Implementation of VisCoIN for the visualisation of concepts activated in the classification of medical images

Indira Fabre *indira.fabre@telecom-paris.fr*
Telecom Paris, 19 Place Marguerite Perey, 91120 Palaiseau

Abstract

The purpose of this work is to implement the VisCoIN model, a method that relies on a generative model (StyleGAN2) and a classifier (ResNet50), developed for visualizing the concepts activated during image classification. The first objective is to assess the reproducibility of the published results on the CUB-200 dataset. The second objective is to evaluate the transferability of the VisCoIN model to medical images, specifically using the NCT-CRC-HE dataset, and to assess its relevance as a tool for understanding the decision-making process of deep learning models in the context of medical image analysis. We were able to reproduce the published results to a certain extent. However, the obtained FID score was significantly worse than the one reported in the original paper, leading to a struggling reconstruction, and we provide possible explanations for this discrepancy. Transferring the VisCoIN model to medical images proved to be more difficult. While we obtained some results indicating that the methodology is relevant, the overall architecture likely needs to be adaped to better suit the characteristics of medical images in order to produce more usable outcomes.

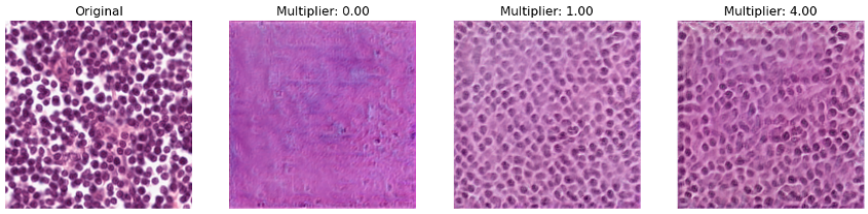


Figure 1: Example obtained with VisCoIN trained on the NCT-CRC-HE dataset, with 16 concepts, showing the activation of the best concept "nuclei" in class 3 (LYMPHOCYTES)

1 Context and related work

In the context of developing interpretable models for prediction, networks that rely on learning high-level concepts are particularly valuable, as these concepts are aligned with human cognitive representations. However, understanding what these concepts are and understanding their visual characteristics remains a significant challenge The VisCoIN method (Parekh et al., 2025) addresses this by mapping concept features to the latent space of a pretrained generative model. This enables high-quality visualisations and supports an intuitive and interactive procedure for interpreting the learnt concepts.

While artificial intelligence (AI) plays an increasingly important role in medical image analysis tasks such as detection, classification, diagnosis, segmentation, prediction, and image quality enhancement, it remains a highly specialized and high-stakes domain. It poses unique challenges, including limited access to end users and annotated data, knowledge gap between domain experts and machine learning designers, and the need for high accuracy and reliability in predictions (Chen et al., 2022).

As a consequence, without clear explanations of how and why a particular decision or output is reached, medical professionals (such as physicians or radiographers), may hesitate to fully rely on the results gener-

ated by machine learning algorithms. The integration of AI into routine clinical practice continues to face obstacles, largely due to this lack of trust. Therefore, enhancing the interpretability and transparency of deep learning models is essential for their effective adoption in healthcare settings. In recent years, there has been a growing body of research on the application of explainable AI techniques to medical imaging tasks (Champendal et al. (2023)).

Explainable AI methods on medical imaging involve post-hoc and by design (or self-explainable methods). While post-hoc methods focus on providing explanations for a black-box model after it has been trained, self-explainable AI methods are designed to be interpretable by nature (Hou et al., 2024).

These methods incorporate explainability as an integral part of the model during the training process and can be based on input, architecture or output. *Input explainability* focuses on integrating additional explainable inputs with deep features of medical images obtained from various anatomical locations and modalities to produce final predictions. By incorporating external knowledge and context-specific information, the accuracy and reliability of these predictions can be significantly improved. *Model explainability* aims to design inherently intepretable model architectures of deep neural networks to transform the model into an interpretable format. *Output explainability* refers to the model's ability to generate not just predictions for various medical image tasks but also accompanying the result with an explanation generator. This capability aids in understanding the rationale behind the model's predictions, facilitating informed medical decision-making.

As an example of *model explainability* method, Chen & Krishnan (2022) reported a self-supervised method for tissue phenotyping based on the learning of characterizations of histopathologic biomarkers. They show that vision transformers using DINO-based (Caron et al., 2021) knowledge distillation are able to learn interpretable features in histology images and show that different attention heads learn distinct morphological phenotypes.

The VisCoiN methodology, by Parekh et al. (2025), is an example of *output explainability*. It appears to be a promising candidate for application to medical imaging datasets. It falls into the category of interpretable predictive models that are built on top of pretrained models (rather than training the complete model from scratch). It provides visualizations of learned concepts, which may be more informative than simple attention maps, especially when relevant information is distributed across an image. It focuses on accuracy and fidelity, which are key in this sensitive and high stakes domain. It also has the advantage to be applicable even to small datasets, which are common in medical contexts due to data privacy, annotation constraints, and overall availability of data, in particular in the case of rare diseases.

After reproducing the results of the VisCoIN methodology on a real-world images dataset, we describe the transferability of the method to a medical images dataset in the following sections.

2 Architecture of the VisCoIN model

The design of VisCoIN is illustrated in Figure 2. It is based on the Concept-based Interpretable Network (CoIN) architecture (Alvarez-Melis & Jaakkola (2018), Koh et al. (2020)) but relies on a fixed pretrained classifier f used with a fixed pretrained generative model G. These two models are not modified during the training of VisCoIN.

The objective of this architecture is to explain the predictions of the classifier f by learning a dictionary of K concepts and mapping this dictionary to the latent space of the generative model G to visualize the concepts.

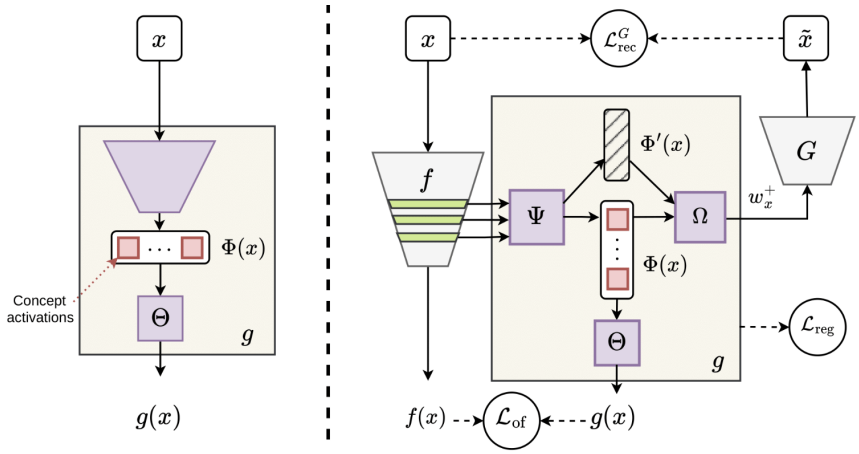


Figure 2: **Left:** Standard CoIN system, that makes prediction from extracted concepts. **Right:** VisCoIN system, leveraging a pretrained generative model G for visualization and a pretrained classifier f - figure taken from (Parekh et al., 2025).

The explainability of the model f is obtained through $g(x)$, which is a concept-based prediction model, such as $g(x) = \Theta(\Phi(x))$. It computes the final prediction using concept activations $\Phi(x)$, which are the outputs of the concept extractor Ψ. During training, three subnetworks are trained: Ψ, Ω and Θ.

$\Phi(x)$ is a dictionary of concepts functions. It is modeled by using selected hidden layers of the pretrained classifier f, and taking them as input to the concept extractor Ψ, such that $\Phi(x) = \Psi(hiddenlayers)$.

A second output of the concept extractor Ψ is $\Phi'(x)$. It is an unconstrained supporting representation whose role is to assist Ω in embedding the input in the latent space of the generative model G.

Ω is responsible for the viewability of the explanations of the model, as its output is the input to the generative model G that will produce the visualizations of the concepts.

This gives $\tilde{x} = G(\Omega(\Phi(x), \Phi'(x)))$. Each single fully connected (FC) layer of Ω takes either $\Phi(x)$ or $\Phi'(x)$ as input, and outputs a latent vector in the latent space of G. In the case of 256×256 output resolution (which is the case in this work), there are 14 latent vectors in the generator. According to Katzir et al. (2022), 4 vectors control coarse feature, 4 control mid-level features and 6 control fine-level features of the generated image.

Authors in the original VisCoIN paper (Parekh et al., 2025) report to use $\Phi(x)'$ to predict the first three and last two style vectors of the generator (5 vectors out of 14) and to use $\Phi(x)$ to predict the rest of the style vectors (9 vectors out of 14). The choice reported in their Git repository[1] as default parameters differ, and this will be disscussed in more detail in section 6.2.1.

Θ is designed to pool the feature maps $\Phi_1, \Phi_2, \ldots, \Phi_K$ to obtain a single concept activation of size K and make the final prediction $g(x) = \Theta(\Phi(x))$ by passing it through a linear layer followed by softmax. This simple design allows to estimate the importance of each concept function Φ_k for the final prediction.

2.1 Loss and metrics

The training losses are defined based on the desired properties of the model:

- **Fidelity to output property**: $\mathcal{L}_{\text{of}}$, the ouput fidelity loss, is included to guarantee the ability of g to classify the inputs x correctly. It is defined as a generalized cross entropy (CE) between the outputs of g and f:

$$\mathcal{L}_{\text{of}}(x; \Psi, \Theta) = \alpha \text{CE}(g(x), f(x)). \tag{1}$$

[1] https://github.com/GnRlLeclerc/VisCoIN

- **Fidelity to input, quality of reconstruction and viewability properties**: $\mathcal{L}_{\text{rec}}^{G}$, the reconstruction loss, combines these properties by applying constraints between inputs x and their reconstruction $\tilde{x} = G(\Omega(\Phi(x), \Phi'(x)))$. ℓ_1 and ℓ_2 penalties ensure pixel-wise reconstruction for *fidelity to input*. The perceptual similarity between the input x and the reconstruction $\tilde{x}$ is measured using Learned Perceptual Image Patch Similarity (LPIPS, Zhang et al. (2018)), and linked to *viewability*. LPIPS computes the similarity between the activations of two images patches for a predefined network. This measure has been shown to match human perception well. A low LPIPS score means that the two image patches are perceptually similar. A final *reconstruction classification* term is defined as $\text{CE}(f(\tilde{x}), f(x))$. It encourages the generative model to reconstruct $\tilde{x}$ with more classification-specific features pertaining to input x.

  The reconstruction loss is the sum of these four terms, weighted by hyperparameters β and γ. It is computed through the pretrained generative model G and the pretrained classifier f:

$$\mathcal{L}_{\text{rec}}^{G}(x; \Psi, \Omega) = \|\tilde{x} - x\|_2^2 + \|\tilde{x} - x\|_1 + \beta \text{LPIPS}(\tilde{x}, x) + \gamma \text{CE}(f(\tilde{x}), f(x)). \tag{2}$$

- **Sparsity property**: This property is enforced through the regularization of the concept extractor Ψ and the concept translator Ω, with the term $\mathcal{L}_{\text{reg}}$. Ψ is regularized to encourage both sparsity and diversity. Sparsity is induced through an ℓ_1 penalty on the activations $\Phi(x)$, diversity is promoted through a kernel orthogonality loss $\mathcal{L}_{\text{orth}}$ applied on the weights of the final convolution layer of Ψ. This allows to reduce redundancy between the concepts in the learnt dictionary Φ. Ω is regularized to encourage the predicted latent vectors $w_x = \Omega(\Phi(x))$ to be close to an average latent vector $\bar{w}$. The global regularization loss $\mathcal{L}_{\text{reg}}$ is the sum these three terms, with δ as a hyperparameter to weigh the sparsity term.

$$\mathcal{L}_{\text{reg}}(x; \Psi, \Omega) = \mathcal{L}_{\text{reg}-\Psi}(x; \Psi) + \mathcal{L}_{\text{reg}-\Omega}(x; \Omega), \tag{3}$$

$$\mathcal{L}_{\text{reg}-\Omega}(x; \Omega) = \|w_x - \bar{w}\|_2^2, \quad \mathcal{L}_{\text{reg}-\Psi}(x; \Psi) = \delta \|\Phi(x)\|_1 + \mathcal{L}_{\text{orth}}(\Psi). \tag{4}$$

The resulting training loss and optimization are:

$$\mathcal{L}_{\text{train}}(x; \Psi, \Theta, \Omega) = \mathcal{L}_{\text{of}}(x; \Psi, \Theta) + \mathcal{L}_{\text{rec}}^{G}(x; \Psi, \Omega) + \mathcal{L}_{\text{reg}}(x; \Psi, \Omega), \tag{5}$$

$$\hat{\Psi}, \hat{\Theta}, \hat{\Omega} = \arg\min_{\Psi, \Theta, \Omega} \frac{1}{N} \sum_{x \in \mathcal{S}} \mathcal{L}_{\text{train}}(x; \Psi, \Theta, \Omega). \tag{6}$$

The role of the four hyperparameters α, β, γ and δ, whose effect is extensively discussed in the original paper (Parekh et al., 2025), can be interpreted as follows:

- α controls the importance of the output fidelity loss. A small weight will degrade the performance of the system, while a large weight was found to affect reconstruction.
- β controls the importance of the perceptual similarity between the input and the reconstruction. Values higher than 3 are reported to lead to instability during training. Low values are detrimental to the perceptual similarity.
- γ controls the importance of the reconstruction classification term. Variations of γ lead to a trade-off between faithfulness and perceptual similarity. A smaller value can be used when the reconstruction is more challenging. Higher values can push the model to generate spurious feaures captured by the classifier at the expense of input quality.
- δ controls the importance of the sparsity term. A high δ will impact both the performance and the reconstruction by encouraging activation of a smaller number of concepts. A low δ might result in poor sparsity and thus poor interpretability of the concepts activation.

3 Datasets description

3.1 Reproducibility on the CUB-200 dataset

We first reproduced the results on the CUB-200 dataset (Wah et al., 2011), published in the original VisCoIN paper. The CUB-200 dataset is a widely used benchmark for fine-grained visual classification tasks. The 2011 version is a collection of 11,788 images of 200 bird species. The dataset also contains for each image a set of 312 binary attributes (such as "has crest", "has red belly", "has yellow belly"), bounding boxes, and part locations. The binary attributes can be useful to compare the concepts extracted with VisCoIN to actual attributes of each bird species.

The images are of various size (usually around 500×350 pixels) and were resized to a fixed resolution of 256×256 pixels for the experiments, by using a random crop of 256×256 pixels from the original image. An extract of the dataset is shown in Figure 3. The dataset is further split into a training set of 5994 images and a test set of 5794 images.

Figure 3: Extract of the CUB-200 dataset after resizing to 256×256 pixels.

3.2 Application to a medical images dataset

We then applied the VisCoIN method to the NCT-CRC-HE-100K dataset (Kather et al., 2018). It is a set of 100,000 non-overlapping image patches from hematoxylin and eosin (H&E) stained histological images of human colorectal cancer (CRC) and normal tissue. All images are 224×224 pixels (px) at 0.5 microns per pixel (MPP) and are color-normalized using Macenko's method. The dataset consists if 9 tissue classes: Adipose (ADI), background (BACK), debris (DEB), lymphocytes (LYM), mucus (MUC), smooth muscle (MUS), normal colon mucosa (NORM), cancer-associated stroma (STR), colorectal adenocarcinoma epithelium (TUM) (see extract in Figure 4). These images were manually extracted from $N = 86$ H&E stained human cancer tissue slides from formalin-fixed paraffin-embedded (FFPE) samples from the NCT Biobank (National Center for Tumor Diseases, Heidelberg, Germany) and the UMM pathology archive (University Medical Center Mannheim, Mannheim, Germany). Tissue samples contained CRC primary tumor slides and tumor tissue from CRC liver metastases; normal tissue classes were augmented with non-tumorous regions from gastrectomy specimen to increase variability.

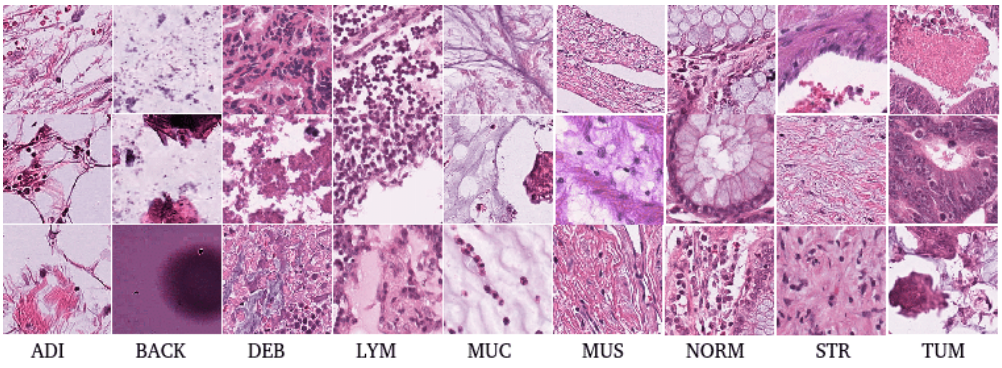


Figure 4: Extract of the NCT-CRC-HE-100K dataset (Hatami et al., 2021)

For testing, the CRC-VAL-HE-7K dataset was used. This is a set of 7180 image patches from 50 patients with colorectal adenocarcinoma (with no overlap with patients in NCT-CRC-HE-100K). Like in the larger data set, images are 224 × 224 px at 0.5 MPP.

The images of the dataset were resized to 256 × 256 pixels using LANCZOS interpolation. Further normalisation is performed during the training of the GAN and of the classifier.

4 Implementation setup

We trained all models on a single Tesla P100-16GB GPU.

5 Pretraining of the classifier f and the generative model G

5.1 Pretraining f

f classifiers were trained on each dataset before using them for VisCoIN training. We used Adam optimizer with a learning rate of 10^{-4} and a weight decay of 10^{-4} for both NCT-CRC-HE and CUB-200. The training is initialized in both cases with pretrained weights from ImageNet then fine-tuned for 30 epochs on CUB-200 and only for 10 epochs on NCT-CRC-HE (the dataset being much larger, convergence was achieved faster). Because of the imbalance between classes in the NCR-CRC-HE dataset, we chose to use a weighted cross entropy loss, with weights inversely proportional to the number of samples in each class. To the best of our knowledge, authors of the original VisCoIN paper (Parekh et al., 2025) do not mention rebalancing classes for the CUB-200 dataset. For comparison purposes and because of only slight imbalance between classes, we chose to not apply any rebalancing for the CUB-200 dataset either.

Training parameters and results are summarized in Table 1. We can see that good accuracy is achived, with results similar to those reported on the CUB-200 dataset.

	CUB-200		**NCT-CRC-HE**
# epochs	30		10
Batch size	32		32
	Parekh et al. (2025)	**This work**	**This work**
Learning rate (initial)	10^{-4}	10^{-4}	10^{-4}
Weight decay	-	10^{-4}	10^{-4}
Test accuracy (in %) ↑	80.56	**82.50**	93.19

Table 1: Results from classifier training on CUB-200 and NCT-CRC-HE datasets. Training parameters and results from the reference paper are given for comparison.

5.2 Pretraining G

We used pretrained StyleGAN2-ADA models finetuned on our target datasets to obtain the generator G for our experiments, following the methodology described in the original VisCoIN paper (Parekh et al., 2025), which is adapted from the StyleGAN2-ADA paper (Karras et al., 2020). We use the code available from the official repository of the StyleGAN2-ADA implementation[2]. We started from available checkpoints from NVIDIA[3], and finetuned them to a reasonable extent with the published default parameters.

The choice of the checkpoint to finetune from was first based on the idea to choose a model trained on a dataset of a close domain. For this reason, regarding the NCT-CRC-HE dataset, a first attempt was made using a StyleGAN2-ADA pretrained on the BreCaHAD dataset. It is a dataset of microscopic biopsy images with histopathological annotation and diagnosis for 162 breast cancers. Because this model was trained on 512 $\times$ 512 images, our images were resized to this resolution to be able to use this pretrained model. The training was overall slow (20h for 200K images) and the quality of the generator remained limited (best FID: 31.47). As a second attempt, we followed the methodology described in (Karras et al., 2020) for transfer learning, and used FFHQ-140k with matching 256 $\times$ 256 resolution as a starting point. The results were better and similar to results reported for BreCaHAD in the original paper (best FID: 14.15 and KID: 2.79×10^3). This had the additional advantage, compared to the first approach, to divide training time by four and allowed for an easier transferability to the VisCoIN methodology, based on 256 $\times$ 256 resolution.

Regarding the CUB-200 dataset, we chose to start from the same FFHQ-140k checkpoint. The checkpoint used in Parekh et al. (2025) was pretrained on ImageNet, but the authors did not specify the source. The obtained FID was 12.68, versus a reported FID of 9.4 in Parekh et al. (2025).

The training results are described in Table 2 and compared to reference papers. We reported both FID and KID scores. Fréchet Inception Distance (FID) is a metric used to asses the quality of images created by a generative model. It compares the distribution of generated images to the distribution of a set of real images by calculating mean and covariance statistics of many images generated by the model are comparing them to the same statistics from images of the reference set. When estimating the Fréchet distance, it is assumed that the embeddings for each image set (real and generated), come from a multivariate normal distribution. As specified in Karras et al. (2020), FID is not an ideal metric for small datasets, since it becomes dominated by the inherent bias when the number of real images is unsufficient. Kernel Inception distance (KID, Bińkowski et al. (2021)) is unbiased by design and more suitable. Both measures show better performance when they have lower values.

	CUB-200		**BreCaHAD**	**NCT-CRC-HE**		
	(Parekh et al., 2025)	**This work**	(Karras et al., 2020)	**This work**		
starting checkpoint	ImageNet (unspecified)	FFHQ-140k	From scratch / FFHQ-140k	BreCaHAD	FFHQ-140k	FFHQ-140k
# real training images	2M	1.5M	25M	220K	500K	1.5M
training time	21h	40h	-	20h	11h	40h
FID ↓	around **9.4**	12.68	15.71 / -	31.47	16.85	**14.15**
KID ($\times 10^3$) ↓	-	5.55	2.88 / 3.36	-	3.75	**2.79**

Table 2: GAN training results on CUB-200 and NCT-CRC-HE datasets. Available measures from reference papers are given for comparison when available.

Images generated from the obtained pretrained StyleGAN2-ADA model for the NCT-CRC-HE dataset are shown in Figure 5. Images generated from the obtained pretrained StyleGAN2-ADA model for the CUB-200 dataset are available in appendix A.1 along with the FID evolution during training, for both models.

[2]https://github.com/NVlabs/stylegan2-ada-pytorch
[3]https://nvlabs-fi-cdn.nvidia.com/stylegan2-ada-pytorch/pretrained/

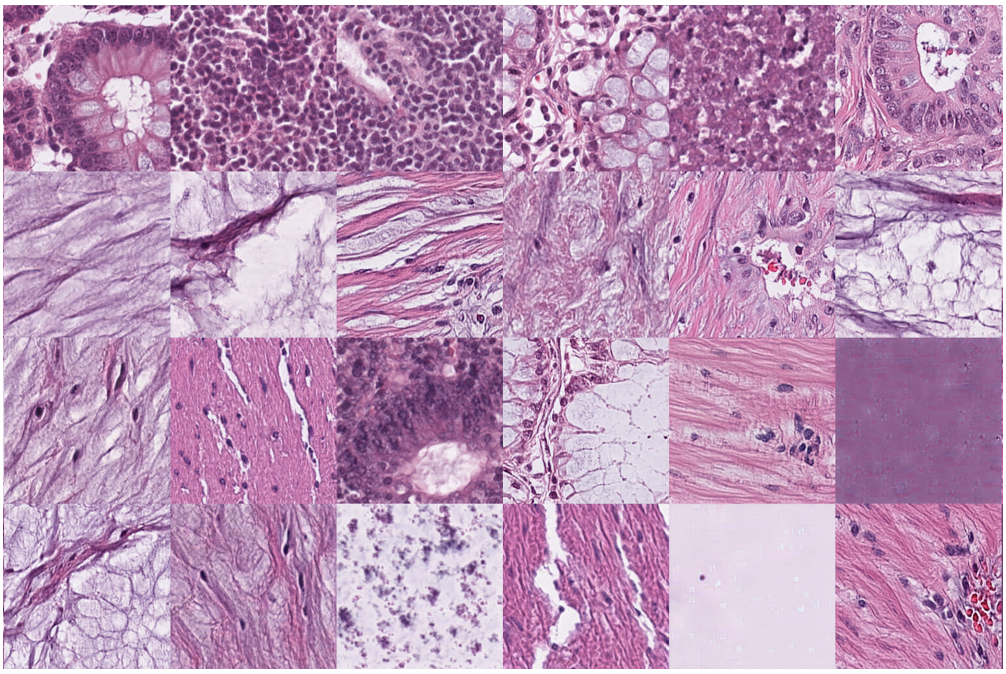


Figure 5: Sample images generated by the obtained pretrained StyleGAN2-ADA model for the NCT-CRC-HE dataset.

6 VisCoIN training

6.1 Reproducibility on the CUB-200 dataset

We trained VisCoIN on the CUB-200 and NCT-CRC-HE datasets using the obtained pretrained classifier f and the pretrained generative model G.

We used the parameters reported in the original VisCoIN paper (Parekh et al., 2025) for the CUB-200 dataset, and implemented a code adapted from the official repository[4]. The model was trained for 100K iterations, with a learning rate of 0.0001. During training, each batch consisted of 8 samples from training data and 8 synthetic samples randomly generated using G. We fixed $\alpha = 0.5$, $\beta = 3.0$, $\gamma = 0.1$ and $\delta = 0.2$. Training and testing losses and FID scores are available in A.2. We used a coarse layer value of 3 and a mid-layer value of 12, as reported in the paper.

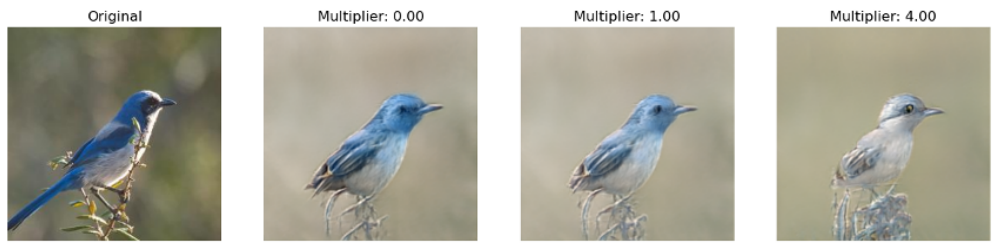
	Parekh et al. (2025)	**This work**
Original f accuracy (in %) ↑	80.56	**82.50**
VisCoIN accuracy (in %) ↑	79.44	**81.40**
FID ↓	**15.85**	64.41
LPIPS ↓	**0.545**	0.561

Table 3: Results from VisCoIN training on the CUB-200 dataset

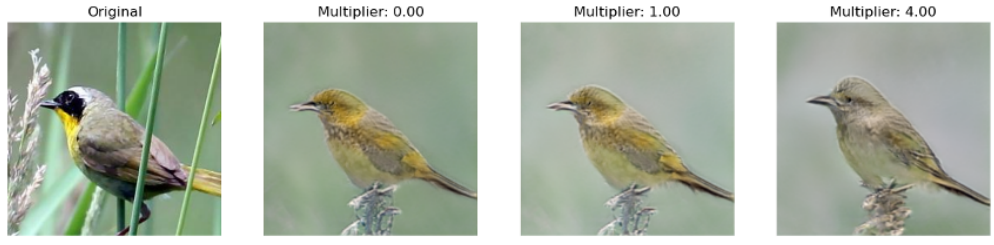
We can see from the results reported in Table 3 that we keep a good accuracy for the classification task with the VisCoIN model g, which shows a drop in accuracy, compared to the original classifier f, similar to the one reported in the original paper.

However, the obtained FID score of 64.41 is significantly worse than the one reported in the original paper, which was 15.85. Our starting GAN model was already showing poorer performance than the one used in

[4]https://github.com/GnRlLeclerc/VisCoIN



(a) Concept "white belly" in class 74



(b) Concept "brown upperparts" in class 200

Figure 6: Qualitative examples obtained with VisCoIN trained on the CUB-200 dataset. The first column shows the original image, the second column shows the reconstructed image with concepts of highest relevance ($r_{k,c} > 0.8$) deactivated, the third column shows the reconstructed image with activated concepts with the original $\Phi(x)$, fourth column shows the reconstructed images with activated concepts by imputing $4 \times \Phi(x)$ on concepts of highest relevance

the original paper (12.68 vs 9.4) but the gap does not seem to be sufficient to explain the huge difference in FID score.

Obtained images are shown in Figure 6 with activation of the best concepts. We can see that the reconstruction struggles to capture the details of the birds (additional examples are avaiblable in Figure 13 in the Appendix). Moreover, when the background is dominant on the image, the reconstruction is often strongly impacted by its presence, with no bird clearly identifiable in the reconstructed image (see example in Figure 14).

This effect could be explained by the methodology used to preprocess the dataset. We randomly cropped the images to 256×256 pixels, which in some cases led to a bird only partially present or even absent from the image. This led the pretrained G model to generate also images which are not representative of any bird species (see examples in Figure 9).

In spite of these limits, we consider that the results obtained on the CUB-200 dataset are satisfactory, as they show good qualitative examples when activating the best concepts learned by the VisCoIN model. We can see that the reconstrution quality of the images is limited but we can identify relevant features pertaining to the bird species, such as the white belly in Figure 6a or the brown upperparts in Figure 6b.

6.2 Transferability to the NCT-CRC-HE dataset

The first attempts to train VisCoIN on the NCT-CRC-HE dataset were not satisfactory. The following sections describe the reasoning and experiments that were conducted to understand the most impactful parameters and try to adapt the VisCoIN model to the NCT-CRC-HE dataset. Detailed parameters for each experiment are reported in Table 8 in the Appendix.

Note: Not enough experiments were run to do a statistical analysis of the results, but we can report that repeated experiments with same parameters show a typical FID variation of ± 2 points and LPIPS variation

of ±0.02 in VisCoIN scores. These observations are taken into account in the next section to asses the actual impact of parameters variation on the results.

6.2.1 Variation of the mapping parameters to the latent vectors of the generator

The experiments in this section are performed using a StyleGAN2-ADA model pretrained on only 500K real images.

As explained in Section 2, in the case of 256 × 256 output resolution, 14 style vectors are used in the generator G. The coarse layer is the generator layer index below which the style vectors are predicted using support representation from $\Phi'(x)$. The mid-layer is the generator layer index over which the style vectors are predicted using support representation from $\Phi'(x)$.

In the original VisCoIN paper, the authors used $\Phi'(x)$ to predict the first three and last two style vectors of the generator (5 vectors out of 14). $\Phi(x)$ is used to predict the rest of the style vectors (9 vectors out of 14). This corresponds to a coarse layer value of 3 and a mid layer value of 12.

This choice was made by the authors because relevant features for classification are expected to be controlled mostly by mid-level and fine-level style vectors. In the reported code on the original repository, default parameters are different and use a coarse layer value of 2 and a mid layer value of 10.

Our medical images differ from images on which the original VisCoIN paper was trained, which represent natural images: faces (CelebA-HQ dataset), animals (CUB-200 dataset) and cars (Stanford-Cars dataset). We have more prominent texture and shape features, whereas the images of the original paper have more distinctive small details and colors. Discussion with the authors of the VisCoIN paper confirmed that thes layer parameters could be a key element in the success of the VisCoIN model on the NCT-CRC-HE dataset.

We assessed the performance of the model for 256 concepts with different values for the coarse and mid-layers, using values reported in the original paper, in the reference code, and higher coarse level value.

The results are shown in Table 4. No obvious visual difference could be observed between the three experiments, when reconstructing images using the trained model. In all threee experiments, the accuracy of the trained model remains high. The "lower level" experiment show the best FID value, but also the worst LPIPS score. Discussion with the authors of the original paper led us to favor a better LPIPS score over a better FID score, since the LPIPS score is more relevant to assess the quality of the reconstruction. For this reason we chose to keep for the next experiments the mapping parameters of coarse layer 4 and mid-layer 12, which show intermediate results for FID and LPIPS.

	Mid-level	**Higher level**	**Lower level**
Original f accuracy (in %) ↑		93.19	
Coarse layer	3	4	2
Mid-layer	12	12	10
VisCoIN accuracy (in %) ↑	**93.63**	93.48	93.40
FID ↓	82.77	81.08	**77.18**
LPIPS ↓	0.431	**0.428**	0.440

Table 4: Results from VisCoIN training on the NCT-CRC-HE dataset, with different values of coarse and mid-layers.

6.2.2 Further training of the generator G

Since the first results obtained with the pretrained StyleGAN2-ADA model were not satisfactory, we further trained the generator G to reach 1.5M seen real images and compared the results.

As reported in Table 5, this additional training did not have a significant impact on the VisCoIN model scores. This experiment does not support the hypothesis that the bad results obtained with the pretrained

	Initial training	Further training
Original f accuracy (in %) ↑	93.19	
# training images for G	500K	1.5M
VisCoIN accuracy (in %) ↑	93.48	93.84
FID ↓	81.08	81.19
LPIPS ↓	0.428	0.433

Table 5: Comparaison of the results obtained for VisCoIN starting from a pretrained StyleGAN2-ADA model with 500K and 1.5M real images, with 256 concepts

StyleGAN2-ADA model were due to the quality of the pretrained G model. Nevertheless, as reported in the reference paper, the VisCoIN's system quality is limited by the quality of the pretrained G. For this reason, the next experiments were all made with the GAN pretrained on 1.5M real images.

6.2.3 Variation of the loss hyperparameters

We then evaluated the impact of the hyperparameters γ and δ on the performance of VisCoIN.[5] Hyperparameters α and β were set to default values, respectively 0.5 and 3.0. Coarse and mid-layers were set to 4 and 12, respectively, based on the conclusions from the previous sections. The number of concepts was set to 256, which is the value for CUB-200 and Stanford Cars in the original paper (Parekh et al., 2025). In the NCT-CRC-HE dataset, the number of classes is lower (9 classes, compared to 200 classes in CUB-200 and 196 for Stanford Cars), and we expect that the number of concepts can be lowered. Nevertheless, the first trials were made with 256 concepts, since lower values are reported to strongly affect the reconstruction, which is a key issue for our dataset.

	Default	Low γ	High γ	Low δ	High δ
Original f accuracy (in %) ↑	93.19				
γ - Weight for reconstruction-classification	0.1	0.05	0.2	0.1	0.1
δ - Weight for sparsity	0.2	0.2	0.2	0.05	2
VisCoIN accuracy (in %) ↑	93.84	93.44	93.52	93.63	92.91
FID ↓	81.19	82.55	84.33	81.76	**76.71**
LPIPS ↓	0.433	0.439	0.441	**0.430**	0.441

Table 6: Results from VisCoIN training on the NCT-CRC-HE dataset, with different values for γ and δ loss hyperparameters, with 256 concepts.

We can see from the results reporte in Table 6 that the decreasing the values of γ and δ does not significantly impact the performance of the VisCoIN model. On the opposite, we see a significant variation in FID scores when increasing γ and δ values. A higher γ value leads to a higher FID score, which means poorer performance of the model. This is consistent with the fact that smaller γ values need to be used when the reconstruction is challenging, as reported in Section 2.1. A higher δ value leads to a better FID score. This modification encourages the activation of a smaller number of concepts. This suggests that 256 concepts might indeed be too high for the NCT-CRC-HE dataset, which has only 9 classes. This hypothesis is to a certain extent confirmed by the results reported in the next section.

6.2.4 Variation of the number of concepts

A last attempt of hyperparameter tuning was made by varying the number of concepts. We compare in Table 7 the results obtained with 16, 32, 64 and 256 concepts.

[5]For $\gamma = 0.2$, the VisCoIN accuracy value is measured after 80K iterations due to a problem in saving the trained model. FID and LPIPS scores are correctly reported after 100K iterations

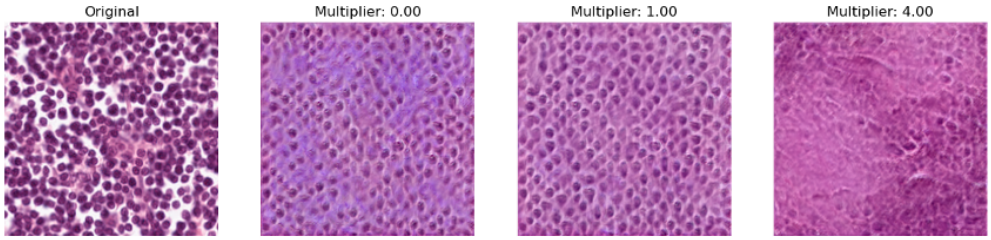
# of concepts	16	32	64	256
Original f accuracy (in %) ↑	93.19			
VisCoIN accuracy (in %) ↑	92.20	92.98	92.70	**93.84**
FID ↓	76.42	**75.31**	78.31	81.19
LPIPS ↓	0.480	0.463	0.444	**0.433**

Table 7: Results from VisCoIN training on the NCT-CRC-HE dataset, with different number of concepts

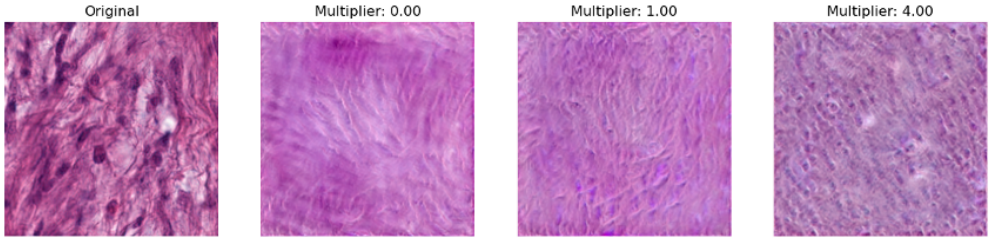
We can see that the accuracy of the VisCoIN model is not significantly impacted by the number of concepts. The FID score decreases with the number of concepts until 32 concepts, then increases again, suggesting that the number of concepts becomes too low to capture the relevant features of the images. This is consistent with the observations from the previous section and with the fact that the number of concepts can be reduced when using datasets of low diversity, as reported in the original VisCoIN paper. In our case, the NCT-CRC-HE dataset consists of only with only 9 classes, and similar-looking images (color, shapes).

The LPIPS score is evolving in the opposite direction and slowly increases when reducing the number of concepts from 256 to 16. In all cases, the visual reconstruction of the images remains of poor quality.

6.2.5 Discussion



(a) Activation of the best concept in class 3 (LYMPHOCYTES): "background"



(b) Activation of the 4 best concepts in class 7 (CANCER ASSOCIATED STROMA): "nuclei of stromal cells"

Figure 7: Qualitative examples obtained with VisCoIN trained on the NCT-CRC-HE dataset, with 64 concepts. The first column shows the original image, the second column shows the reconstructed image with best concepts deactivated, the third column shows the reconstructed image with activated best concepts with the original $\Phi(x)$, the fourth column shows the reconstructed images with activated concepts by imputing $4 \times \Phi(x)$ on the best concepts

We report in Figure 7 qualitative examples obtained with VisCoIN trained on the NCT-CRC-HE dataset (64 concepts). The only class with a reconstruction that allows to visually identify the class is class 3 (LYMPHOCYTES), characterized by dense clusters of small round nuclei (see Figure 7a). The other classes are not visually identifiable, and the reconstruction quality is poor.

The main features that we manage to isolate are the nuclei of the cells, and fiber or background structure. For example, for the class 3 (LYMPHOCYTES), when the model is trained with 64 concepts, the best concept activated is the "background" (see Figure 7a), which is not relevant for the classification task. When the model is trained with only 16 concepts, the best concept activated is the "nuclei" (see Figure 1), which is more relevant for the classification task (with 16 concepts, all other classes are very poorly reconstructed). Additional qualitative examples are reported in Figure 17 in the Appendix.

Overall we can see that the VisCoIN model applied to the NCT-CRC-HE dataset does not manage to reconstruct the images correctly but is able to isolate some relevant features, such as the nuclei of the cells. It seems that the predominance of background information, with no specific area in the image that can be isolated to explain the classification task, is the main reason for the poor results obtained with VisCoIN on this dataset.

7 Conclusions and future work

VisCoIN was originally developed for real-world images, such as faces, animals and cars. It performs well on the reported datasets, focusing on explaining the activated concepts that are localized on specific regions of the image, while effectively ignoring the background. We were able to reproduce satisfactory results on the CUB-200 dataset. However, in the medical images dataset we used, an additional challenge arises: there is no irrelevant background that the model can disregard, as the entire image is relevant for the classification task. The key elements in this context rely on texture and global patterns rather than on localized regions that can explain the classification outcomes.

Although we attempted to adjust hyperparameters and mapping layers, these modifications proved insufficient. We managed to understand that a reduced number of concepts is adapted to our use case. However, architectural changes to the model appear necessary in order to adapt VisCoIN effectively to the NCT-CRC-HE dataset.

The code for this work is available at `https://github.com/IndiraFa/XAI_medical_images_viscoin`.

Acknowledgments

We thank Telecom Paris for providing the computational resources for this work. We thank the authors of the original VisCoIN paper for their work, for making their code public, and for being available for discussions and questions.

References

David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks, 2018. URL `https://arxiv.org/abs/1806.07538`.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans, 2021. URL `https://arxiv.org/abs/1801.01401`.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. URL `https://arxiv.org/abs/2104.14294`.

Mélanie Champendal, Henning Müller, John O. Prior, and Cláudia Sá dos Reis. A scoping review of interpretability and explainability concerning artificial intelligence methods in medical imaging. *European Journal of Radiology*, 169:111159, 2023. ISSN 0720-048X. doi: https://doi.org/10.1016/j.ejrad.2023.111159. URL `https://www.sciencedirect.com/science/article/pii/S0720048X23004734`.

Haomin Chen, Catalina Gomez, Chien-Ming Huang, and Mathias Unberath. Explainable medical imaging ai needs human-centered design: guidelines and evidence from a systematic review. *npj Digital Medicine*, 5 (1):156, 2022. doi: 10.1038/s41746-022-00699-2. URL `https://doi.org/10.1038/s41746-022-00699-2`.

Richard J. Chen and Rahul G. Krishnan. Self-supervised vision transformers learn visual concepts in histopathology, 2022. URL https://arxiv.org/abs/2203.00585.

Nima Hatami, Mohsin Bilal, and Nasir Rajpoot. Deep multi-resolution dictionary learning for histopathology image analysis, 04 2021.

Junlin Hou, Sicen Liu, Yequan Bie, Hongmei Wang, Andong Tan, Luyang Luo, and Hao Chen. Self-explainable ai for medical image analysis: A survey and new outlooks, 2024. URL https://arxiv.org/abs/2410.02331.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data, 2020. URL https://arxiv.org/abs/2006.06676.

Jakob Nikolas Kather, Niels Halama, and Alexander Marx. 100,000 histological images of human colorectal cancer and healthy tissue, April 2018. URL https://doi.org/10.5281/zenodo.1214456.

Oren Katzir, Vicky Perepelook, Dani Lischinski, and Daniel Cohen-Or. Multi-level latent space structuring for generative control, 2022. URL https://arxiv.org/abs/2202.05910.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models, 2020. URL https://arxiv.org/abs/2007.04612.

Jayneel Parekh, Quentin Bouniot, Pavlo Mozharovskyi, Alasdair Newson, and Florence d'Alché Buc. Restyling unsupervised concept based interpretable networks with generative models, 2025. URL https://arxiv.org/abs/2407.01331.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

A Appendix

A.1 Pretraining G

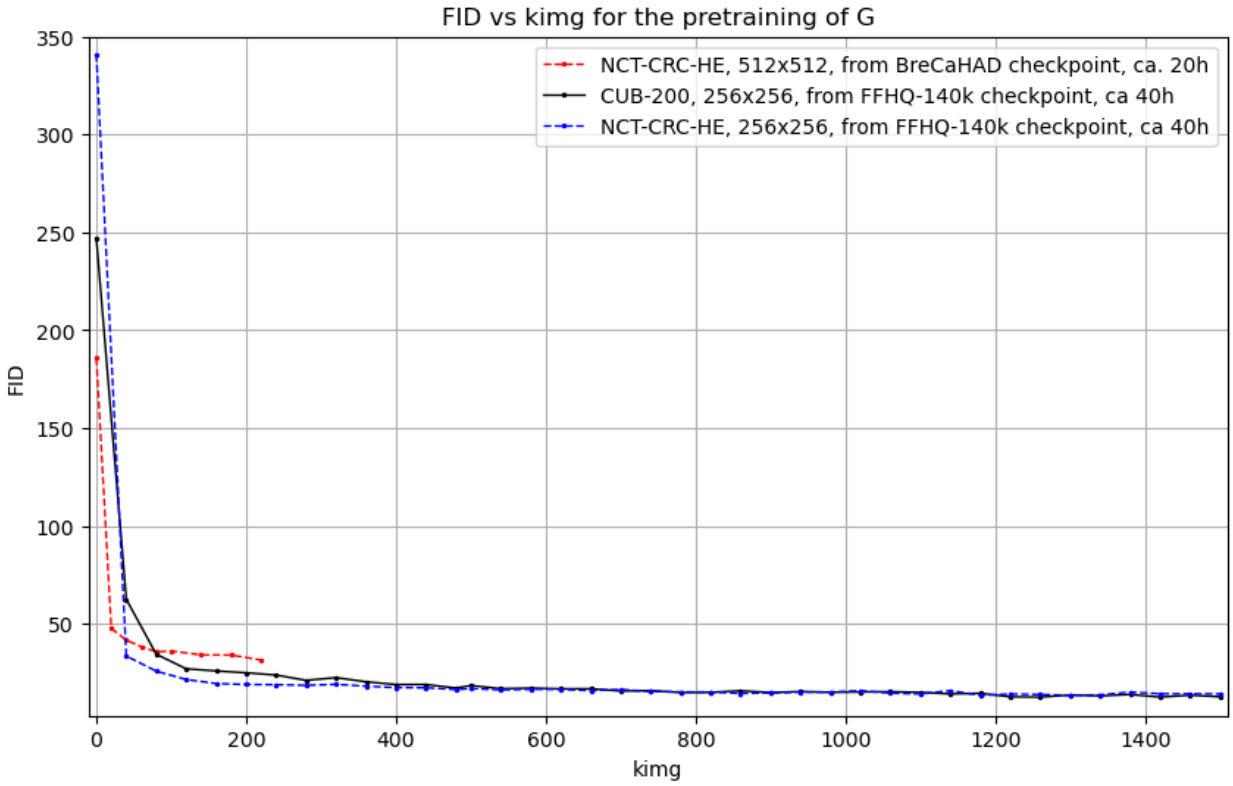


Figure 8: Evolution of the FID score during the pretraining of G

Figure 9: Sample images generated by the pretrained StyleGAN2-ADA model for the CUB dataset.

A.2 Training Viscoin on CUB-200

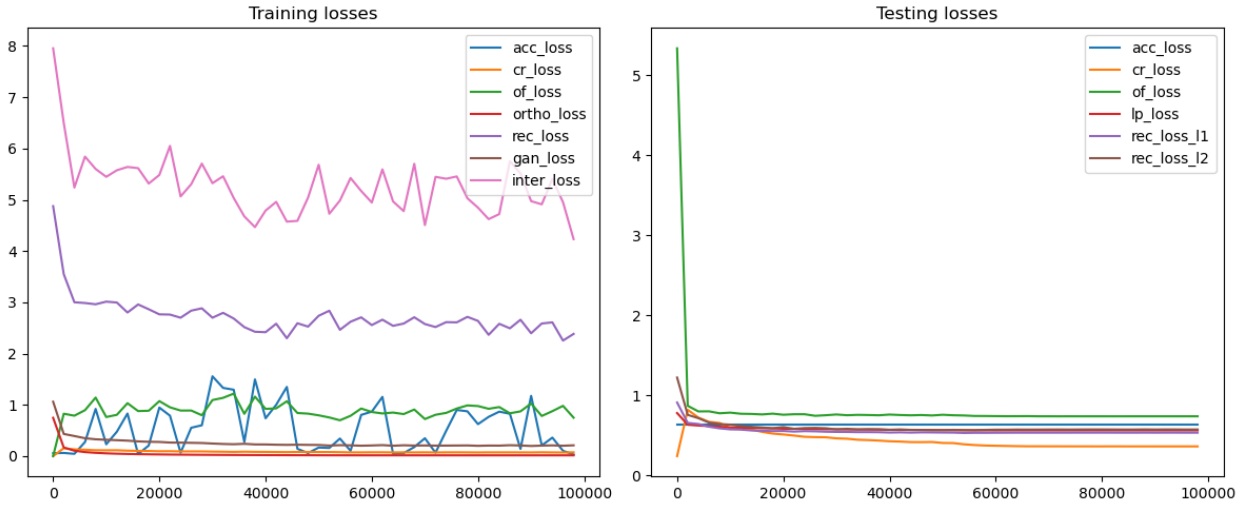


Figure 10: Evolution of the training and testing losses during the training of VisCoIN on the CUB-200 dataset.

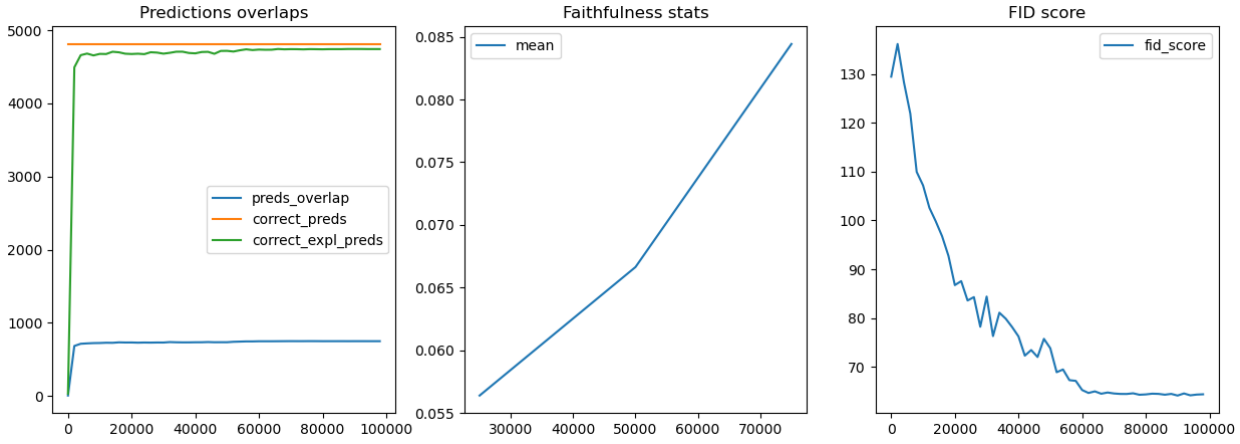


Figure 11: Evolution of predictions overlaps, faithfulness and FID during the training of VisCoIN on the CUB-200 dataset.

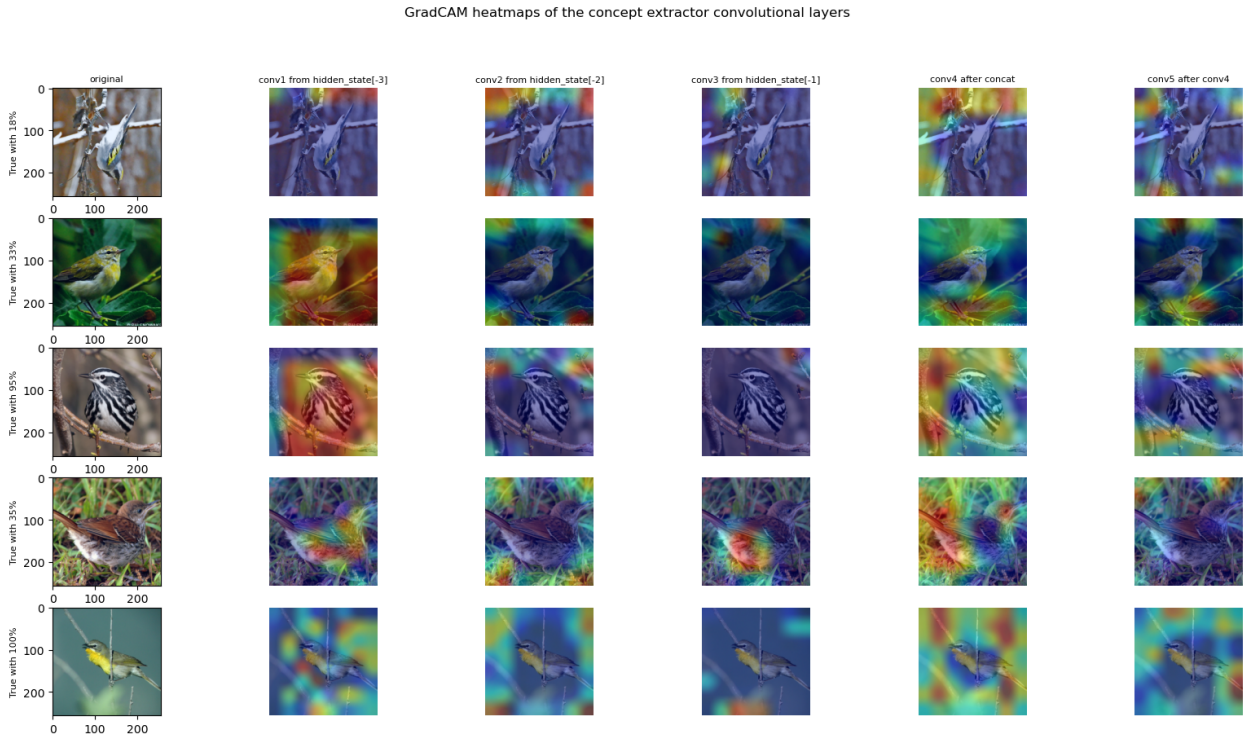
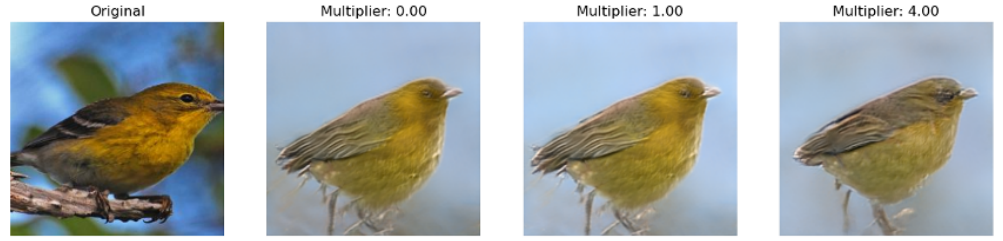
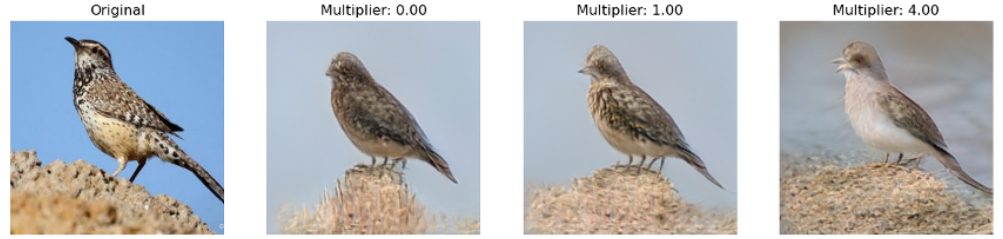


Figure 12: Grad-CAM visualizations for the CUB-200 dataset. The first column shows the original image, the other columns show the Grad-CAM results for different layers of the VisCoIN model.

(a) Concept "blue head" in class 15



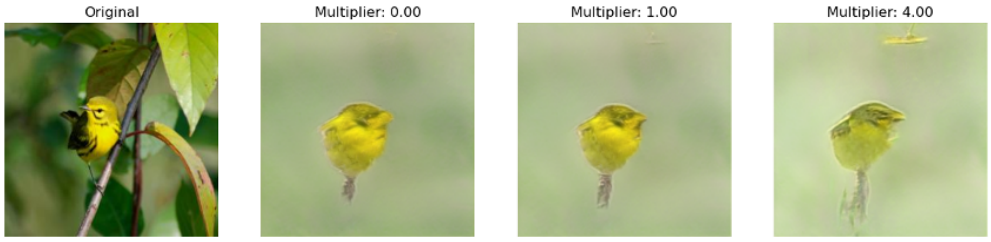
(b) Concept "brown upper parts" in class 175



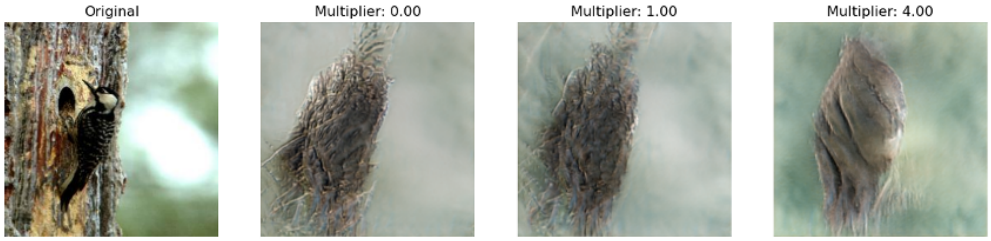
(c) Concept "white belly" in class 194

(d) Concept "brown upper parts" in class 200

Figure 13: Additional qualitative examples obtained for the CUB-200 dataset, threshold = 0.8



(a) Difficult reconstruction in class 176 due to the presence of leaves and branches in the background



(b) Difficult reconstruction in class 190 due to the presence of a trunk in the background

Figure 14: Qualitative example of cases where the reconstruction is difficult due to background predominance, threshold = 0.8

A.3 Training Viscoin on NCT-CRC-HE

Table 8 summarizes the experiments conducted on the NCT-CRC-HE dataset. Other parameters are fixed:

- iterations = 100K
- learning rate = 0.0001
- $\alpha = 0.5$
- $\beta = 3.0$

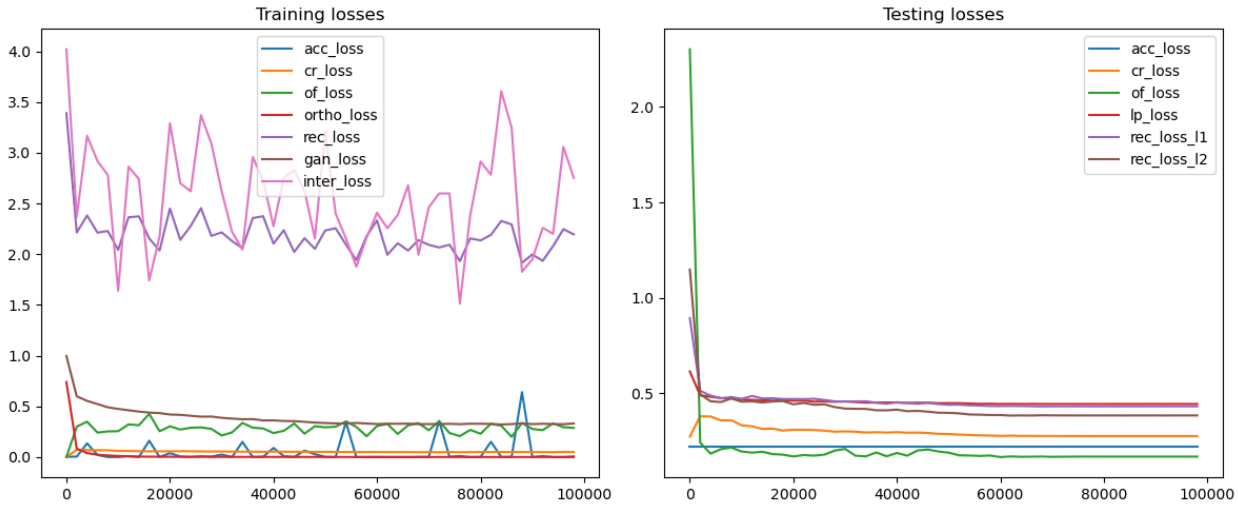


Figure 15: Evolution of the training and testing losses during the training of VisCoIN on the NCT-CRC-HE dataset, with 64 concepts.

Experiment ref	# real images for G training	Coarse layer	Mid-layer	γ	δ	# concepts
6.2.1 - Mid-level	500K	3	12	0.1	0.2	256
6.2.1 - Higher level	500K	**4**	**12**	0.1	0.2	256
6.2.1 - Lower level	500K	**2**	**10**	0.1	0.2	256
6.2.2 - 1.5M real images	1.5M	4	12	0.1	0.2	256
6.2.3 - low γ	1.5M	4	12	**0.05**	0.2	256
6.2.3 - high γ	1.5M	4	12	**0.2**	0.2	256
6.2.3 - low δ	1.5M	4	12	0.1	**0.05**	256
6.2.3 - high δ	1.5M	4	12	0.1	**2**	256
6.2.4 - 16 concepts	1.5M	4	12	0.1	0.2	**16**
6.2.4 - 32 concepts	1.5M	4	12	0.1	0.2	**32**
6.2.4 - 64 concepts	1.5M	4	12	0.1	0.2	**64**

Table 8: Table of experiments. Values that differ from the default parameters are reported in bold.

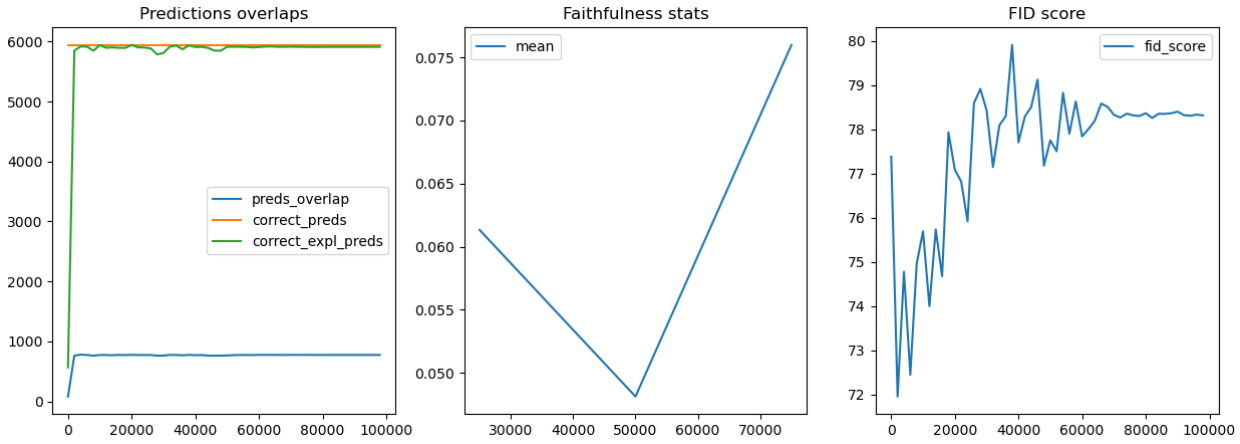
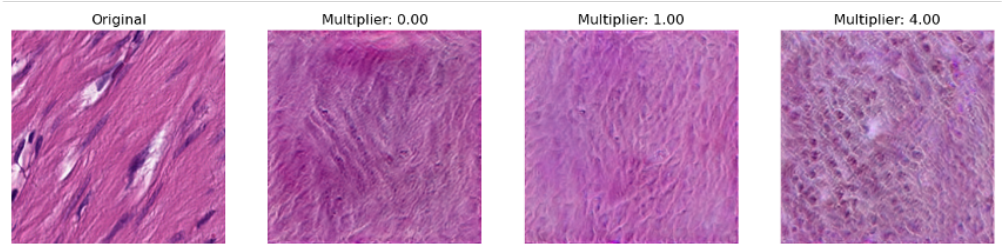
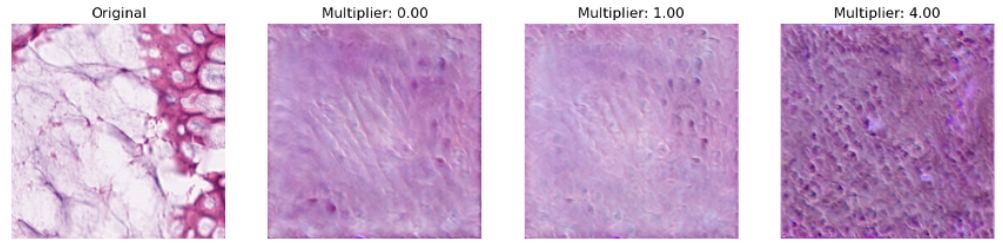


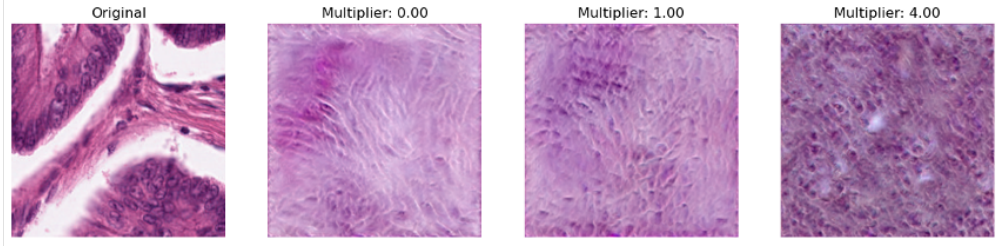
Figure 16: Evolution of predictions overlaps, faithfulness and FID during the training of VisCoIN on the NCT-CRC-HE dataset, with 64 concepts.



(a) Activation of 4 best concepts in class 5 (SMOOTH MUSCLE): "nuclei of smooth muscle cells"



(b) Activation of best concepts (threshold = 0.7) in class 6 (NORMAL COLON MUCOSA): "round shapes"



(c) Activation of 3 best concepts in class 8 (COLORECTAL ADENOCARCINOMA): "pleomorphic nuclei"

Figure 17: Additional qualitative examples obtained for the NCT-CRC-HE dataset, with 64 concepts.