FeatInv: Spatially resolved mapping from feature space to input space using conditional diffusion models

Nils Neukirch
Division AI4Health
Carl von Ossietzky Universität Oldenburg
nils.neukirch@uol.de

Johanna Vielhaben
Explainable Artificial Intelligence Group
Fraunhofer Heinrich-Hertz-Institute
johanna.vielhaben@hhi.fraunhofer.de

Nils Strodthoff
Division AI4Health
Carl von Ossietzky Universität Oldenburg
nils.strodthoff@uol.de

Abstract

Internal representations are crucial for understanding deep neural networks, such as their properties and reasoning patterns, but remain difficult to interpret. While mapping from feature space to input space aids in interpreting the former, existing approaches often rely on crude approximations. We propose using a conditional diffusion model – a pretrained high-fidelity diffusion model conditioned on spatially resolved feature maps – to learn such a mapping in a probabilistic manner. We demonstrate the feasibility of this approach across various pretrained image classifiers from CNNs to ViTs, showing excellent reconstruction capabilities. Through qualitative comparisons and robustness analysis, we validate our method and showcase possible applications, such as the visualization of concept steering in input space or investigations of the composite nature of the feature space. This approach has broad potential for improving feature space understanding in computer vision models.
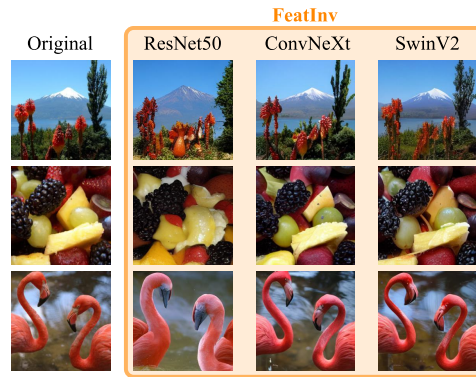
Figure 1: *FeatInv* learns a probabilistic mapping from feature space to input space and thereby provides a visualization of how a sample is perceived by the respective model. The goal is to identify input samples within the set of natural images whose feature representations align most closely with the original feature representation of a given model. In this figure, we visualize reconstructed samples obtained by conditioning on the feature maps of the penultimate layer from ResNet50, ConvNeXt and SwinV2 models.

Preprint. Under review.

1 Introduction

The feature space is vital for understanding neural network decision processes as it offers insights into the internal representations formed by these models as they process input data. While it serves as the foundation for many modern explainability approaches [21, 3], its importance extends beyond interpretability. The feature space provides a rich resource for investigating fundamental properties of deep neural networks, including their robustness against perturbations, invariance characteristics, and symmetry properties [4]. By analyzing the geometry and topology of these learned representations, researchers can gain insights into model generalization capabilities, failure modes, and the emergence of higher-order patterns in the data. This perspective enables advancements in theoretical understanding of neural networks while informing practical improvements in architecture design and training methodologies.

An important challenge in examining the feature space is establishing a connection back to the input domain, especially for classification models that map to labels rather than the same domain as the input. One aspect of this challenge involves identifying which part of the input a particular region or unit in feature space is sensitive to. GradCAM [24] pioneered this by linearly upsampling a region of interest in feature space to the input size. However, linear upsampling imposes a rather strong implicit assumption. As an alternative, one might consider the entire receptive field of a feature map location, yet in deep architectures these fields tend to be broad and less informative.

The more intricate second aspect of this challenge is to derive a mapping from the entirety of the feature space representation back to the input domain – beyond mere localization. Recent works proposed to leverage conditional generative models to learn such a mapping by conditioning them on feature maps [4, 6, 23]. However, these approaches either build on pooled feature maps (discarding finegrained spatial details of the feature map), only provide deterministic mappings (overlooking the inherent uncertainty of this ill-posed problem), or do not utilize state-of-the-art generative models. To the best of our knowledge, there is no probabilistic model that provides high-fidelity input samples when conditioned on a spatially resolved feature map – thereby integrating both aspects of the challenge described above. We aim to close this gap with this submission.

More specifically, in this work we put forward the following contributions:

1. We demonstrate the feasibility of learning high-fidelity mappings from feature space to input space using a conditional diffusion model of the ControlNet-flavor, as exemplified in Fig. 1. We investigate this for different computer vision models, ranging from CNNs to ViTs.

2. We provide quantitative evidence that generated samples align with the feature maps of the original samples and that the samples represent high-fidelity natural images, see Tab. 1. and carry out a qualitative model comparison, see Fig. 3 as well as a robustness analysis, see Tab. 2.

3. We provide a specific use-cases for the application of the proposed methodology to visualize concept-steering in input space, see Fig. 4, as well as to provide insights into the composite nature of the feature space, see Fig. 5.

2 Methods

Approach In this work, we propose a method called *FeatInv* to approximate an inverse mapping from a model's feature space to input space. Our method conditions a pretrained stable diffusion model on a spatially resolved feature map extracted from a pretrained CNN/ViT model of our choice. As described in detail in the next paragraph, the feature maps are provided as conditional information along with an unspecific text prompt ("a high-quality, detailed, and professional image") to a conditional diffusion model of the ControlNet [29] flavor. Importantly, rather than achieving a precise reconstruction of the original sample in input space, our goal is to infer high-fidelity, synthetic images whose feature representations align with those of the original image when passed through a pretrained CNN/ViT model.

Architecture and training procedure We use a ControlNet [29] architecture, building on a pretrained diffusion models, in our case a MiniSD [20] model operating at an input resolution of 256×256. The ControlNet is a popular approach to condition a pretrained diffusion model on dense inputs such as segmentation maps or depth maps. It leverages a pretrained (text-conditional) diffusion
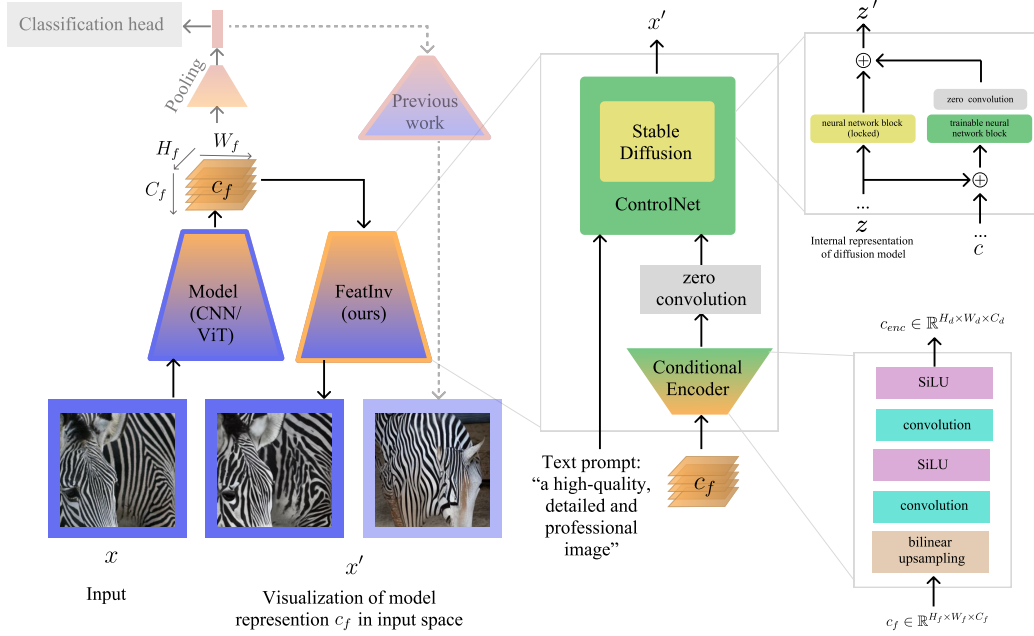
Figure 2: **Schematic overview of the *FeatInv* approach.** *Left:* Given a spatially resolved feature map c_f of some pretrained model, we aim to infer an input x' within the set of natural images, whose feature representation aligns as closely as possible with c_f, i.e., to learn a probabilistic mapping from feature space to input space. Previous work consider spatially pooled feature maps, whereas this work conditions on spatially resolved feature maps. *Middle:* We leverage a pretrained diffusion model, which gets conditioned on c_f by means of a ControlNet architecture, which parametrizes an additive modification on top of the frozen diffusion model. *Right top:* The ControlNet adds trainable copies of blocks in the stable diffusion model, which are conditioned on the conditional input and added to the output of the original module, which is kept frozen. *Right bottom:* The feature map c_f is processed through bilinear upsampling and a shallow convolutional encoder to serve as conditional input for the ControlNet.

model, whose weights are kept frozen. The trainable part of the ControlNet model mimics the internal structure of the pretrained diffusion model, with additional layers introduced to incorporate conditioning inputs. These conditional inputs are processed by a dedicated encoder and inserted into the corresponding computational blocks, where their outputs are added to those of the original diffusion model. Convolutional layer that are initialized to zero ensure that the optimization of the ControlNet model starts from the pretrained diffusion model.

Conditional input encoder An import design choice is the conditional input encoder, which maps the feature map (with shape $H_f \times W_f \times C_f$, where H_f, W_f, C_f correspond to the height, width and channels of the feature map, respectively) to the diffusion model's internal representation space (with shape $H_d \times W_d \times C_d$). As a definite example for 224×224 input resolution, for the output of ResNet50's final convolutional block with $H_f = W_f = 7$, $C_f = 2048$ we learn a mapping to the diffusion model's internal representation space with $H_d = W_d = 32$ and $C_d = 320$. To this end, we first use bilinear upsampling to reach the target resolution. Then, we allow for a shallow CNN to learn a suitable mapping from the model's representation space to the diffusion model's representation space.

Pooled vs. unpooled To demonstrate superiority over prior work [4], we also consider the case of pooled feature representations obtained from average-pooling spatial tokens/feature maps. In order to process them using the same pipeline as for conditioning on spatially resolved feature maps, we copy the C_d-dimensional input vector along H_d and W_d times to reach an input tensor with shape $H_d \times W_d \times C_d$ as before.

Training The ControlNet is trained using the same noise prediction objective as the original diffusion model [12]. Control signals are injected at multiple layers throughout the network, rather than being restricted to the middle layers, allowing them to influence the denoising process at various stages. Training was conducted on the ImageNet training set with a batch size of 8 and a learning rate of 1e-5 using an AdamW optimizer with the stable diffusion model locked. The ControlNet was trained on ImageNet for three epochs over approximately 45 to 60 hours (depending on backbone) of compute time on two NVIDIA L40 GPUs. During the course of this project, about five times more models were trained until the described setup was reached.

Full pipeline We work with the original input resolution of the respective pretrained models, which varies between 224×224 and 384×384 for the considered models, see the supplementary material for a detailed breakdown. Even though the approach allows conditioning on any feature map, we restrict ourselves to the last spatially resolved feature map, i.e., directly before the pooling layer, and learn mappings to MiniSD's internal feature space. The MiniSD model always returns an image with resolution 256×256, which we upsample/downsample to the model's expect input resolution via bilinear upsampling/downsampling. The full generation pipeline is visualized in Fig. 2.

3   Related Work

Conditional diffusion models Achieving spatially controllable image generation while leveraging a pretrained diffusion model has been a very active area of research recently, see [30] for a recent review. Applications include the conditional generation of images from depth maps, normal maps or canny maps. Popular approaches in this direction include ControlNet [29] or GLIGEN [14]. The mapping from feature maps as conditional input is structurally similar to the mentioned cases of spatially controllable generation. However, there is a key distinction. In the previously mentioned cases, the conditional information typically matches the resolution of the input image. This often necessitates downsampling to reach the diffusion model's internal representation space. In contrast, commonly used classification models (including CNNs and vision transformers) leverage feature maps with a reduced spatial resolution. Consequently, the spatial resolution of the conditional information is typically lower dimensional than the diffusion model's internal representation space. This difference necessitates an upsampling operation before conditioning on feature maps.

Feature visualization The idea to reveal structures in feature space to understand what a neural network has learned is an old one. Approaches range from identifying input structures or samples that maximize the activation of certain feature neurons [7, 19] to approximate inversion of the mapping from input to features space [28]. Our approach clearly stands in the tradition of the latter approach. Previous work has attempted to learn a deterministic mapping that "inverts" AlexNet feature maps [6]. This approach was recently extended to invert vision transformer representations [22]. In contrast, FeatInv learns a probabilistic mapping using state-of-the-art diffusion models and investigates state-of-the-art model architectures. Other approaches tackle the problem using invertible neural networks to connect VAE latent representations to input space [23] and/or disentangle these representations using concept supervision [8]. In contrast, FeatInv does not rely on a particular encoder/decoder structure but can use any pretrained neural network as encoder. The closest prior work to our approach is [4], which also uses a diffusion model to learn a mapping from feature space to input space. However, it uses pooled representations as input, i.e. neglects the spatial resolution of the feature map. We argue that pooled representations are too coarse for many applications as they disregard the finegrained spatial structure of the feature space.

Representation surgery Finally, related feature inversion approaches have also been explored beyond computer vision, for example in natural language processing [18]. Here, the ability to invert latent representations is seen as an essential component for representation surgery approaches [1]. *FeatInv* enables similar approaches for computer vision models.

4   Results

We investigate three models ResNet50 [11] (original torchvision weights), ConvNeXt [16] and SwinV2 [15] [1] all of which have been pretrained/finetuned on ImageNet1k. ConvNeXt and SwinV2 represent modern convolution-based and vision-transformer-based architectures, identified as strong

---

[1]timm model weights: convnext_base.fb_in22k_ft_in1k, swinv2_base_window12to24_192to384_22kft1k.

Table 1: **Reconstruction quality and image quality of the individual models**: For the three considered backbones, we indicate three performance metrics to assess the reconstruction quality: Cosine similarity in feature space (cosine-sim), calculated by taking the median of the cosine similarity of all superpixels, top5(1) matches using the top1 prediction of the original sample as ground truth (top5(1) match) and FID-scores (FID) to assess the quality of the generated samples. We consider generative models conditioned on unpooled feature maps (rows 1-3) and models conditioned on pooled feature maps (rows 4-6). The results indicate that the proposed approach produces high-fidelity input samples as perceived by the respective models.

	Model	cosine-sim	top5(1) match	FID
unpooled	ResNet50	0.40	87% (63%)	12.32
	ConvNeXt	0.61	94% (76%)	7.14
	SwinV2	0.57	95% (81%)	11.56
pooled	ResNet50	0.09	47% (22%)	26.37
	ConvNeXt	0.18	42% (18%)	31.01
	SwinV2	0.14	47% (23%)	36.21

backbones in [10]. We include ResNet50 due to its widespread adoption. For each model, we train a conditional diffusion model conditioned on the representations of the last hidden layer before the final pooling layer to reconstruct the original input samples. Below, we report on quantitative and qualitative aspects of our findings.

4.1 Quantitative and qualitative comparison

Experimental setup For each ImageNet class, we reconstructed 10 validation set samples with FeatInv, resulting in 10000 reconstructed samples. We adjust the diffusion model's guidance scale to optimize the FID score between original and reconstructed samples on the validation set, resulting in a guidance scale of 8(1.75) for (un)pooled conditional input, see the supplementary material for details. We reconstruct with a sample step size of 50. For each model this took roughly 12 hours on a single NVIDIA L40 GPU. The generated samples are assessed according to two complementary quality criteria, reconstruction quality and sample quality:

1. **Reconstruction quality** The encoded generated image should end up close to the feature representation of the original samples, which can be understood as a reconstruction objective that is implemented implicitly by conditioning the diffusion model on a chosen feature map. (1a) The most obvious metric is cosine similarity between both feature maps. However, not all parts of the feature space will be equally important for the downstream classifier. (1b) Most reliable measure is the classifier output itself. Focusing on top-predictions, one can also compare top-k predictions to the top prediction for the original sample. More general alignment measures between generated input and original feature representation are not helpful in this context, as we require a precise reconstruction of the original feature space for the downstream classifier above the layer under consideration.

2. **Sample quality** We aim to generate samples within the set of high-fidelity natural images. In our case, this objective is is implemented through the use of a pretrained diffusion model. Apart from qualitative assessments in the following sections, we rely on FID-scores as established measures to assess sample quality.

Reconstruction quality Comparing identical models conditioned either on pooled or unpooled feature maps, not surprisingly unpooled models show a significantly higher reconstruction quality. Samples generated by models conditioned on unpooled feature maps show a very good alignment with the feature maps of the original samples (cosine similarities above 0.57 and top5 matching predictions of 94% or higher for the two modern vision backbones). Samples conditioned on pooled feature maps show some alignment but fail to accurately reconstruct the respective feature map and are therefore unreliable for investigations of structural properties of models. These findings support
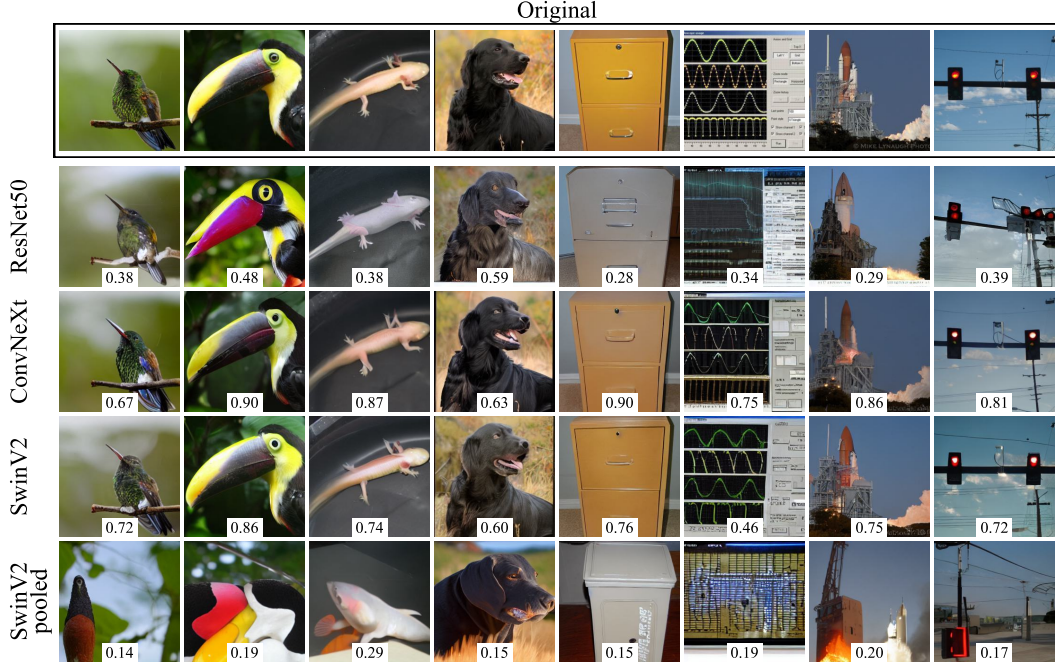
Figure 3: **Qualitative comparison of reconstructed samples** for the ResNet50, ConvNeXt, SwinV2 and SwinV2 pooled models. The cosine similarity of the original feature map and that of the reconstruction is noted at the bottom edge of the images. The qualitative comparison confirms the insights from the quantitative analysis in Tab. 1. The two modern vision backbones, ConvNeXt and SwinV2, show reconstructions that resemble the original very closely, not only in terms of semantic content and spatial alignment but also in terms of color schemes and finegrained details. Semantic content and composition also mostly matches in case of the ResNet50, but not even the semantic content seems to be captured when using pooled representations (SwinV2 pooled as an example).

the hypothesis that the approach yield feature space reconstructions that closely match the original feature representations.

Sample quality The corresponding class-dependent diffusion model achieves an FID score around 29, which is typically considered as good quality. The models conditioned on pooled representations still show acceptable FID scores between 26 and 36. Interestingly, models conditioned on unpooled representations show a significant increase in image quality with FID scores between 7 and 12. These results support the statement that the created samples were sampled from the space of high-fidelity natural images.

Backbone comparison Within each category (pooled vs. unpooled), there is a gap between the two most recent model architectures ConvNeXt and SwinV2, notwithstanding the architectural differences (CNN vs. Vision transformer) between the two, in comparison to the older Resnet50 models. The former achieve cosine similarities of .6 or higher and top5 matches of 95% or higher in the unpooled category. This suggests that there is a qualitative difference between the representations of ResNet50-representations and representations of more modern image backbones.

Qualitative comparison In Fig. 3, we present a qualitative comparison based on randomly selected samples. The visual impressions of ConvNeXt and SwinV2 reconstructions are similar to each other while also being close to the input sample despite the fact that they were trained on high-level semantic feature maps, i.e., without a reconstruction objective in input space. The ResNet50 reconstructions seem in many cases an interpretation of the sample's semantic content (see e.g. 2. toucan or 5. file), albeit with the correct spatial composition, while matching specific color composition and textures much less accurately than ConvNeXt and SwinV2. We primarily attribute the differences between ResNet and ConvNeXt/SwinV2 to the nature of the feature spaces themselves, stressing qualitative difference between modern architectures such as ConvNeXt and SwinV2 and older model architectures such as ResNet50, which are much more pronounced than the differences between

Table 2: **Cross-model evaluation**: Percentage of matching of the actual predictions (top5/top1) and the predictions based on the reconstructions for different models. The FeatInv models based on the ResNet50, ConvNeXt and SwinV2 features were used for the reconstruction and evaluated by the same three models.

	evaluated by		
conditioned on	ResNet50	ConvNeXt	SwinV2
ResNet50	87% / 63%	85% / 60%	87% / 65%
ConvNeXt	92% / 72%	94% / 76%	95% / 80%
SwinV2	94% / 77%	95% / 79%	95% / 81%

different model architectures such as ViTs and CNNs. The samples obtained from conditioning on pooled feature representations often seem to capture overall semantic content of the image correctly (file, space shuttle, traffic light), but fail to reflect the details of the composition of the image.

Robustness evaluation To assess the robustness of the presented results, we carry out cross-model comparisons where we measure model performance based on samples generated by conditioning on the feature map extract from a different model. The results for this experiment are compiled in Tab. 2. It turns out that all three sets of samples (conditioned on features generated by the three different backbones) transfer quite remarkably to other models. In the supplementary material, we also present results supporting the robustness of our approach when applied to out-of-distribution (OOD) samples.

4.2 Application: *FeatInv-Viz* – Visualizing concept steering in input space

Concept steering in input space In generative NLP, steering is sometimes used to verify concept interpretations by reducing or magnifying concepts in the model activations and observing how this changes the generated output text, as famously demonstrated at the example of the Golden Gate concept [5] in Claude 3 Sonnet. This approach is not directly applicable to vision classifiers. However, with our method of inverting model representations from feature to input space, we can observe the effect of concept steering within hidden model activations in the input representation space instead of the output. This enables a novel method for concept visualization, with benefits over existing approaches (see below).

Concept definition Concepts are typically defined as structures in feature space such as individual neurons, single directions or multi-dimensional subspaces. Many concept-based XAI methods define a way to decompose a feature vector into concepts from a dictionary/concept bank [9]. In this work, we use concepts from multi-dimensional concept discovery (MCD) [25], which defines concepts as linear subspaces in feature space. Nevertheless, our approach is applicable to any concept discovery method.

Concept visualization through attenuated feature maps A common challenge for unsupervised concept discovery methods is inferring the meaning of discovered concepts. To address this, we steer a concept in feature space and observe the effect in input space. Specifically, we attenuate coefficients for the concept under consideration to 25% (see the Supplementary Material for a details). Then, we use FeatInv to map the original and the modified feature map to input space using identical random seeds for the diffusion process. By comparing the resulting images, we gain insights into how the concept is expressed in input space. We call this method *FeatInv-Viz* and present it in Algorithm 1.

Algorithm 1: *FeatInv-Viz* : Visualization of concept steering in input space

Input: Model m, concept decomposition $\phi = \sum_i \phi_i$, concept with id c
Output: Visualization of concept c in input space
Notation: $x \in \mathbb{R}^{3 \times H \times W}$ where $x^{(j)}$ refers to color channels with $j \in \{R, G, B\}$
$\phi' \leftarrow \sum_{i \neq c} \phi_i + 0.25 \cdot \phi_c$;   `// Attenuated feature map`
for $i = 1$ **to** n **do**
    $s_i \leftarrow \text{RandomSeed}()$
    $x_i \leftarrow \text{FeatInv}(\phi, seed = s_i)$;   `// Original reconstruction`
    $x'_i \leftarrow \text{FeatInv}(\phi', seed = s_i)$;   `// Attenuated reconstruction`
    $\Delta_i \leftarrow \sqrt{\sum_{j \in \{R,G,B\}} (x_i^{(j)} - x_i'^{(j)})^2}$;   `// Euclidean distance`
return $median\{\Delta_i\}_{i=1}^n$;   `// Median along sample axis`

Exemplary results Fig. 4 shows exemplary concept steering visualizations for four samples from the Indigo Bunting class. Here, we decomposed ConvNeXt's feature space into three linear concept subspaces. *FeatInv-Viz* provides a visualization of these concepts in input space. The method provides a very finegrained visualization of which specific regions in input space change upon steering each concept in feature space.

Benefits We emphasize that *FeatInv-Viz* extends commonly used concept activation maps in two ways: First, it provides a finegrained visualization rather than a coarse upscaling [2, 25] of a lower-resolution feature map. Second, it goes beyond merely verifying alignment with a predefined concepts [2], by providing counterfactual information from targeted feature-map manipulations.
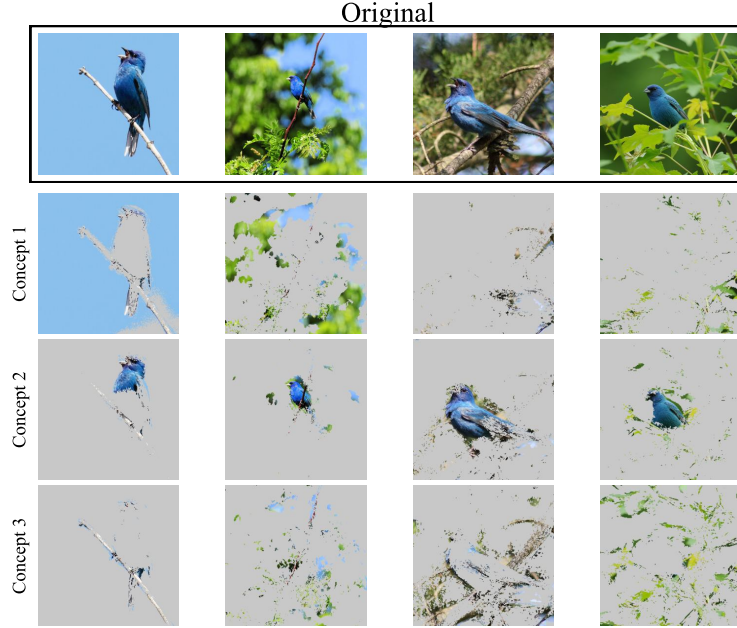

Figure 4: *FeatInv-Viz* **visualization of three concepts** identified within ConvNeXt's feature space of the Indigo Bunting class, which can be associated with sky/background, bird head/breast and branches/leaves. For the visualization we normalize the respective outputs of Algorithm 1 and threshold it below 0.33 as a binary mask to indicate unaffected regions of the image.

4.3 Application: Investigating the composite nature of the feature space

In NLP, well-known examples of feature-space arithmetic – e.g. king − man + woman = queen [17] – have shaped our understanding of embedding geometries. FeatInv offers insights into the composite nature of the feature space in vision models by conditioning on feature maps from two samples. In particular, we investigate the effect of convex linear superpositions of two feature maps. To this end we linearly interpolate between the feature representations of two input samples and visualize

reconstructions for different weighted combinations, as shown in Fig. 5. We also indicate the cosine similarity between the reconstruction and the weighted feature map, which is highest for the original feature maps and typically reaches its lowest value for the equally weighted interpolated feature map. This can be seen as an indication that the weighted average of two feature maps is in general not a well-defined operation. Nevertheless, foreground objects from one image and background from a second, seem to be reasonably combined through linear superposition (see e.g. bird, landscape).
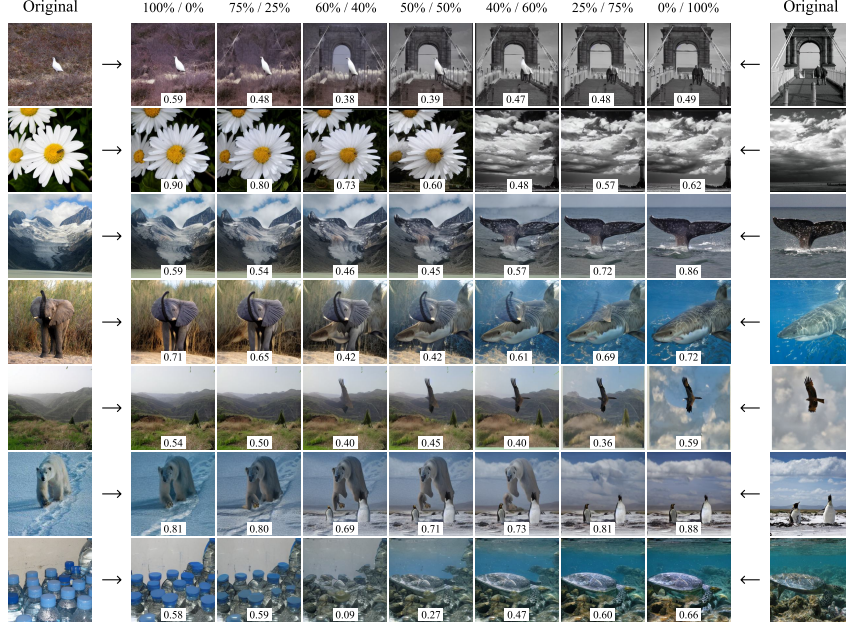

Figure 5: **Reconstructions from weighted combinations of two ConvNeXt feature maps**. The cosine similarity between the weighted feature map and that of the reconstruction is noted at the bottom edge of the images.

4.4 Limitations and future work

Our work is subject to different limitations, which provide directions for future investigations: First, the present work focuses exclusively on the domain of natural images. It would be very instructive to extend the approach to other domains, such as medical imaging. Second, the proposed approach building on the ControlNet method, builds on a pretrained diffusion model, which might not be readily available in any application contexts. Third, every model and layer choice requires training a dedicated FeatInv model, which represents a computation hurdle. First experiments and the results in Tab. 2 indicate that finetuning could be beneficial to alleviate this issue. Finally, both application scenarios rely on modifications of the feature space. In order to obtain reliable results, it would be instrumental to introduce measures to detect input samples, i.e., feature maps that are outside the scope of the model.

5 Summary and Discussion

In this work, we address the problem of obtaining insights into the structure of a given model's feature map by means of a learned probabilistic mapping from feature space to input space, implemented as a conditional diffusion model. We demonstrate the feasibility of training such a model in a ControlNet-style achieving very accurate and robust reconstruction results across different model architectures. We present two possible applications both of which relate to gaining inside into manipulated feature maps. However, we believe that the proposed approach could be widely applicable to further applications. We envision a potentially positive societal impact through improved model understanding, along the lines of the concept steering use case. The source code underlying our investigations is available at `https://github.com/AI4HealthUOL/FeatInv`.

References

[1] M. Avitan, R. Cotterell, Y. Goldberg, and S. Ravfogel. A practical method for generating string counterfactuals. *arXiv preprint 2402.11355*, 2025. URL `https://arxiv.org/abs/2402.11355`.

[2] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6541–6549, 2017.

[3] L. Bereska and S. Gavves. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=ePUVetPKu6`. Survey Certification, Expert Certification.

[4] F. Bordes, R. Balestriero, and P. Vincent. High fidelity visualization of what your self-supervised representation knows about. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL `https://openreview.net/forum?id=urfWb7VjmL`.

[5] T. Bricken, A. Reddy, T. Conerly, V. Varma, L. Chan, C. Burns, and N. Nanda. Scaling monosemanticity: Learning features that resist polysemanticity in large language models. *Transformer Circuits*, 2024. URL `https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html`. Accessed: April 29, 2025.

[6] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29, 2016.

[7] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. 2009.

[8] P. Esser, R. Rombach, and B. Ommer. A disentangling invertible interpretation network for explaining latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9223–9232, 2020.

[9] T. FEL, V. Boutin, L. Béthune, R. Cadene, M. Moayeri, L. Andéol, M. Chalvidal, and T. Serre. A holistic approach to unifying automatic concept extraction and concept importance estimation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 54805–54818. Curran Associates, Inc., 2023.

[10] M. Goldblum, H. Souri, R. Ni, M. Shu, V. Prabhu, G. Somepalli, P. Chattopadhyay, M. Ibrahim, A. Bardes, J. Hoffman, et al. Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. *Advances in Neural Information Processing Systems*, 36, 2024.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[13] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

[14] Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, and Y. J. Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023.

[15] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022.

[16] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[17] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In L. Vanderwende, H. Daumé III, and K. Kirchhoff, editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL `https://aclanthology.org/N13-1090/`.

[18] J. X. Morris, V. Kuleshov, V. Shmatikov, and A. M. Rush. Text embeddings reveal (almost) as much as text. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL `https://openreview.net/forum?id=EDuKP7DqCk`.

[19] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems*, 29, 2016.

[20] J. Pinkney. minisd. `https://huggingface.co/justinpinkney/miniSD`, 2023. Hugging Face Model Repository.

[21] D. Rai, Y. Zhou, S. Feng, A. Saparov, and Z. Yao. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*, 2024.

[22] J. Rathjens, S. Reyhanian, D. Kappel, and L. Wiskott. Inverting transformer-based vision models. *arXiv preprint 2412.06534*, 2024. URL `https://arxiv.org/abs/2412.06534`.

[23] R. Rombach, P. Esser, and B. Ommer. *Making Sense of CNNs: Interpreting Deep Representations and Their Invariances with INNs*, page 647–664. Springer International Publishing, 2020. ISBN 9783030585204. doi: 10.1007/978-3-030-58520-4_38. URL `http://dx.doi.org/10.1007/978-3-030-58520-4_38`.

[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[25] J. Vielhaben, S. Bluecher, and N. Strodthoff. Multi-dimensional concept discovery (MCD): A unifying framework with completeness guarantees. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=KxBQPz7HKh`.

[26] R. Wightman, H. Touvron, and H. Jegou. Resnet strikes back: An improved training procedure in timm. In *NeurIPS 2021 Workshop on ImageNet: Past, Present, and Future*, 2021. URL `https://openreview.net/forum?id=NG6MJnVl6M5`.

[27] R. Yi, H. Tian, Z. Gu, Y.-K. Lai, and P. L. Rosin. Towards artistic image aesthetics assessment: A large-scale dataset and a new method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22388–22397, June 2023.

[28] M. Zeiler. Visualizing and understanding convolutional networks. In *European conference on computer vision*, volume 1311, 2014.

[29] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[30] T. Zhang, Z. Wang, J. Huang, M. M. Tasnim, and W. Shi. A survey of diffusion based image generation models: Issues and their solutions. *arXiv preprint arXiv:2308.13142*, 2023.

A Ablation studies

A.1 Additional models

For the ablation studies, we consider ResNet50B[2] [26] as second ResNet-model to study the impact of the training procedure, in addition to the models studied in the main text.

Choice of conditional encoder Next to the bilinear upsampling followed by a shallow CNN, we explore a different design choice for the conditional encoder: To this end, we explore the use of cross-attention as used in the Perceiver architecture [13], which uses a learnable representation of predefined size and connects it via cross-attention to the representation that serves as input for the mapping. Unlike the previous approach, this approach is not subject to any locality assumptions and therefore the most flexible approach that is in particular suitable for model architectures without built-in locality assumptions such as vision transformers. We carry out a comparison between the two conditional encoder models for the case of the SwinV2 ViT, where we expect the impact to be most pronounced as the ViT operates on visual tokens, which might not align well with the upscaling operation in the convolutional encoder. On the contrary, our results indicates that the convolutional encoder yields better classification results and therefore for the following experiments we restrict ourselves to the convolutional encoder.

Class-conditional baseline We tested the models on an unconditional baseline to see if MiniSD is able to generate good representations of the classes and if they can be correctly classified. To define the classes for the input prompt as precisely as possible, we use the WordNet hierarchy and create the prompt as follows: 'a high-quality, detailed, and professional image of a CLASS, which is a kind of SUPERCLASS' as for example 'a high-quality, detailed, and professional image of a tench, which is a kind of cyprinid'. The unconditional baseline created consists of 10 samples per class, i.e. 10,000 samples in total.

Table 3: **Additional model insights**: For the different backbones, we indicate the corresponding canonical input sizes as well as the top5(1) accuracy on an unconditional dataset, created by MiniSD. Since only the base model is used for the prediction of these unconditional samples, the results do not differ between unpooled and pooled as well as for the different SwinV2 architectures. We show three performance metrics: Cosine similarity in feature space (cosine-sim), calculated by taking the median of the cosine similarity of all superpixels, top5(1) matches using the top1 prediction of the original sample as ground truth (top5(1) match) and FID-scores (FID) to assess the quality of the generated samples. We consider generative models conditioned on unpooled feature maps (rows 1-5) and models conditioned on pooled feature maps (rows 6-9).

	Model	input	MiniSD top5(1)	cosine-sim	top5(1) match	FID
unpooled	ResNet50	224×224	89% (68%)	0.40	87% (63%)	12.32
	ResNet50B		88% (65%)	0.50	90% (68%)	18.58
	ConvNeXt	288×288	92% (71%)	0.61	94% (76%)	7.14
	SwinV2	384×384	93% (72%)	0.57	95% (81%)	11.56
	SwinV2(Perceiver)			0.37	79% (55%)	22.72
pooled	ResNet50	224×224	89% (68%)	0.09	47% (22%)	26.37
	ResNet50B		88% (65%)	0.14	44% (21%)	37.44
	ConvNeXt	288×288	92% (71%)	0.18	42% (18%)	31.01
	SwinV2	384×284	93% (72%)	0.14	47% (23%)	36.21

A.2 FID scores vs. Guidance Scale

In Fig. 6, we present several original images alongside their respective reconstructions, generated using the model trained with the ConvNeXt backbone. These samples were produced with varying levels of guidance scale. Since the guidance scale significantly influences the output, we evaluated the model's performance across different scale settings, again using the ConvNeXt-based model. To quantify the quality of the generated images, we employed FID and cosine distance, as illustrated

---

[2]timm model weights: resnet50.a1_in1k

in Fig. 7. We opted to use cosine distance (rather than similarity) to ensure that, in both metrics, lower values indicate better performance, simplifying comparison. This suggests that the unpooled feature map requires minimal text guidance, whereas for the pooled variant, guidance is of substantial importance, as natural looking images cannot be generated without its influence.
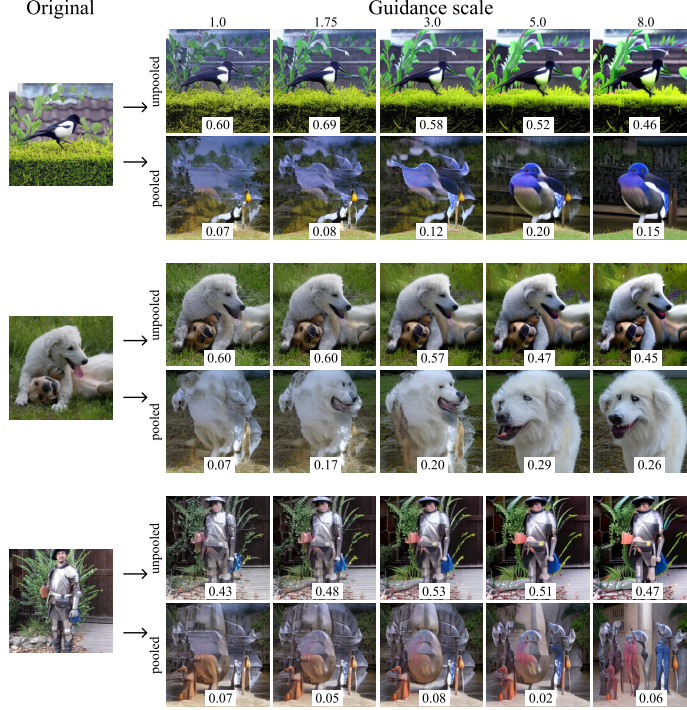

Figure 6: **Unpooled and pooled reconstructions with different guidance scales from ConvNeXt feature maps**. The cosine similarity between the original feature map and that of the reconstruction is noted at the bottom edge of the images. While increasing the guidance scale in unpooled samples results in higher color saturation and more distorted object shapes, the same increase improves object fidelity in pooled feature maps. This trend is also roughly reflected in the corresponding cosine distances.
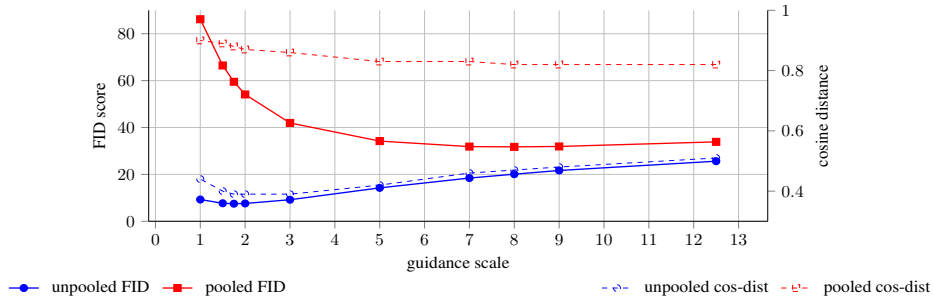

Figure 7: **Impact of guidance scale on generation quality (FID) and semantic alignment (cos-dist)**. FID values (solid lines) and cosine distance (dashed lines) for models with ConvNeXt backbone with and without pooling. Lower values in both metrics indicate better performance. The unpooled model (blue) achieves optimal quality at a guidance scale of 1.75, with both low FID and low cosine distance. In contrast, the pooled model (red) performs significantly worse at all scales, with increasing FID and cosine distance as the guidance scale decreases. This model reaches its best performance near a scale of 8.0.

A.3 Generation on unpooled vs. pooled feature maps

We analyzed the generated samples for each model using both unpooled and pooled feature maps. As shown in Tables 1 and 3, the samples generated from pooled feature maps consistently exhibit lower visual quality compared to those generated from unpooled feature maps. This indicates that preserving spatial detail during feature extraction is crucial for high-quality image generation. Fig. 8 illustrates this difference with examples from the class zebra, clearly highlighting the superior fidelity of samples generated from unpooled features.
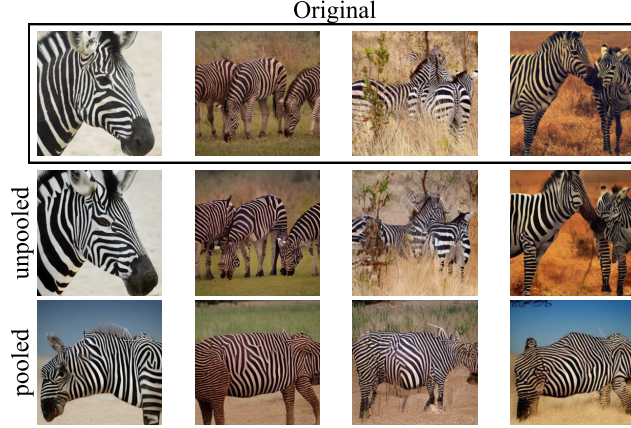

Figure 8: **Comparison of unpooled and pooled generated samples** showing examples of the class zebra using. All samples were generated using the model trained with the ConvNeXt backbone. The unpooled feature maps result in sharper and more realistic generations, while the pooled versions show a noticeable loss in detail and structure, mostly showing a general context of the original image. The cosine similarity between the original feature map and that of the reconstruction is noted at the bottom edge of the images

B OOD evaluation

In Fig. 9, we qualitatively test the robustness of our findings by conditioning on samples that are slightly out of the model scope of the original models finetuned on ImageNet. To this end, we use the BAID [27] dataset, which differs in style from the samples in ImageNet. The ControlNet trained on ConvNeXt features still shows a good reconstruction quality of the semantic content but the style of the reconstruction and the original image differ more strongly than in the in-distribution case in Fig. 3. Nevertheless, the results speak for the robustness of the proposed approach.
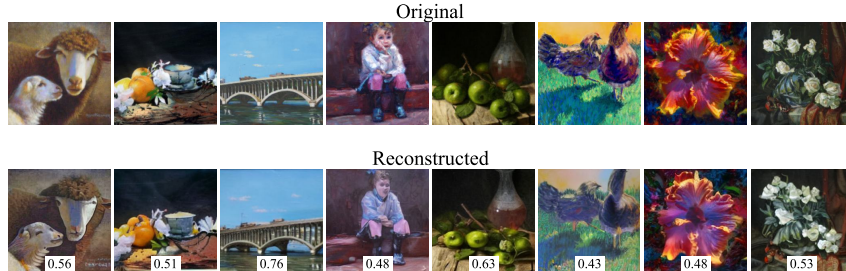

Figure 9: **Reconstruction on OOD samples**. Comparison of randomly selected original and generated samples from the BAID [27] dataset, which differs in style from the samples in ImageNet. The basis for the reconstruction was the ControlNet trained on ConvNeXt features, which received the ConvNeXt features of the samples shown as input. The cosine similarity between the original feature map and that of the reconstruction is noted at the bottom edge of the images

C Details on concept steering

For concept discovery, we rely on multi-dimensional concept discovery (MCD) [25]. For every feature vector ϕ, MCD provides a concept decomposition $\phi = \sum_{i=1}^{n_c+1} \phi_i$, where ϕ_i is associated with the concept i (of in total n_c concepts), which represents a linear subspace of the feature space, and concept $n_c + 1$ corresponds to the orthogonal complement of the span of all concept subspaces. The latter is necessary to achieve a complete decomposition not explicitly captured through concept subspaces. For a given feature vector, one can now quantify the contribution ϕ_i from concept i and visualize its magnitude $|\phi_i|_2$ across the entire feature map to obtain a spatially resolved concept activation map. One option to align such a coarse concept activation map with the input image is to use bilinear upsampling. This process often leads to rather diffuse concept activation maps. Even though we use MCD for demonstration, this alignment step is a common challenge for most concept-based attribution maps.

We used the learned mapping from feature space to input space to infer high-resolution concept visualizations. To this end, the component ϕ_i associated with concept i was multiplied by 0.25 to attenuate them. In our experiments, decreasing the feature values worked better than increasing them. We speculate that increasing feature vector components can eventually result in feature vectors that exceed the magnitude of feature vectors seen during training of the FeatInv model.

To obtain higher-resolution representations, we used FeatInv to reconstruct five samples for each concept using both the original feature map and the concept-manipulated feature map. Using the same random seed for each pair ensured that the original and manipulated reconstructions were directly comparable, with differences attributable solely to the feature manipulation. For each pair, we computed the pixel-wise difference, and to produce a representative difference map for each concept, we took the median across the five resulting difference maps. This yielded a high-resolution (256×256) activation map that highlights the specific regions of the image affected by the manipulation.

D Spatial composition

In Fig. 10, we show spatially composed combinations of two feature maps. The results indicate that feature maps exhibit a very local influence, which aligns well with the simple upscaling of the feature map resolution to the input resolution.
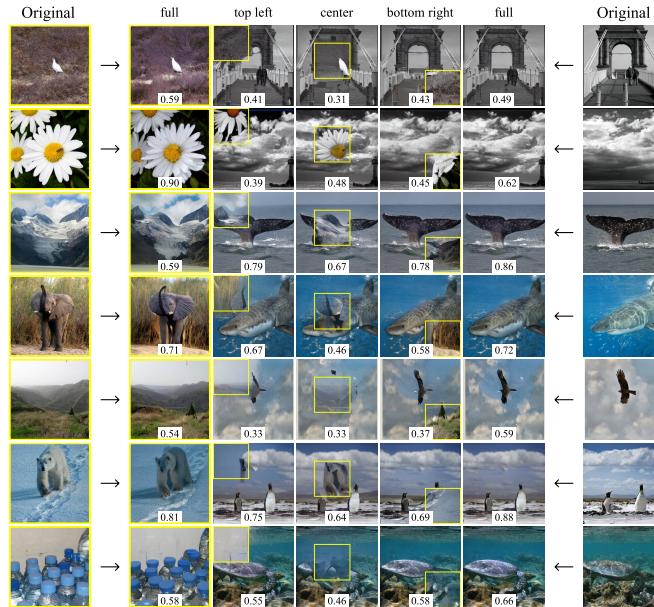

Figure 10: **Reconstructions of spatially composed mixtures of two ConvNeXt feature maps**. The cosine similarity between the manipulated map and that of the reconstruction is noted at the bottom edge of the images. The yellow outlines show the part of the feature map that was manipulated