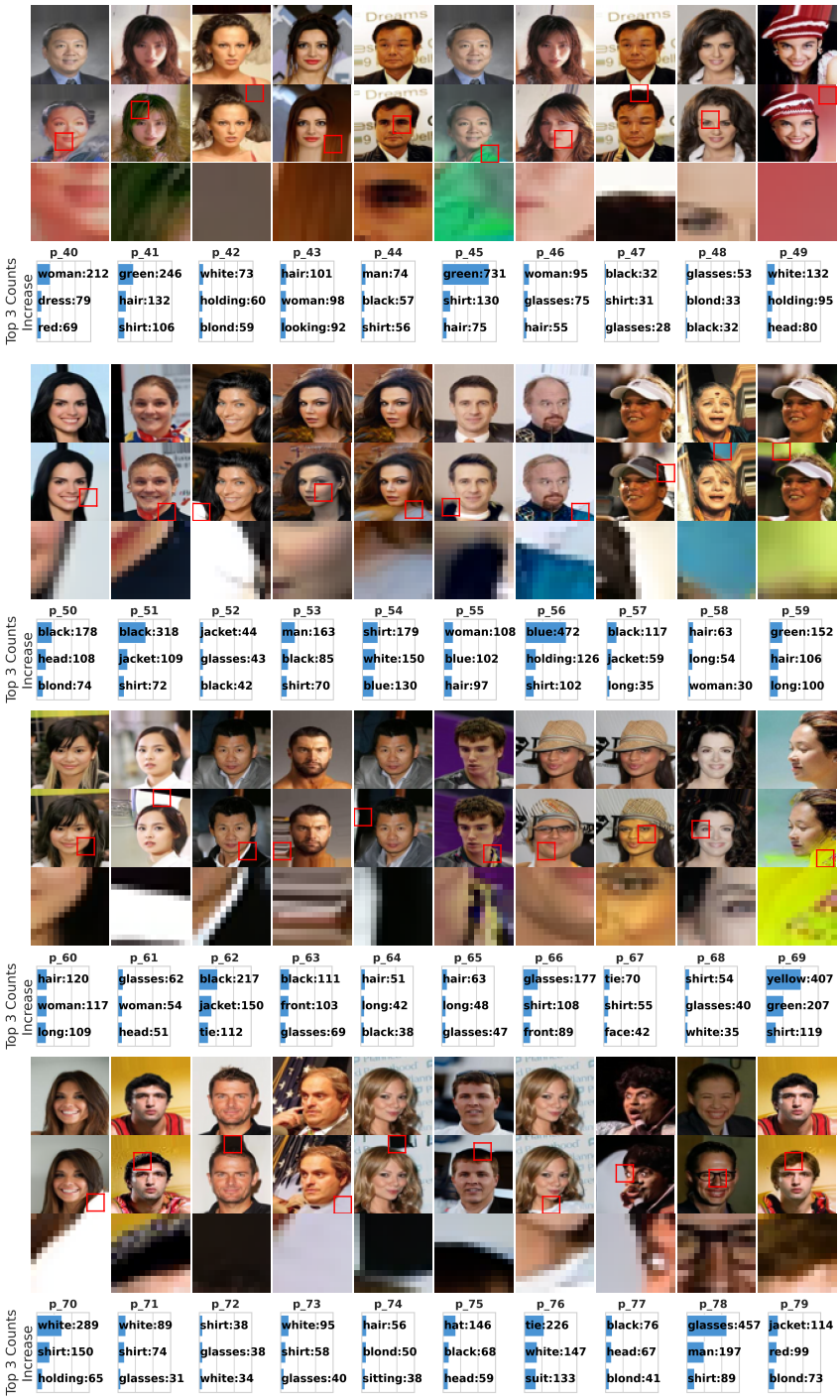


Figure 18: **Visualization of learned prototypes on CelebA with the top three increased words using BLIP for captioning (p40-p79).** Each row shows 10 prototypes with three views per prototype (top to bottom): original image, image enhanced with prototype j, and most activated patch (serves as the prototype visualization), and the top three increased words.

vocabulary. For instance, prototype 34 in this run corresponds to a heavy eye-makeup enhancement, but the model fails to describe it and instead produces only gender-related terms.

Figure 19: **Visualization of learned prototypes on CelebA with the top three increased words using BLIP for captioning (p80-p99).** Each row shows 10 prototypes with three views per prototype (top to bottom): original image, image enhanced with prototype j, and most activated patch (serves as the prototype visualization), and the top three increased words.

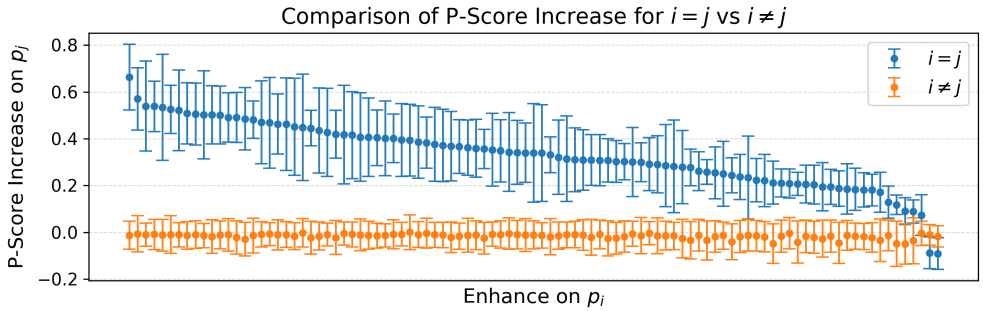


Figure 20: **Faithfulness evaluation of prototype scores.** Enhancing prototype p_i leads to a clear increase in its own score ($i = j$), while scores of other prototypes remain near zero ($i \neq j$), indicating that the similarity measure behaves as intended.

F.2 Faithfulness measurement

To evaluate the faithfulness of the prototype scores, we measure whether the score of prototype p_i increases when the corresponding condition is enhanced. For each prototype p_i, we generate an enhanced image using p_i (x-axis) and compute the change in all prototype scores (y-axis). The results are shown in Fig. 20. The target prototype shows a clear increase (mean around 0.35), whereas the remaining prototypes remain near zero, indicating that the similarity score functions as expected. The two observed outliers correspond to prototypes that encode minimal semantic information based on their visualizations.

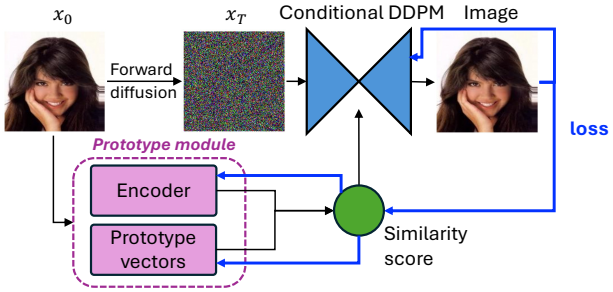


Figure 21: **Training Overview.** The black arrows show the forward computation: the encoder extracts prototypes and obtains similarity scores, which condition the DDPM during generation. The blue arrows indicate the backward path of the denoising loss, which updates both the prototype module and the conditional DDPM jointly.

G Training Strategy

The training strategy for Patronus is direct and simple. All components are jointly trained with solo training objectives, i.e. the denoiser loss. Fig. 21 shows the training overview with both forward computation and backward loss propagation in black and blue arrow respectively.

H Robustness of prototype activation control

We evaluate the robustness of prototype activation control by comparing the cosine similarity between clean images and images generated with noise-perturbed prototype activations. We add random noise with magnitudes [0.1, 0.5, 1.0, 2.0, 5.0, 10.0]% of the maximum activation value (fixed to 2.0 in our experiments). mportantly, we perturb all prototype activations, not just a single prototype. Image similarity is measured using the cosine similarity of InceptionV3 embeddings.

Fig. 22 shows the results, clearly illustrating that the prototype activation score function is highly robust, that the cosine similarity decreases slightly at low noise levels. When the noise magnitude increase to 10%, the embedding similarity becomes comparable to that between random clean image pairs. Visual examples of generated images with noise added to the activation scores are provided in Fig. 23.

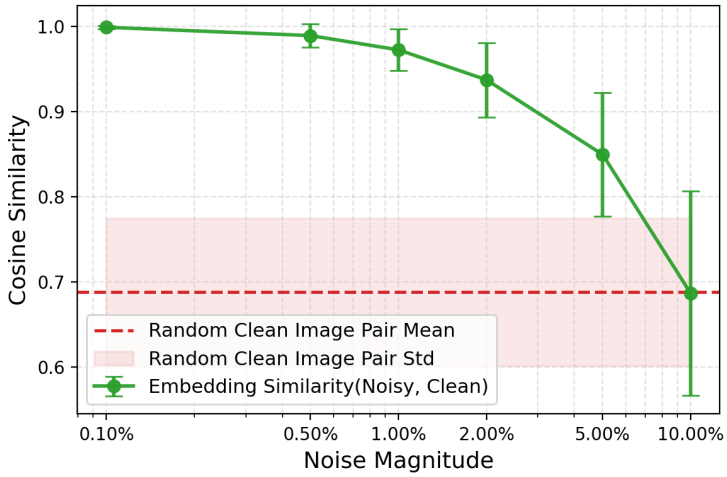


Figure 22: **Prototype activation control robustness** quantified by comparing the embedding similarity between clean images and images generated with added activation noise.

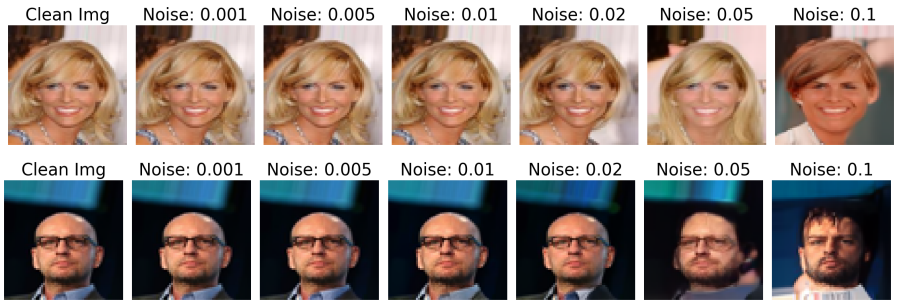


Figure 23: **Visual examples of generated images after noise is added to the activation scores.**