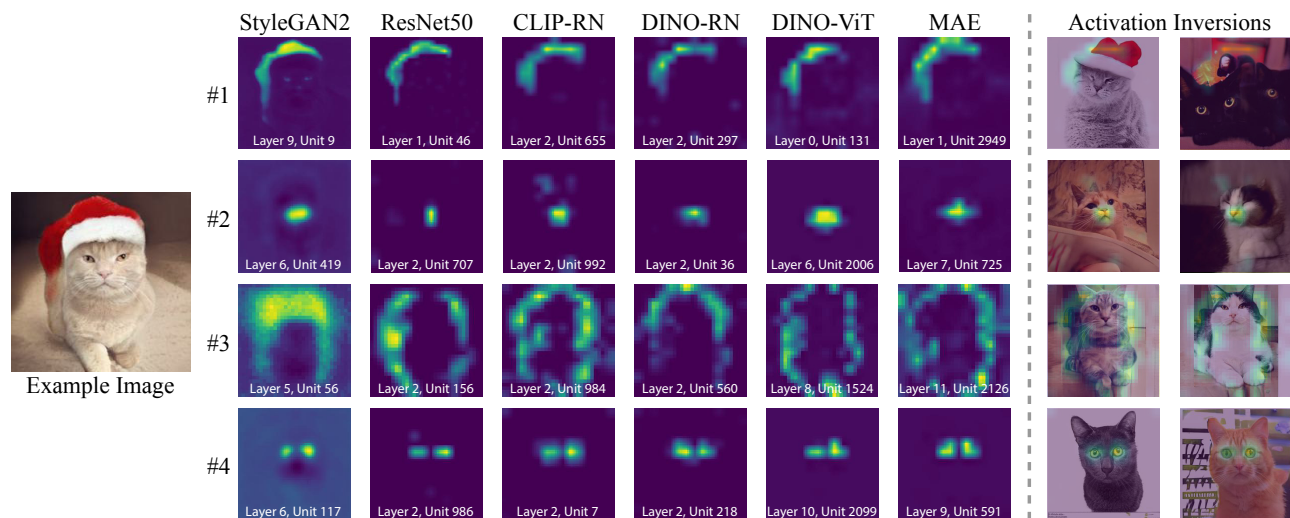# Rosetta Neurons: Mining the Common Units in a Model Zoo

Amil Dravid*
Northwestern

Yossi Gandelsman*
UC Berkeley

Alexei A. Efros
UC Berkeley

Assaf Shocher
UC Berkeley, Google

Figure 1: Mining for "Rosetta Neurons." Our findings demonstrate the existence of matching neurons across different models that express a shared concept (such as object contours, object parts, and colors). These concepts emerge without any supervision or manual annotations. We visualize the concepts with heatmaps and a novel inversion technique (two right columns).

Abstract

Do different neural networks, trained for various vision tasks, share some common representations? In this paper, we demonstrate the existence of common features we call "Rosetta Neurons" across a range of models with different architectures, different tasks (generative and discriminative), and different types of supervision (class-supervised, text-supervised, self-supervised). We present an algorithm for mining a dictionary of Rosetta Neurons across several popular vision models: Class Supervised-ResNet50, DINO-ResNet50, DINO-ViT, MAE, CLIP-ResNet50, Big-GAN, StyleGAN-2, StyleGAN-XL. Our findings suggest that certain visual concepts and structures are inherently embedded in the natural world and can be learned by different models regardless of the specific task or architecture, and without the use of semantic labels. We can visualize shared concepts directly due to generative models included in our analysis. The Rosetta Neurons facilitate model-to-model translation enabling various inversion-based manipulations, including cross-class alignments, shifting, zooming, and more, without the need for specialized training.

1. Introduction

One of the key realizations of modern machine learning is that models trained on one task end up being useful for many other, often unrelated, tasks. This is evidenced by the success of backbone pretrained networks and self-supervised training regimes. In computer vision, the prevailing theory is that neural network models trained for various vision tasks tend to share the same concepts and structures because they are inherently present in the visual world. However, the precise nature of these shared elements and the technical mechanisms that enable their transfer remain unclear.

In this paper, we seek to identify and match units that express similar concepts across different models. We call them *Rosetta* [1] *Neurons* (see fig. 1). How do we find them, considering it is likely that each model would express them differently? Additionally, neural networks are usually overparameterized, which suggests that multiple neurons can

---

* Equal contribution.

Project page, code and models: `https://yossigandelsman.github.io/rosetta_neurons`

[1]The Rosetta Stone is an ancient Egyptian artifact, a large stone inscribed with the same text in three different languages. It was the key to deciphering Egyptian hieroglyphic script. The original stone is on public display at the British Museum in London.
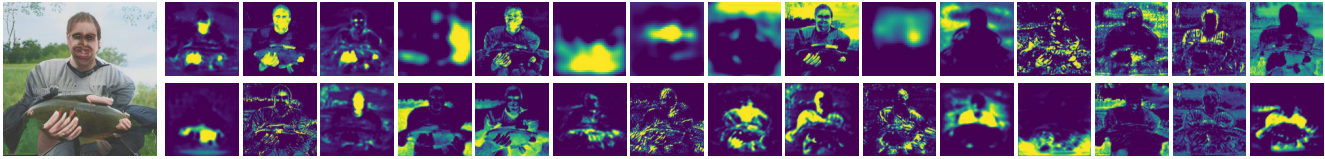
Figure 2: Visualization of all the concepts for one class. An example of the set of all concepts emerging for ImageNet "Tench" class by matching the five discriminative models from Table 2 and clustering within StyleGAN-XL. GAN heatmaps are visualized over one generated image.

express the same concept (synonyms). The layer and channel that express the concept would also differ between models. Finally, the value of the activation is calibrated differently in each. To address these challenges, we carefully choose the matching method we use. We found that post ReLU/GeLU values tend to produce distinct activation maps, thus these are the values we match. We compare units from different layers between the models while carefully normalizing the activation maps to overcome these differences. To address synonym neurons, we also apply our matching method on a model with itself and cluster units together according to the matches.

We search for Rosetta Neurons across eight different models: Class Supervised-ResNet50 [13], DINO-ResNet50, DINO-ViT [4], MAE [12], CLIP-ResNet50 [24], BigGAN [3], StyleGAN-2 [15], StyleGAN-XL [29]. We apply the models to the same dataset and correlate different units of different models. We mine the Rosetta neurons by clustering the highest correlations. This results in the emergence of model-free global representations, dictated by the data.

Fig. 2 shows an example image and all the activation maps from the discovered Rosetta Neurons. The activation maps include semantic concepts such as the person's head, hand, shirt, and fish as well as non-semantic concepts like contour, shading, and skin tone. In contrast to the celebrated work of Bau *et al*. on Network Dissection [2, 1], our method does not rely on human annotations or semantic segmentation maps. Therefore, we allow for the emergence of non-semantic concepts.

The Rosetta Neurons allow us to translate from one model's "language" to another. One particularly useful type of model-to-model translation is from discriminative models to generative models as it allows us to easily visualize the Rosetta Neurons. By applying simple transformations to the activation maps of the desired Rosetta Neurons and optimizing the generator's latent code, we demonstrate realistic edits. Additionally, we demonstrate how GAN inversion from real image to latent code improves when the optimization is guided by the Rosetta Neurons. This can be further used for out-of-distribution inversion, which performs image-to-image translation using a regular latent-to-image GAN. All of these edits usually require specialized training (e.g. [8, 14, 38]), but we leverage the Rosetta Neurons to

perform them with a fixed pre-trained model.

The contributions of our paper are as follows:

- We show the existence of Rosetta Neurons that share the same concepts across different models and training regimes.

- We develop a method for matching, normalizing, and clustering activations across models. We use this method to curate a dictionary of visual concepts.

- The Rosetta Neurons enables model-to-model translation that bridges the gap between representations in generative and discriminative models.

- We visualize the Rosetta Neurons and exploit them as handles to demonstrate manipulations to generated images that otherwise require specialized training.

2. Related Work

Visualizing deep representations. The field of interpreting deep models has been steadily growing, and includes optimizing an image to maximize the activations of particular neurons [36, 33, 22], gradient weighted activation maps [32, 23, 25, 30], nearest neighbors of deep feature representations [20], etc. The seminal work of Bau *et al*. [1, 2] took a different approach by identifying units that have activation maps highly correlated with semantic segments in corresponding images, thereby reducing the search space of meaningful units. However, this method necessitates annotations provided by a pre-trained segmentation network or a human annotator and is confined to discovering explainable units from a predefined set of classes and in a single model. Whereas all previous works focused on analyzing a single, specific neural network model, the focus of our work is in capturing commonalities across many different networks. Furthermore, unlike [2, 1], our method does not require semantic annotation.

Explaining discriminative models with generative models. GANAlyze [10] optimized the latent code of a pretrained GAN to find directions that affect a classifier decision. Semantic Pyramid [31] explored the subspaces of generated images to which the activations of a classifier are invariant. Lang *et al*. [21] trained a GAN to explain attributes that underlie classifier decisions. In all of these cases, the
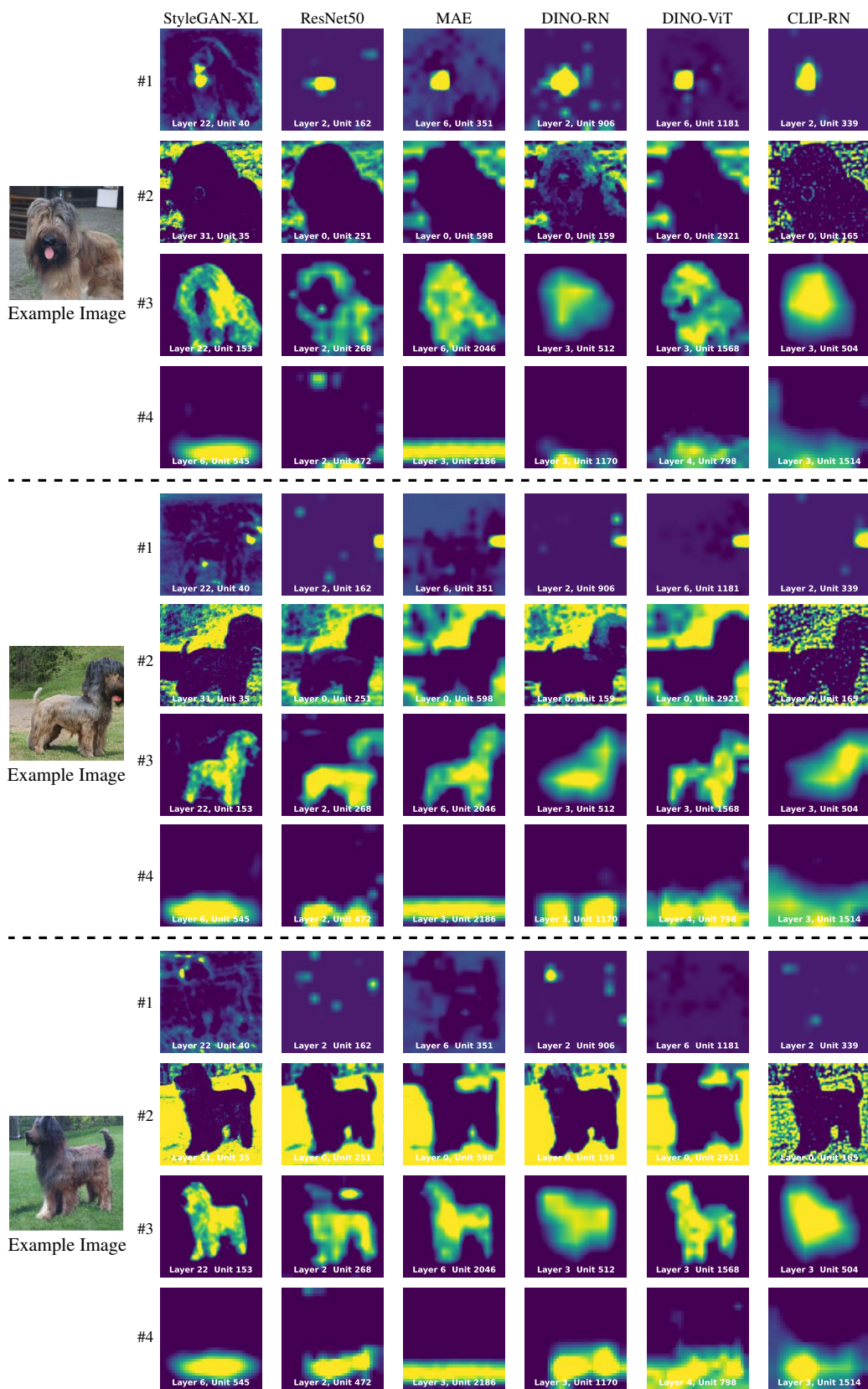
Figure 3: Rosetta Neuron Dictionary. A sample from the dictionary curated for the ImageNet class "Briard". The full dictionary can be found in the supplementary material. The figure presents 4 emergent concepts demonstrated in 3 example images. For each model, we present the normalized activation maps of the Rosetta Neuron matching the shared concept.
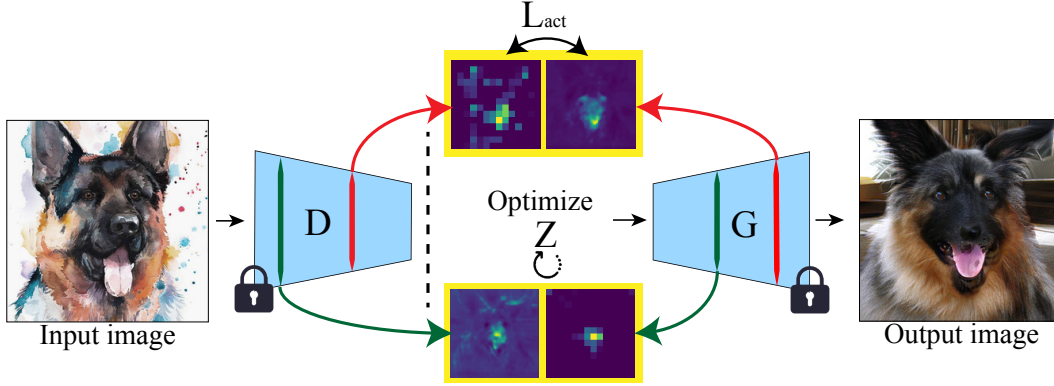
Figure 4: Rosetta Neurons guided image inversion. An input image is passed through a discriminative model D (i.e.: DINO) to obtain the Rosetta Neurons' activation maps. Then, the latent code Z of the generator is optimized to match those activation maps, according to the extracted pairs.

point where the generative and discriminative models communicate is in the one "language" they both speak - pixels; which is the output of the former and an input of the latter. Our method for bridging this gap takes a more straightforward approach: we directly match neurons from pre-trained networks and identify correspondences between their internal activations. Moreover, as opposed to [21] and [31], our method does not require GAN training and can be applied to any off-the-shelf GAN and discriminative model.

Analyzing representation similarities in neural networks. Our work is inspired by the neuroscience literature on representational similarity analysis [18, 7] that aims to extract correspondences between different brain areas [11], species [19], individual subjects [5], and between neural networks and brain neural activities [34]. On the computational side, Kornblith *et al.* [17] aimed to quantify the similarities between different layers of discriminative convolutional neural networks, focusing on identifying and preserving invariances. Esser, Rombach, and Ommer [9, 28] trained an invertible network to translate non-local concepts, expressed by a latent variable, across models. In contrast, our findings reveal that individual neurons hold shared concepts across a range of models and training regimes without the need to train a specialized network for translation. This leads to another important difference: the concepts we discover are local and have different responses for different spatial locations in an image. We can visualize these responses and gain insights into how these concepts are represented in the network.

3. Method

Our goal is to find Rosetta Neurons across a variety of models. We define Rosetta Neurons as two (or more) neurons in different models whose activations (outputs) are positively correlated over a set of many inputs. Below we explain how to find Rosetta Neurons across a variety of models and describe how to merge similar Rosetta Neurons

into clusters that represent the same concepts.

3.1. Mining common units in two models

Preliminaries. Given two models $F^{(1)}, F^{(2)}$, we run n inputs through both models. For discriminative models, this means a set of images $\{I_i\}_{i=1}^n$. If one of the models is generative, we first sample n random input noises $\{Z_i\}_{i=1}^n$ and generate images $I_i = F^{(1)}(z_i)$ that will be the set of inputs to the discriminative model $F^{(2)}$. We denote the set of extracted activation maps of F by F^{act}. The size $|F^{act}|$ is the total number of channels in all the layers. The j-th intermediate activation map of F when applied to the i-th input is then F_i^j. That is $F_i^j = F^j(I_i)$ for a discriminative model and $F_i^j = F^j(z_i)$ for a generative one.

Comparing activation maps. To compare units $F^{(1)j}$ and $F^{(2)k}$, namely, the j-th unit from the first model with the k-th unit from the second one, we first bilinearly interpolate the feature maps to have the same spatial dimensions according to the maximum of the two map sizes. Our approach to perform matching is based on correlation, similar to [18], but taken across both data instances and spatial dimensions. We then take the mean and variance across the n images and across the spatial dimensions of the images, where x combines both spatial dimensions of the images.

$$\overline{F^j} = \frac{1}{nm^2} \sum_{i,x} F_{i,x}^j$$
$$var(F^j) = \frac{1}{nm^2 - 1} \sum_{i,x} \left(F_{i,x}^j - \overline{F^j} \right)^2 \tag{1}$$

Next, the measure of distance between two units is calculated by Pearson correlation:

$$d(F^{(1)j}, F^{(2)k}) = \frac{\sum\limits_{i,x} \left(F_{i,x}^{(1)j} - \overline{F^{(1)j}} \right) \left(F_{i,x}^{(2)k} - \overline{F^{(2)k}} \right)}{\sqrt{var(F^{(1)j}) \cdot var(F^{(2)k})}} \tag{2}$$

In our experiments, this matching is computed between a generative model G and a discriminative model D. The

images used for D are generated by G applied to n sampled noises.

Filtering "best buddies" pairs. To detect reliable matches between activation maps, we keep the pairs that are mutual nearest neighbors (named "best-buddies" pairs by [6]) according to our distance metric and filter out any other pair. Formally, our set of "best buddies" pairs is:

$$BB(F^{(1)}, F^{(2)}; K) = \{(j, k)|$$

$$F^{(1)k} \in KNN(F^{(2)j}, F^{(1)act}; K) \qquad (3)$$

$$\wedge \ F^{(2)j} \in KNN(F^{(1)k}, F^{(2)act}; K)\}$$

Where $KNN(F^{(a)j}, F^{(b)act})$ is the set of the K-nearest neighbors of the unit j from model $F^{(a)}$ among all the units in model $F^{(b)}$:

$$KNN(F^{(a)j}, F^{(b)act}; K) = \underset{q_1 \dots q_K \subseteq F^{(b)act}}{\arg\min} \sum_{k=1}^{K} d(F^{(a)j}, q_k)$$

As shown in [6], the probability of being mutual nearest neighbors is maximized when the neighbors are drawn from the same distribution. Thus, keeping the "best buddies" discards noisy matches.

3.2. Extracting common units in m models

Merging units between different models. To find similar activation maps across many different discriminative models $D_i, i \in [m]$, we merge the "best buddies" pairs calculated between D_i and a generator G for all the i's. Formally, our Rosetta units are:

$$R(G, D_1 ... D_m) = \{(j, k_1, ..., k_m)|\forall i : (j, k_i) \in BB(G, D_i)\}$$
$$(4)$$

This set of tuples includes the "translations" between similar neurons across all the models. Note that when $m = 1$, $R(G, D_1) = BB(G, D_1)$.

Clustering similar units into concepts. Empirically, the set of Rosetta units includes a few units that have similar activation maps for the n images. For instance, multiple units may be responsible for edges or concepts such as "face." We cluster them according to the self "best-buddies" of the generative model, defined by $BB(G, G; K)$. We set two Rosetta Neurons in R to belong to the same cluster if their corresponding units in G are in $BB(G, G; K)$.

Curating a dictionary. After extracting matching units for a dataset across a model zoo, we enumerate the sets of matching Rosetta Neurons in the clustered R. Fig. 3 is a sample from such a dictionary. Fig. 2 shows a list of all the concepts for a single image. Since the concepts emerge and are not related to human annotated labels, we simply enumerate them and present each concept on several example images to visually identify it. Using 1600 instances generated by the GAN, Distances are taken between all possible bipartite pairs of units, the $K = 5$ nearest neighbors are extracted, from which Best-Buddies are filtered. Typically for the datasets and models we experimented with, around 50 concepts emerge. The exact list of models used in our experiments and the datasets they were trained on can be found in Table. 2. See supplementary material for the dictionaries.

Figure 5: Out-of-distribution inversions. By incorporating the Rosetta Neurons in the image inversion process, we can invert sketches and cartoons (first row), and generate similar in-distribution images (last row). A subset of the Rosetta Neurons from the input images that were matched during the inversion process is shown in the middle rows.

4. Visualizing the Rosetta Neurons

As we involve a generative model in the Rosetta Neurons mining procedure, we can utilize it for visualizing the discovered neurons as well. In this section, we present how to visualize the neurons via a lightweight matches-guided inversion technique. We then present how direct edits of the activation maps of the neurons can translate into a variety of generative edits in the image space, without any generator modification or re-training.

4.1. Rosetta Neurons-Guided Inversion

To visualize the extracted Rosetta Neurons, we take inspiration from [31], and use the generative model G to produce images for which the generator activation maps of the Rosetta Neurons best match to the paired activation maps extracted from $D(I_v)$, as shown in figure 4. As opposed to [31], we do not train the generative model to be conditioned on the activation maps. Instead, we invert images through the fixed generator into some latent code z, while maximizing the similarity between the activation maps of the paired Rosetta Neurons. Our objective is:

$$\arg\min_{z}(-L_{act}(z, I_v) + \alpha L_{reg}(z)) \qquad (5)$$

Where α is a loss coefficient, L_{reg} is a regularization term (L_2 or L_1), and $L_{act}(z, I_v)$ is the mean of normalized sim-

ilarities between the paired activations:

$$L_{act}(z, I_v) =$$

$$\frac{1}{|BB(G,D)|} \sum_{\substack{(j,k) \in \\ BB(G,D)}} \frac{\sum_x \left(G_x^j - \overline{G^j}\right)\left(D_x^k - \overline{D^k}\right)}{\sqrt{var(G^j) \cdot var(D^k)}} \quad (6)$$

Where G^j is the j-th activation map of $G(z)$ and D^k is the k-th activation map of $D(I_v)$. For obtaining this loss, we use the mean and variance precomputed by Eq. 1 over the entire dataset during the earlier mining phase. However, we calculate the correlation over the spatial dimensions of a single data instance.

The Rosetta neurons guided inversion has two typical modes. The first mode is when both the initial activation map and the target one have some intensity somewhere in the map (e.g. two activation maps that are corresponding to "nose" are activated in different spacial locations). In this case, the visual effect is an alignment between the two activation maps. As many of the Rosetta neurons capture object parts, it results in image-to-image alignment (e.g., fig. 6). The second mode is when either the target or the initial activation map is not activated. In this case, a concept will appear or disappear (e.g., fig. 9).

Visualizing a single Rosetta Neuron. We can visualize a single Rosetta Neuron by modifying the loss in our inversion process (eq. 6). Rather than calculating the sum over the entire set of Rosetta Neurons, we do it for a single pair that corresponds to the specific Rosetta neuron. When this optimization procedure is applied a few times on the same input neuron pair starting from a few different randomly initialized latent codes, we get a diverse set of images that are matching to the same activation map of the wanted Rosetta Neuron. This allows a user to disentangle and detect what is the concept that is specifically represented by the given neuron. Figure 1 present two optimized images for each of the presented Rosetta Neurons. This visualization allows the viewer to see that Concept #1 corresponds to the concept "red color," rather than to the concept "hat."

Inverting out-of-distribution images. The inversion process presented above does not use the generated image in the optimization, as opposed to common inversion techniques that calculate the pixel loss or perceptual loss between the generated image the input image. Our optimization process does not compare the image pixel values, and as many of the Rosetta Neurons capture high-level semantic concepts and coarse structure of the image, this allows us to invert images outside of the training distribution of the generative model. Figure 6 presents a cross-class image-to-image translation that is achieved by Rosetta Neurons guided inversion. As shown, the pose of the input images of dogs is transferred to the poses of the optimized cat images,

Figure 6: Cross-class image-to-image translation. Rosetta Neurons guided inversion of input images (top row) into a StyleGAN2 trained on LSUN cats [35], allows us to preserve the pose of the animal while changing it from dog to cat (bottom row). See supplementary material for more examples.

as the Rosetta Neurons include concepts such as "nose," "ears," and "contour" (please refer to Figure 1 for a subset of the Rosetta Neurons for this set of models).

Figure 5 presents the inversion results for sketches and cartoons, and a subset of the Rosetta Neurons that were used for optimization. As shown, the matches-guided inversion allows us to "translate" between the two domains via the shared Rosetta Neurons and preserve the scene layout and object pose. Our lightweight method does not require dedicated models or model training, as opposed to [38, 14].

Inverting in-distribution images. We found that adding the loss term in eq. 5 to the simple reconstruction loss objective improves the inversion quality. Specifically, we optimize:

$$\arg\min_z (L_{rec}(G(z), I_v) + \alpha L_{reg}(z) - \beta L_{act}(z, I_v)) \quad (7)$$

Where L_{rec} is the reconstruction loss between the generated image and the input image, and β is a loss coefficient. The reconstruction loss can be pixel loss, such as L_1 or L_2 between the two images, or a perceptual loss.

We compare the inversion quality with and without the Rosetta Neurons guidance and present the PSNR, SSIM, and LPIPS [37] for StyleGAN-XL inversion. We use solely a perceptual loss as a baseline, similarly to [29]. We add our loss term to the optimization, where the Rosetta Neurons are calculated from 3 sets of matches with StyleGAN-XL: matching to DINO-RN, matching to CLIP-RN, and matching across all the discriminative models in Table 2. We use the same hyperparameters as in [29], and set $\alpha = 0.1$ and $\beta = 1$.

Table 1 presents the quantitative inversion results for 5000 randomly sampled images from the ImageNet validation set (10% of the validation set, 5 images per class), as done in [29]. Figure 7 presents the inversion results for the baseline and for the additional Rosetta Neurons guidance using the matches between all the models. As shown qualitatively and quantitatively, the inversion quality improves

	PSNR ↑	SSIM ↑	LPIPS ↓
Perceptual loss	13.99	0.340	0.48
+DINO matches	15.06	0.360	0.45
+CLIP matches	15.20	0.362	**0.44**
+All matches	**15.42**	**0.365**	0.46

Table 1: Inversion quality on ImageNet. We compare the inversion quality for StyleGAN-XL when Rosetta Neurons guidance is added, for 3 sets of matches - StyleGAN-XL & DINO-RN, StyleGAN-XL & CLIP-RN and all the models from figure 3.

Model	Training dataset	Resolution
StyleGAN-XL	ImageNet	256
StyleGAN2	LSUN(cat)	256
StyleGAN2	LSUN(horse)	512
BigGAN	ImageNet	256
ResNet50	ImageNet	224
DINO-ResNet50	ImageNet	224
DINO-VIT-base	ImageNet	224
MAE-base	ImageNet	224
CLIP	WebImageText	224

Table 2: Models used in the paper.

when the Rosetta Neurons guiding is added. We hypothesize this is due to the optimization objective that directly guides the early layers of the generator and adds layout constraints. These soft constraints reduce the optimization search space and avoid convergence to local minima with low similarity to the input image.
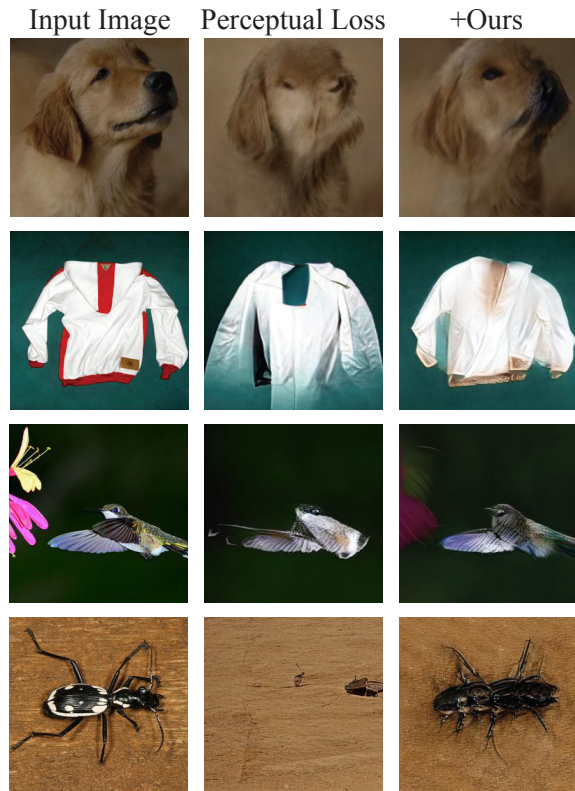
4.2. Rosetta Neurons Guided Editing

The set of Rosetta Neurons allows us to apply controlled edits on a generated image $I_{src} = G(z)$ and thus to provide a counterfactual explanation to the neurons. Specifically, we modify the activation maps corresponding to the Rosetta Neurons, extracted from $G(z)$, and re-optimize the latent code to match the edited activation maps according to the same optimization objective presented in eq. 5. As opposed to previous methods like [8], which trained a specifically designed generator to allow disentangled manipulation of objects at test-time, we use a fixed generator and only optimize the latent representation. Next, we describe the different manipulation that can be done on the activation maps, before re-optimizing the latent code:

Zoom-in. We double the size of each activation map that corresponds to a Rosetta Neurons with bilinear interpolation and crop the central crop to return to the original activation map size. We start our re-optimization from the same latent code that generated the original image.

Shift. To shift the image, we shift the activation maps directly and pad them with zeros. The shift stride is relative to the activation map size (e.g. we shift a 4×4 activation map by 1, while shifting 8×8 activation maps by 2).

Copy & paste. We shift the activation maps twice into two directions (e.g. left and right), creating two sets of activation maps - left map, and right map. We merge them by copying and pasting the left half of the left activation map

Figure 7: Image inversions for StyleGAN-XL. We compare inversions by optimizing perceptual loss only (second column), to additional Rosetta Neurons guidance loss, with matches calculated across all the models presented in Figure 3 (third column). See supplementary material for more examples.

and the right half of the right activation map. We found that starting from random z rather than z that generated the original image obtains better results.

Figure 8 shows the different image edits that are done via latent optimization to match the manipulated Rosetta Neurons. We apply the edits for two different generative models (BigGAN and StyleGAN2) to show the robustness of the method to different architectures.

Fine-grained Rosetta Neurons edit. Our optimization procedure allows us to manipulate a subset of the Rosetta Neurons, instead of editing all of the neurons together. Specifically, we can manually find among the Rosetta Neurons a few that correspond to elements in the image that we wish to modify. We create "ground truth" activations by modifying them manually and re-optimizing the latent code to match them. For example - to remove concepts specified by Rosetta Neurons, we set their values to the minimal value in their activation maps. We start our optimization from the latent that corresponds to the input image and optimize until the picked activation maps converge to the manually edited activation maps. Figure 9 presents examples of removed Rosetta Neurons. Modifying only a few activation maps (1 or 2 in the presented images) that correspond to the objects

Input Image	Zoom In	Shift	Copy & Paste

Figure 8: Rosetta Neurons guided editing. Direct manipulations on the activation maps corresponding to the Rosetta neurons are translated to manipulations in the image space. We use two models (top row - StyleGAN2, bottom two rows - BigGAN) and utilize the matches between each of them to DINO-RN.

Figure 9: Single Rosetta Neurons Edits. We optimize the latent input s.t. the value of a desired Rosetta activation reduces. This allows removing elements from the image (e.g. emptying the beer in the glass, reducing the water stream in the fountain, and removing food from a plate). See appendix for more examples.

we aimed to remove, allows us to apply realistic manipulations in the image space. As opposed to [2], we do not rewrite the units in the GAN directly and apply optimization instead, as we found that direct edits create artifacts in the generated image for large and diverse GANs.

Implementation details. For the re-optimization step, we train z for 500 steps, with Adam optimizer [16] and a learning rate of 0.1 for StyleGAN2 and 0.01 for BigGAN. Following [29], the learning rate is ramped up from zero linearly during the first 5% of the iterations and ramped down to zero using a cosine schedule during the last 25% of the iterations. We use $K = 5$ for calculating the nearest neighbors. The inversion and inversion-based editing take less than 5 minutes per image on one A100 GPU.

5. Limitations

Our method can not calculate GAN-GAN matches directly, only through a discriminative model. Unlike discriminative models that can receive the same input image, making two GANs generate the same image is not straightforward. Consequently, we only match GANs with discriminative models.

Secondly, we were unsuccessful when applying our approach to diffusion models, such as [27]. We speculate that this is due to the autoregressive nature of diffusion models, where each step is a conditional generative model from image to image. We hypothesize that as a result, the noisy image input is a stronger signal in determining the outcome of each step, rather than a specific unit. Thus, the units in diffusion models have more of an enhancing or editing role, rather than a generating role, which makes it less likely to identify a designated perceptual neuron.

Lastly, our method relies on correlations, and therefore there is a risk of mining spurious correlations. As shown in Figure 3, the dog in the third example does not have its tongue visible, yet both StyleGAN-XL and DINO-RN activated for Concept #1 in a location where the tongue would typically be found. This may be due to the correlation between the presence of a tongue and the contextual information where it usually occurs.

6. Conclusion

We introduced a new method for mining and visualizing common representations that emerge in different visual models. Our results demonstrate the existence of specific units that represent the same concepts in a diverse set of deep neural networks, and how they can be utilized for various generative tasks via a lightweight latent optimization process. We believe that the found common neurons can be used in a variety of additional tasks, including image retrieval tasks and more advanced generative tasks. Additionally, we hope that the extracted representations will shed light on the similarities and dissimilarities between models that are trained for different tasks and with different architectures. We plan to explore this direction in future work.

Acknowledgements

References

[1] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.

[2] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[5] Andrew C. Connolly, J. Swaroop Guntupalli, Jason D. Gors, Michael Hanke, Yaroslav O. Halchenko, Yu-Chien Wu, Hervé Abdi, and James V. Haxby. The representation of biological classes in the human brain. *The Journal of Neuroscience*, 32:2608 – 2618, 2012.

[6] Tali Dekel, Shaul Oron, Michael Rubinstein, Shai Avidan, and William T. Freeman. Best-buddies similarity for robust template matching. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2021–2029, 2015.

[7] Shimon Edelman. Representation is representation of similarities. *Behavioral and Brain Sciences*, 21(4):449–467, 1998.

[8] Dave Epstein, Taesung Park, Richard Zhang, Eli Shechtman, and Alexei A. Efros. Blobgan: Spatially disentangled scene representations. *European Conference on Computer Vision (ECCV)*, 2022.

[9] Patrick Esser, Robin Rombach, and Björn Ommer. A disentangling invertible interpretation network for explaining latent representations. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9220–9229, 2020.

[10] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. *arXiv preprint arXiv:1906.10112*, 2019.

[11] James Haxby, Maria Gobbini, Maura Furey, Alumit Ishai, Jennifer Schouten, and Pietro Pietrini. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science (New York, N.Y.)*, 293:2425–30, 10 2001.

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021.

[13] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[15] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[17] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. *ArXiv*, abs/1905.00414, 2019.

[18] Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2, 2008.

[19] Nikolaus Kriegeskorte, Marieke Mur, Douglas A. Ruff, Roozbeh Kiani, Jerzy Bodurka, Hossein Esteky, Keiji Tanaka, and Peter A. Bandettini. Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron*, 60:1126–1141, 2008.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

[21] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T. Freeman, Phillip Isola, Amir Globerson, Michal Irani, and Inbar Mosseri. Explaining in style: Training a gan to explain a classifier in stylespace. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 673–682, 2021.

[22] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. https://distill.pub/2020/circuits/zoom-in.

[23] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[25] Sylvestre-Alvise Rebuffi, Ruth Fong, Xu Ji, and Andrea Vedaldi. There and back again: Revisiting backpropagation saliency methods. *CoRR*, abs/2004.02866, 2020.

[26] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021.

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[28] Robin Rombach, Patrick Esser, and Björn Ommer. Network-to-network translation with conditional invertible neural networks. *arXiv: Computer Vision and Pattern Recognition*, 2020.

[29] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIG-GRAPH 2022 Conference Proceedings*, SIGGRAPH '22, New York, NY, USA, 2022. Association for Computing Machinery.

[30] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336–359, 2016.

[31] Assaf Shocher, Yossi Gandelsman, Inbar Mosseri, Michal Yarom, Michal Irani, William T. Freeman, and Tali Dekel. Semantic pyramid for image generation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7455–7464, 2020.

[32] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[33] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[34] Daniel Yamins, Ha Hong, Charles Cadieu, Ethan Solomon, Darren Seibert, and James Dicarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 111, 05 2014.

[35] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[36] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, 2013.

[37] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[38] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.