WHAT DO VISION TRANSFORMERS LEARN? A VISUAL EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision transformers (ViTs) are quickly becoming the de-facto architecture for computer vision, yet we understand very little about why they work and what they learn. While existing studies visually analyze the mechanisms of convolutional neural networks, an analogous exploration of ViTs remains challenging. In this paper, we first address the obstacles to performing visualizations on ViTs. Assisted by these solutions, we observe that neurons in ViTs trained with language model supervision (e.g., CLIP) are activated by semantic concepts rather than visual features. We also explore the underlying differences between ViTs and CNNs, and we find that transformers detect image background features, just like their convolutional counterparts, but their predictions depend far less on high-frequency information. On the other hand, both architecture types behave similarly in the way features progress from abstract patterns in early layers to concrete objects in late layers. In addition, we show that ViTs maintain spatial information in all layers except the final layer. In contrast to previous works, we show that the last layer most likely discards the spatial information and behaves as a learned global pooling operation.

1 INTRODUCTION

Recent years have seen the rapid proliferation of vision transformers (ViTs) across a diverse range of tasks from image classification to semantic segmentation to object detection (Dosovitskiy et al., 2020; He et al., 2021; Dong et al., 2021; Liu et al., 2021; Zhai et al., 2021; Dai et al., 2021). Despite their enthusiastic adoption and the constant introduction of architectural innovations, little is known about the inductive biases or features they tend to learn. While feature visualizations and image reconstructions have provided a looking glass into the workings of CNNs (Olah et al., 2017; Zeiler & Fergus, 2014; Dosovitskiy & Brox, 2016), these methods have shown less success for understanding ViT representations, which are difficult to visualize. In this work we show that, if properly applied to the correct representations, feature visualizations can indeed succeed on VITs. This insight allows us to visually explore ViTs and the information they glean from images.

In order to investigate the behaviors of vision transformers, we first establish a visualization framework that incorporates improved techniques for synthesizing images that maximally activate neurons.

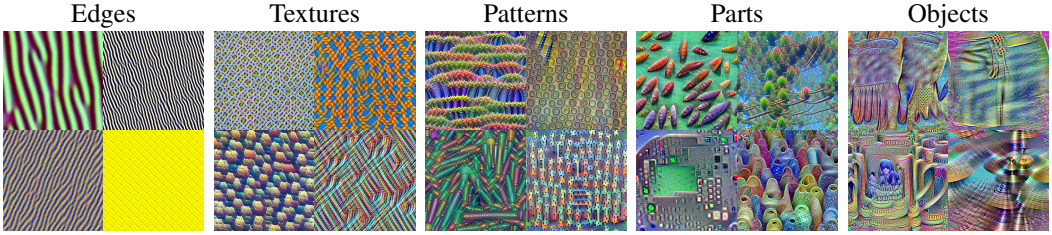


Figure 1: **The progression for visualized features of ViT B-32.** Features from early layers capture general edges and textures. Moving into deeper layers, features evolve to capture more specialized image components and finally concrete objects.

Through dissecting and visualizing the internal representations in the transformer architecture, we find that patch tokens preserve spatial information, even in individual channels, throughout all layers except the last attention block. The last layer of ViTs learns a token-mixing operation akin to average pooling, such that the classification head exhibits comparable accuracy when ingesting a random token instead of the CLS token.

After probing the role of spatial information, we delve into the behavioral differences between ViTs and CNNs. When performing activation maximizing visualizations, we notice that ViTs consistently generate higher quality image backgrounds than CNNs. Thus, we try masking out image foregrounds during inference, and find that ViTs consistently outperform CNNs when exposed only to image backgrounds. These findings bolster the observation that transformer models extract information from many sources in an image to exhibit superior performance on out-of-distribution generalization (Paul & Chen, 2021) as well as adversarial robustness (Shao et al., 2021). Additionally, convolutional neural networks are known to rely heavily on high-frequency texture information in images (Geirhos et al., 2018). In contrast, we find that ViTs perform well even when high-frequency content is removed from their inputs.

We further visualize the effects of language model supervision, i.e. CLIP (Radford et al., 2021), on the features extracted by vision transformers. While both ImageNet-trained ViTs and CLIP-trained vision transformers possess neurons that are activated by visual features (e.g. shapes and colors) and distinct classes, the neurons of CLIP-trained vision transformers are also activated by features that do not represent physical objects, such as visual characteristics relating to parts of speech (e.g. epithets, adjectives, and prepositions) or broader concepts such as morbidity.

Our contributions are summarized as follows:

I. We observe that uninterpretable and adversarial behavior occurs when applying standard methods of feature visualization to the relatively low-dimensional components of transformer-based models, such as attention heads, keys, queries, scores, or values. However, applying these tools to the relatively high-dimensional features of the position-wise feedforward layer results in successful and informative visualizations.

II. We show that patch-wise image activation patterns for ViT features essentially behave like saliency maps, highlighting the regions of the image a given feature attends to. This behavior persists even for relatively deep layers, showing the model preserves the positional relationship between patches instead of using them as global information stores.

III. We compare the behavior of ViTs and CNNs, finding that ViTs make better use of background information and rely less on high-frequency, textural attributes. Both types of networks build progressively more complex representations in deeper layers and eventually contain features responsible for detecting distinct objects.

IV. We investigate the effect of natural language supervision with CLIP on the types of features extracted by ViTs. We find CLIP-trained models include various features clearly catered to detecting components of images corresponding to caption text, such as prepositions, adjectives, and conceptual categories.

2 Related Work

2.1 Optimization-Based Visualization

One approach to understanding what models learn during training is using gradient descent to produce an image which conveys information about the inner workings of the model. This has proven to be a fruitful line of work in the case of understanding CNNs specifically. The basic strategy underlying this approach is to optimize over input space to find an image which maximizes a particular attribute of the model. For example, Erhan et al. (2009) use this approach to visualize images which maximally activate specific neurons in early layers of a network, and Olah et al. (2017) extend this to neurons, channels, and layers throughout a network. Simonyan et al. (2014); Yin et al. (2020) produce images which maximize the score a model assigns to a particular class. Mahendran & Vedaldi (2015) apply a similar method to invert the feature representations of particular image examples.

Recent work Ghiasi et al. (2021) has studied techniques for extending optimization-based class visualization to ViTs. We incorporate and adapt some of these proposed techniques into our scheme for feature visualization.

2.2 Other Visualization Approaches

Aside from optimization-based methods, many other ways to visualize CNNs have been proposed. Dosovitskiy & Brox (2016) train an auxiliary model to invert the feature representations of a CNN. Zeiler & Fergus (2014) use 'deconvnets' to visualize patches which strongly activate features in various layers. Simonyan et al. (2014) introduce saliency maps, which use gradient information to identify what parts of an image are important to the model's classification output. Zimmermann et al. (2021) demonstrate that natural image samples which maximally activate a feature in a CNN may be more informative than generated images which optimize that feature. We draw on some aspects of these approaches and find that they are useful for visualizing ViTs as well.

2.3 Understanding ViTs

Given their rapid proliferation, there is naturally great interest in how ViTs work and how they may differ from CNNs. Although direct visualization of their features has not previously been explored, there has been recent progress in analyzing the behavior of ViTs. Paul & Chen (2021); Naseer et al. (2021); Shao et al. (2021) demonstrate that ViTs are inherently robust to many kinds of adversarial perturbations and corruptions. Raghu et al. (2021) compare how the internal representation structure and use of spatial information differs between ViTs and CNNs. Chefer et al. (2021) produce 'image relevance maps' (which resemble saliency maps) to promote interpretability of ViTs.

3 ViT Feature Visualization

Like many visualization techniques, we take gradient steps to maximize feature activations starting from random noise (Olah et al., 2017). To improve the quality of our images, we penalize total variation (Mahendran & Vedaldi, 2015), and also employ the Jitter augmentation (Yin et al., 2020), the ColorShift augmentation, and augmentation ensembling (Ghiasi et al., 2021). Finally, we find that Gaussian smoothing facilitates better visualization in our experiments as is common in feature visualization (Smilkov et al., 2017; Cohen et al., 2019).

Each of the above techniques can be formalized as follows. A ViT represents each patch p (of an input x) at layer l by an array $A_{l,p}$ with d entries. We define a feature vector f to be a stack composed of one entry from each of these arrays. Let $f_{l,i}$ be formed by concatenating the ith entry in $A_{l,p}$ for all patches p. This vector f will have dimension equal to the number of patches. The optimization objective starts by maximizing the sum of the entries of f over inputs x. The main loss is then

$$\mathcal{L}_{\text{main}}(x, l, i) = \sum_p (f_{l,i})_p. \tag{1}$$

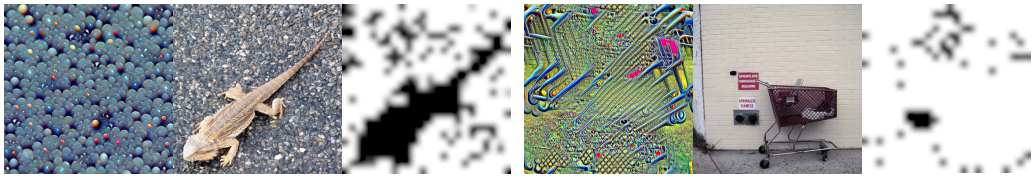
We employ total variation regularization by adding the term $\lambda TV(x)$ to the objective. TV represents the total variation, and λ is the hyperparameter controlling the strength of its regularization effect. We can ensemble augmentations of the input to further improve results. Let $\mathcal{A}$ define a distribution of augmentations to be applied to the input image x, and let a be a sample from $\mathcal{A}$. To create a minibatch of inputs from a single image, we sample several augmentations $\{a_k\}$ from $\mathcal{A}$. Finally, the optimization problem is:

$$x^* = \arg\max_x \sum_k \mathcal{L}_{\text{main}}(a_k(x), l, i) + \lambda TV(a_k(x)). \tag{2}$$

We achieve the best visualizations when $\mathcal{A}$ is $GS(CS(Jitter(x)))$, where GS denotes Gaussian smoothing and CS denotes ColorShift, whose formulas are:

$$GS(x) = x + \epsilon; \;\; \epsilon \sim \mathcal{N}(0, 1)$$

$$CS(x) = \sigma x + \mu; \;\; \mu \sim \mathcal{U}(-1, 1); \;\; \sigma \sim e^{\mathcal{U}(-1,1)}.$$



(a) (b)

Figure 2: *(a)*: **Example feature visualization from ViT feed forward layer.** *Left:* Image optimized to maximally activate a feature from layer 5. *Center:* Corresponding maximally activating ImageNet example. *Right:* The image's patch-wise activation map. *(b)*: **Feature from ViT-B16 most activated by shopping carts.**

Note that even though ϵ and μ are both additive noise, they act on the input differently since μ is applied per channel (i.e. has dimension three), and ϵ is applied per pixel. For more details on hyperparameters, refer to Appendix B.

To better understand the content of a visualized feature, we pair every visualization with the image from the ImageNet validation/train set that most strongly activates the relevant feature. Moreover, we plot the feature's activation pattern by passing the most activating image through the network and showing the resulting pattern of feature activations. Figure 2(a) is an example of such a visualization. From the left-most panel, we hypothesize that this feature corresponds to gravel. The most activating image from the validation set (middle) contains a lizard on a pebbly gravel road. Interestingly, the gravel background lights up in the activation pattern (right) while the lizard does not. The activation pattern in this example behaves like a saliency map (Simonyan et al., 2014), and we explore this phenomenon across different layers of the network further in Section 4.

The majority of the parameters in ViT models belong to the multi-layer transformer architecture within the ViT. The model we adopt for the majority of our visualizations is ViT-B16 implemented based on the work of Dosovitskiy et al. (2020). This model is composed of 12 blocks, each consisting of multi-headed attention layers, followed by a projection layer, and finally followed by a position-wise-feed-forward layer. For brevity, we henceforth refer to the position-wise-feed-forward layer simply as the feed-forward layer. In this model, every patch is always represented by a vector of size 768 except in the feed-forward layer which has a size of 3072 (4 times larger than other layers).

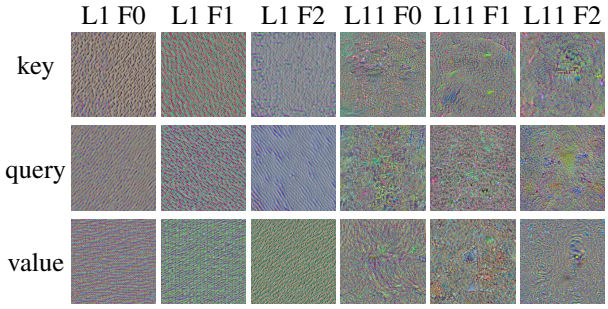
We first attempt to visualize features of the multi-headed attention layer, including visualization of the keys, queries, values, and scores, by performing activation maximization. We find that the visualized feed-forward features are significantly more interpretable than other layers. We attribute this difficulty of visualizing other layers to the property that ViTs pack a tremendous amount of information into only 768 features, (e.g. in keys, queries, and values) which then behave similar to multi-modal neurons, as discussed by Goh et al. (2021), due to many semantic concepts being encoded in a low dimensional space. Furthermore, we find that this behaviour is more extreme in deeper layers. See Figure 3 for examples of visualizations of keys, queries and values in both early and deep layers of the ViT. Inspired by these observations, we visualize the features within the feed-forward layer across all 12 blocks of the ViT. We refer to these blocks interchangeably as layers.

The feed-forward layer depicted in Figure 3 takes an input of size $d = 768$, projects it into a $t = 4$ times higher dimensional space, applies the non-linearity GELU, and then projects back to d dimensional space. Unless otherwise stated, we always visualize the output of the GELU layers in our experiments. We hypothesize that the network exploits these high-dimensional spaces to store relatively disentangled representations. On the other hand, compressing the features into a lower dimensional space may result in the jumbling of features, yielding uninterpretable visualizations.

As mentioned above, we pair every visualization with the image from the ImageNet validation or train set that maximally activates the corresponding feature (see Figure 2(a)).

4 Last-Layer Token Mixing

In this section, we investigate the preservation of patch-wise spatial information observed in the visualizations of patch-wise feature activation levels which, as noted before, bear some similarity to saliency maps. Figure 2(a) demonstrates this phenomenon in layer 5, where the visualized feature is



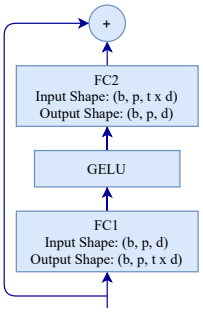


Figure 3: *Left*: **Visualization of key, query, and value.** The visualization both fails to extract interpretable features and to distinguish between early and deep layers. High-frequency patterns and adversarial behavior dominate. *Right*: **ViT feed forward layer**. The first linear layer increases the dimension of the feature space, and the second one brings it back to its initial dimension.

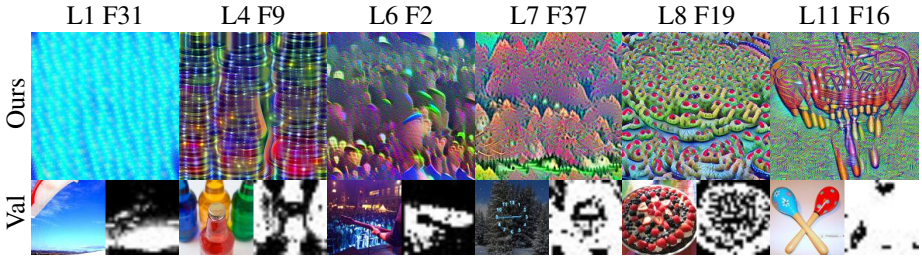


Figure 4: **Feature activation maps in internal layers can effectively segment the contents of an image with respect to a semantic concept.** For each image triple, the visualization on top shows the result of our method, the image on the bottom left is the most activating image from the validation set and the image on the bottom right shows the activation pattern.

strongly activated for almost all rocky patches but not for patches that include the lizard. Additional examples can be seen in Figure 4, where the activation maps approximately segment the image with respect to some relevant aspect of the image. We find it surprising that even though every patch *can* influence the representation of every other patch, these representations remain local, even for individual channels in deep layers in the network. While a similar finding for CNNs, whose neurons may have a limited receptive field, would be unsurprising, even neurons in the first layer of a ViT have a complete receptive field. In other words, ViTs *learn* to preserve spatial information, despite lacking the inductive bias of CNNs. Spatial information in patches of deep layers has been explored in Raghu et al. (2021) through the CKA similarity measure, and we further show that spatial information is in fact present in individual channels.

The last layer of the network, however, departs from this behavior and instead appears to serve a role similar to average pooling. Figure 2(b) shows one example of our visualizations for a feature from the last layer that is activated by shopping carts. The activation pattern is fairly uniform across the image. For classification purposes, ViTs use a fully connected layer applied only on the class token (the *CLS* token). It is possible that the network globalizes information in the last layer to ensure that the CLS token has access to the entire image, but because the CLS token is treated the same as every other patch by the transformer, this seems to be achieved by globalizing across *all* tokens.

Based on the feature activation patterns, we hypothesize that the CLS token plays a relatively minor role throughout the network and is not used for globalization until the last layer. To demonstrate this, we perform inference on images without using the CLS token in layers 1-11, meaning that in these layers, each patch only attends to other patches and not to the CLS token. At layer 12, we then insert a value for the CLS token so that other patches can attend to it and vice versa. This value is obtained by running a forward pass using only the CLS token and no image patches; this value is constant across all input images.

The resulting hacked network that only has CLS access in the last layer can still successfully classify 78.61% of the ImageNet validation set as shown in Table 1. From this result, we conclude that

Table 1: After the last layer, every patch contains the same information. "Isolating CLS" denotes the experiment where attention is only performed between patches before the final attention block, while "Patch Average" and "Patch Maximum" refer to the experiment in which the classification head is placed on top of individual patches without fine-tuning. Experiments conducted on ViT-B16.

Accuracy	Natural Accuracy	Isolating CLS	Patch Average	Patch Maximum
Top 1	84.20	78.61	75.75	80.16
Top 5	97.16	94.18	90.99	95.65

the CLS token captures global information mostly at the last layer, rather than building a global representation throughout the network.

We perform a second experiment to show this last-layer globalization behaviour is not exclusive to the CLS token, but actually occurs across every patch in the last layer. We take the fully connected layer trained to classify images on top of the CLS token, and *without any fine-tuning or adaptation*, we apply it to each patch, one at a time. This setup still successfully classifies 75.75% of the validation set, on average across individual patches, and the patch with the maximum performance achieves 80.16% accuracy (see Table 1), further confirming that the last layer performs a token mixing operation so that all tokens contain roughly identical information. Figure 5 contains a heat-map depicting the performance of this setup across spatial patches, indicating that patches in the center contain similar information to the CLS token, while peripheral patches only contain this information to a lesser extent. This observation stands in stark contrast to the suggestions of Raghu et al. (2021) that ViTs possess strong localization throughout the entire network, and their further hypothesis that the addition of global pooling is required for mixing tokens at the end of the network.

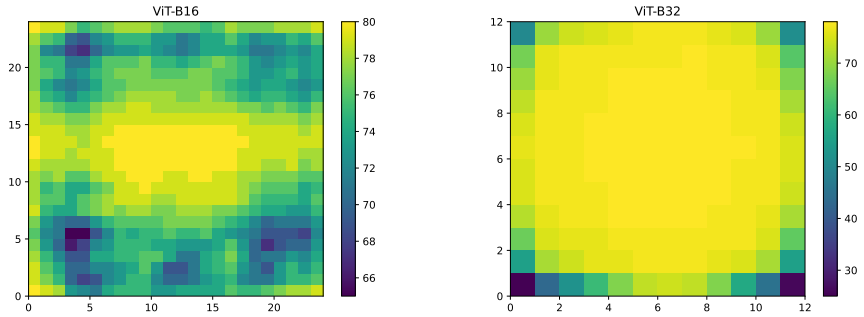


Figure 5: Heat map of classification accuracy on the validation set when we apply the classification head trained to classify images on the top of the CLS token to the other patches.

We conclude by noting that the information structure of a ViT is remarkably similar to a CNN, in the sense that the information is positionally encoded and preserved until the final layer. Furthermore, the final layer in ViTs appears to behave as a learned global pooling operation that aggregates information from all patches, which is similar to its explicit average-pooling counterpart in CNNs.

5 Comparison of ViTs and CNNs

As extensive work has been done to understand the workings of convolutional networks, including similar feature visualization and image reconstruction techniques to those used here, we may be able to learn more about ViT behavior via direct comparison to CNNs. An important observation is that in CNNs, early layers recognize color, edges, and texture, while deeper layers pick out increasingly complex structures eventually leading to entire objects (Olah et al., 2017). Visualization of features from different layers in a ViT, such as those in Figures 1 and 6, reveal that ViTs exhibit this kind of progressive specialization as well.

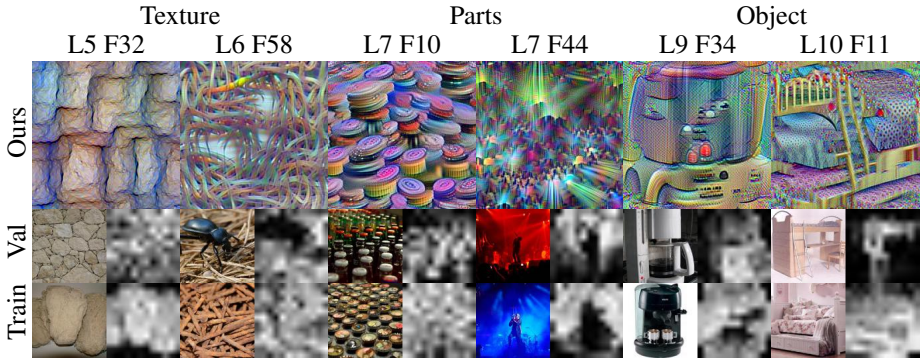


Figure 6: **Complexity of features vs depth in ViT B-32.** Visualizations suggest that ViTs are similar to CNNs in that they show a feature progression from textures to parts to objects as we progress from shallow to deep features.

On the other hand, we observe that there are also important differences between the ways CNNs and ViTs recognize images. In particular, we examine the reliance of ViTs and CNNs on background and foreground image features using the bounding boxes provided by ImageNet Deng et al. (2009). We filter the ImageNet-1k training images and only use those which are accompanied by bounding boxes. If several objects are present in an image, we only take the bounding boxes corresponding to the true class label and ignore the additional bounding boxes. Figure 7(b) shows an example of an image and variants in which the background and foreground, respectively, are masked.

(a) (b)

Figure 7: *(a)*: **ViT-B16 detects background features**. *Left:* Image optimized to maximally activate a feature from layer 6. *Center:* Corresponding maximally activating example from ImageNet. *Right:* The image's patch-wise activation map. *(b)*: **An example of an original image and masked-out foreground and background.**

Figure 7(a) displays an example of ViTs' ability to detect background information present in the ImageNet dataset. This particular feature appears responsible for recognizing the pairing of grass and snow. The right-most panel indicates that this feature is solely activated by the background, and not at all by the patches of the image containing parts of the wolf.

To quantitatively assess each architecture's dependence on different parts of the image on the dataset level, we mask out the foreground or background on a set of evaluation images using the aforementioned ImageNet bounding boxes, and we measure the resulting change in top-5 accuracy. These tests are performed across a number of pretrained ViT models, and we compared to a set of common CNNs in Table 2. Further results can be found in Table 3.

We observe that ViTs are significantly better than CNNs at using the background information in an image to identify the correct class. At the same time, ViTs also suffer noticeably less from the removal of the background, and thus seem to depend less on the background information to make their classification. A possible, and likely, confounding variable here is the imperfect separation of the background from the foreground in the ImageNet bounding box data set. A rectangle containing the wolf in Figure 7(a), for example, would also contain a small amount of the grass and snow at the wolf's feet. However, the foreground is typically contained entirely in a bounding box, so masking out the bounding box interiors is highly effective at removing the foreground. Because ViTs are

better equipped to make sense of background information, the leaked background may be useful for maintaining superior performance. Nonetheless, these results suggest that ViTs consistently outperform CNNs when information, either foreground or background, is missing.

Next, we study the role of texture in ViT predictions. To this end, we filter out high-frequency components from ImageNet test images via low-pass filtering. While the predictions of ResNets suffer greatly when high-frequency texture information is removed from their inputs, ViTs are seemingly resilient. See Figure 14 for the decay in accuracy of ViT and ResNet models as textural information is removed.

Table 2: **ViTs more effectively correlate background information with correct class.** Both foreground and background data are normalized by full image top-5 accuracy.

Normalized Top-5 ImageNet Accuracy			
Architecture	Full Image	Foreground	Background
ViT-B32	98.44	93.91	28.10
ViT-L16	99.57	**96.18**	**33.69**
ViT-L32	99.32	93.89	31.07
ViT-B16	99.22	95.64	31.59
ResNet-50	98.00	89.69	18.69
ResNet-152	98.85	90.74	19.68
MobileNetv2	96.09	86.84	15.94
DenseNet121	96.55	89.58	17.53

6 ViTs with Language Model Supervision

Recently, ViTs have been used as a backbone to develop image classifiers trained with natural language supervision and contrastive learning techniques (Radford et al., 2021). These *CLIP* models are state-of-the-art in transfer learning to unseen datasets. The zero-shot ImageNet accuracy of these models is even competitive with traditionally trained ResNet-50 competitors. We compare the feature visualizations for ViT models with and without CLIP training to study the effect of natural language supervision on the behavior of the transformer-based backbone.

The training objective for CLIP models consists of matching the correct caption from a list of options with an input image (in feature space). Intuitively, this procedure would require the network to extract features not only suitable for detecting nouns (e.g. simple class labels like 'bird'), but also modifying phrases like prepositions and epithets. Indeed, we observe several such features that are not present in ViTs trained solely as image classifiers.

Figure 8(a) shows the image optimized to maximally activate a feature in the fifth layer of a ViT CLIP model alongside its two highest activating examples from the ImageNet dataset. The fact that all three images share sharp boundaries indicates this feature might be responsible for detecting caption texts relating to a progression of images. Examples could include "before and after," as in the airport images or the adjective "step-by-step" for the iPod teardown. Similarly, Figure 8(b) and 8(c) depict visualizations from features which seem to detect the preposition "from above", and adjectives relating to a multitude of the same object, respectively.

The presence of features that represent conceptual categories is another consequence of CLIP training. Unlike ViTs trained as classifiers, in which features detect single objects or common background information, CLIP-trained ViTs produce features in deeper layers activated by objects in clearly discernible conceptual categories. For example, the top left panel of Figure 9(a) shows a feature activated by what resembles skulls alongside tombstones. The corresponding seven highly activating images from the dataset include other distinct objects such as bloody weapons, zombies, and skeletons. From a strictly visual point of view, these classes have very dissimilar attributes, indicating this feature might be responsible for detecting components of an image relating broadly to morbidity. In Figure 9(b), we see that the top left-most panel shows a disco ball, and the corresponding images from the dataset contain boomboxes, speakers, a record player, audio recording equipment, and a performer. Again, these are visually distinct classes, yet they are all united by the concept of music.

(a) Before and after/Step-by-step (b) From above (c) Many

Figure 8: *Left:* Feature optimization shows sharp boundaries, and maximally activating ImageNet examples contain distinct, adjacent images. *Middle:* Feature optimization and maximally activating ImageNet photos all show images from an elevated vantage point. *Right:* Feature optimization shows a crowd of people, but maximally activating images indicate that the repetition of objects is more relevant than the type of object.

Given that the space of possible captions for images is substantially larger than the mere one thousand classes in the ImageNet dataset, high performing CLIP models understandably require higher level organization for the objects they recognize. Moreover, the CLIP dataset is scraped from the internet, where captions are often more descriptive than simple class labels.

7 Discussion

In order to dissect the inner workings of vision transformers, we introduce a framework for optimization-based feature visualization. We then identify which components of a ViT are most amenable to producing interpretable images, finding that the high-dimensional inner projection of the feed-forward layer is suitable while the key, query, and value features of self-attention are not.

Applying this framework to said features, we observe that ViTs preserve spatial information of the patches even for individual channels across all layers with the exception of the last layer, indicating that the networks learn spatial relationships from scratch. We further show that the sudden disappearance of localization information in the last attention layer results from a learned token mixing behavior that resembles average pooling.

In comparing CNNs and ViTs, we find that ViTs make better use of background information and are able to make vastly superior predictions relative to CNNs when exposed only to image backgrounds despite the seemingly counter-intuitive property that ViTs are not as sensitive as CNNs to the loss of high-frequency information, which one might expect to be critical for making effective use of background. We also conclude that the two architectures share a common property whereby earlier layers learn textural attributes, whereas deeper layers learn high level object features or abstract concepts. Finally, we show that ViTs trained with language model supervision learn more semantic and conceptual features, rather than object-specific visual features as is typical of classifiers.

We make our code repository available at: `https://github.com/anonymous2023iclr/ViTVis`

(a) Category of morbidity (b) Category of music

Figure 9: **Features from ViT trained with CLIP that relates to the category of morbidity and music.** *Top-left image in each category:* Image optimized to maximally activate a feature from layer 10. *Rest:* Seven of the ten ImageNet images that most activate the feature.

REFERENCES

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 782–791, 2021.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *arXiv preprint arXiv:2104.13840*, 1(2):3, 2021.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *arXiv preprint arXiv:2106.04803*, 2021.

Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Peco: Perceptual codebook for bert pre-training of vision transformers. *arXiv preprint arXiv:2111.12710*, 2021.

Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4829–4837, 2016.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Amin Ghiasi, Hamid Kazemi, Steven Reich, Chen Zhu, Micah Goldblum, and Tom Goldstein. Plug-in inversion: Model-agnostic inversion for vision with data augmentations. 2021.

Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 6(3):e30, 2021.

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

Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. *arXiv preprint arXiv:2103.16302*, 2021.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. *arXiv preprint arXiv:2111.09883*, 2021.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5188–5196, 2015.

Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *arXiv preprint arXiv:2105.10497*, 2021.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. *arXiv preprint arXiv:2105.07581*, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34, 2021.

Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of visual transformers. *arXiv preprint arXiv:2103.15670*, 2021.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*, 2014.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, volume 139, pp. 10347–10357, July 2021a.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021b.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. *arXiv preprint arXiv:2104.06399*, 2021.

Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8715–8724, 2020.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. *arXiv preprint arXiv:2106.04560*, 2021.

Roland Zimmermann, Judy Borowski, Robert Geirhos, Matthias Bethge, Thomas Wallis, and Wieland Brendel. How well do feature visualizations support causal understanding of cnn activations? *Advances in Neural Information Processing Systems*, 34, 2021.

A FAILED EXAMPLES

Figure 10 shows few examples of our visualization method failing when applied on low dimensional spaces. We attribute this to entanglement of more than 768 features when represented by vectors of size 768. We note that, due to skip connections, activation in previous layers can cause activation in the next layer for the same feature, consequently, the visualizations of the same features in different layers can share visual similarities.

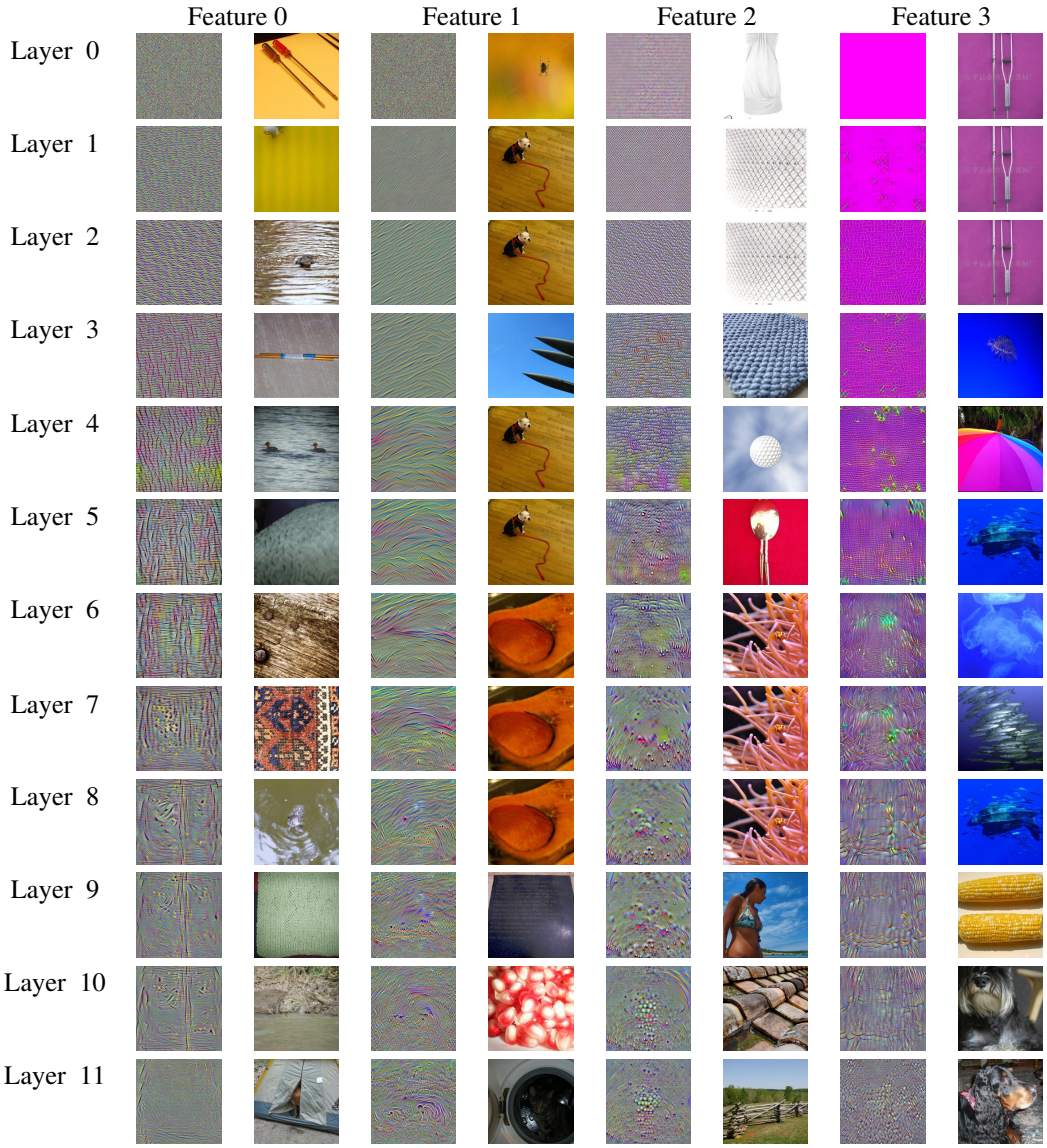


Figure 10: **Some examples of failed visualizations on the input of the attention layers**. Same visualization technique fails when applied on low dimensional (e.g. on key, query, value, etc) spaces. We believe that the visualization shows roughly meaningful and interpretable visualizations in early layers, since there are not many different features to be embedded. However, in deeper layers the features are entangled, so it is more difficult to visualize them. For every example, the picture on the left shows the results of optimization and the picture on the right shows the most activating image from ImageNet1k validation set.

B EXPERIMENTAL SETUP AND HYPERPARAMETERS

As mentioned before, we ensemble augmentations to the input. More specifically, we use $\mathcal{A}$ is $GS(CS(Jitter(x)))$ as our augmentation. The bound for $Jitter$ is $(-32, 32)$ for both directions vertical and horizontal. The hyper parameters for CS are always $mean = 1.0$ and $std = 1.0$ in all of the experiments. For GS, the $mean$ is always 0, however, for the std we have a linear scheduling, where at the beginning of the optimization the $std = 0.5$ and at the end of the optimization $std = 0.0$. We use a batch-size $n = 8$ for all of our experiments. We use ADAM as our choice of optimizer with $\beta = (0.5, 0.99)$. Optimization is done in 400 steps and at every step, we re-sample the augmentations $Jitter$, GS and CS. We also use a CosineAnealing learning rate scheduling, starting from $lr = 0.1$ at the beginning and $l = 0$ the end. The hyper-parameter used for total variation $\lambda_{tv} = 0.00005$.

For all ViT experiments, we use the pretrained models from https://github.com/lukemelas/PyTorch-Pretrained-ViT. For clip models, we use pretrained models by Wightman (2019). The rest of models we use are from https://github.com/pytorch/vision.

For all of our experiments, we use GeForce RTX 2080 Ti GPUs with 12GB of memory. All inferences on ImageNet are done under 20 minutes on validation set and under 1 hour on training set using only 1 GPU. All visualization experiments take at most 90 seconds to complete.

C MODELS

In our experiments, we use publicly available pre-trained models from various sources. The following tables list the models used from each source, along with references to where they are introduced in the literature.

Name	Paper
B_16_imagenet1k	Dosovitskiy et al. (2021)
B_32_imagenet1k	Dosovitskiy et al. (2021)

Figure 11: Pre-trained models used from : https://github.com/lukemelas/PyTorch-Pretrained-ViT.

Name	Paper
deit_base_patch16_224	Touvron et al. (2021b)
deit_base_distilled_patch16_384	Touvron et al. (2021b)
deit_base_patch16_384	Touvron et al. (2021b)
deit_tiny_distilled_patch16_224	Touvron et al. (2021b)
deit_small_distilled_patch16_224	Touvron et al. (2021b)
deit_base_distilled_patch16_224	Touvron et al. (2021b)

Figure 12: Pre-trained models from Touvron et al. (2021a) .

Name	Paper
coat_lite_mini	Xu et al. (2021)
coat_lite_small	Xu et al. (2021)
coat_lite_tiny	Xu et al. (2021)
coat_mini	Xu et al. (2021)
coat_tiny	Xu et al. (2021)
convit_base	d'Ascoli et al. (2021)
convit_small	d'Ascoli et al. (2021)
convit_tiny	d'Ascoli et al. (2021)
pit_b_224	Heo et al. (2021)
pit_b_distilled_224	Heo et al. (2021)
pit_s_224	Heo et al. (2021)
pit_s_distilled_224	Heo et al. (2021)
pit_ti_224	Heo et al. (2021)
pit_ti_distilled_224	Heo et al. (2021)
swin_base_patch4_window7_224	Liu et al. (2021)
swin_base_patch4_window7_224_in22k	Liu et al. (2021)
swin_base_patch4_window12_384	Liu et al. (2021)
swin_base_patch4_window12_384_in22k	Liu et al. (2021)
swin_large_patch4_window7_224	Liu et al. (2021)
swin_large_patch4_window7_224_in22k	Liu et al. (2021)
swin_large_patch4_window12_384	Liu et al. (2021)
swin_large_patch4_window12_384_in22k	Liu et al. (2021)
swin_small_patch4_window7_224	Liu et al. (2021)
swin_tiny_patch4_window7_224	Liu et al. (2021)
twins_pcpvt_base	Chu et al. (2021)
twins_pcpvt_large	Chu et al. (2021)
twins_pcpvt_small	Chu et al. (2021)
twins_svt_base	Chu et al. (2021)
twins_svt_large	Chu et al. (2021)
twins_svt_small	Chu et al. (2021)

Figure 13: Pre-trained models used from: Wightman (2019)

D EFFECT OF LOW-PASS FILTERING

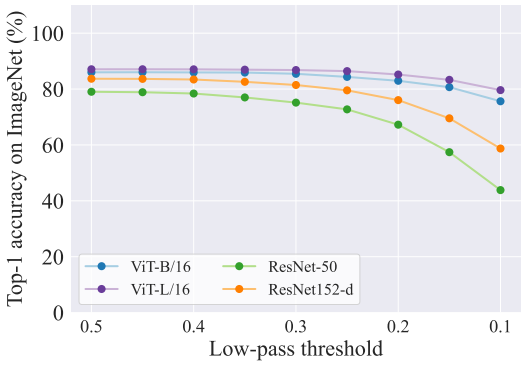


Figure 14: **Effect of low-pass filtering on top-1 ImageNet accuracy.** CNNs are *more* dependent on high frequency textural image information than ViTs.

E ADDITIONAL VISUALIZATIONS

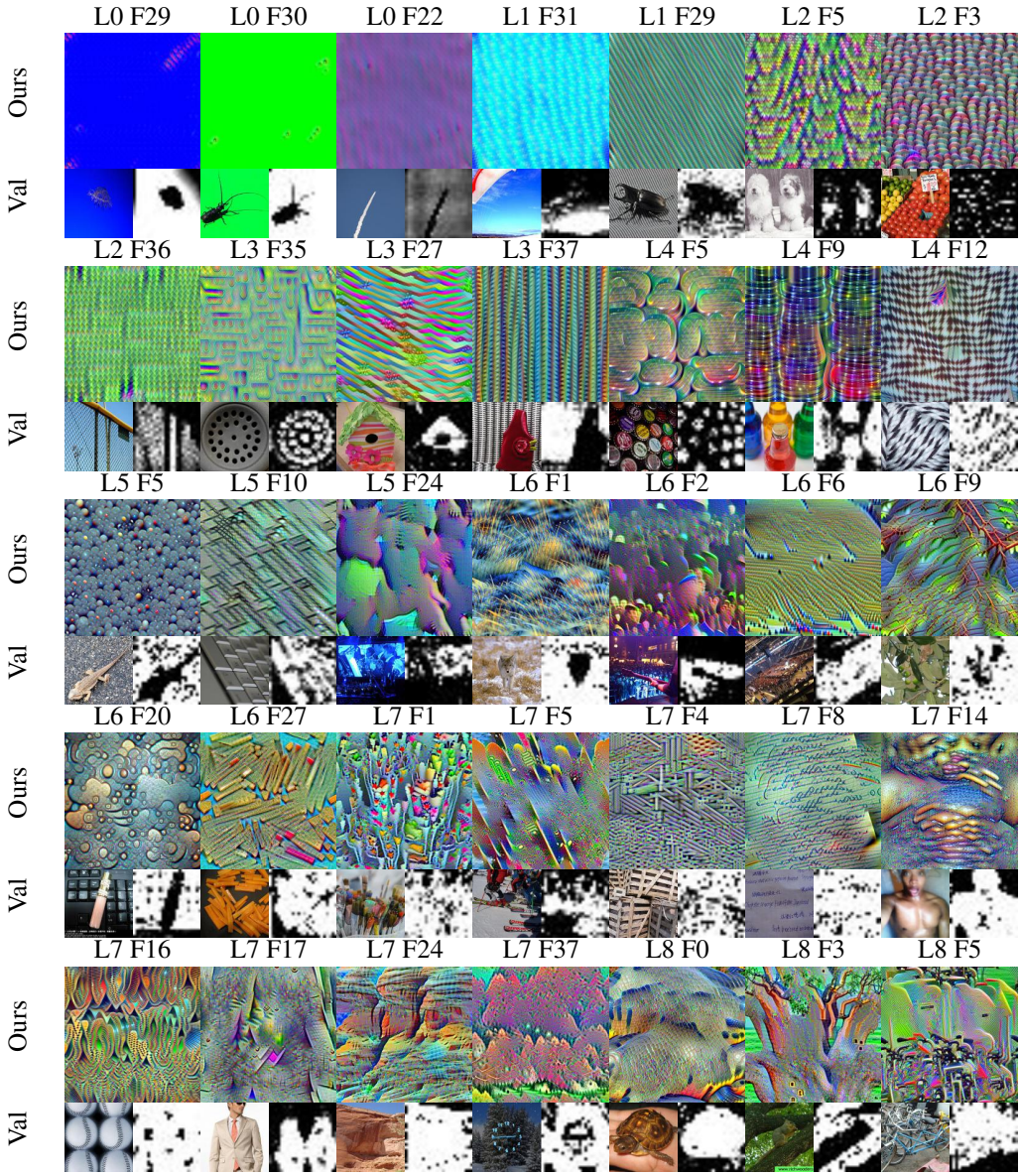


Figure 15: Visualization of ViT-base-patch16

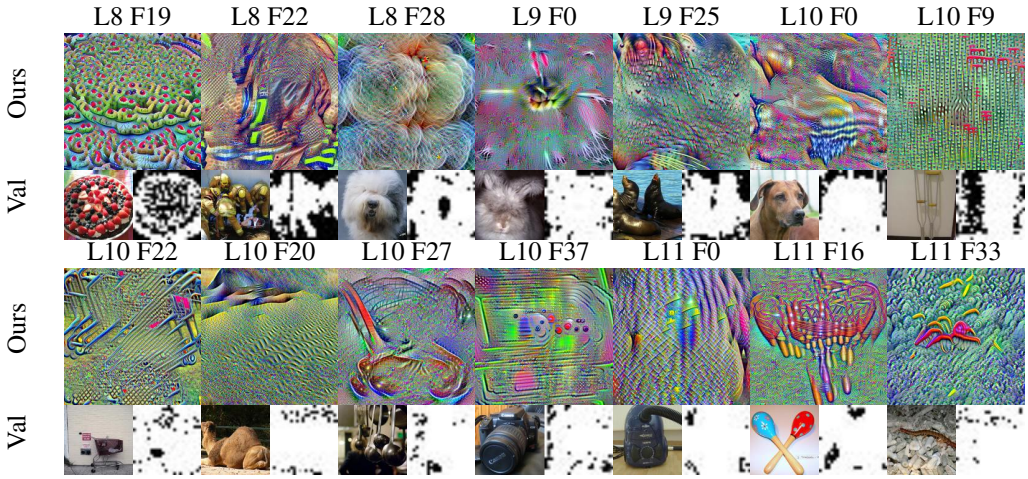


Figure 16: (Cont.) Visualization of ViT-base-patch16

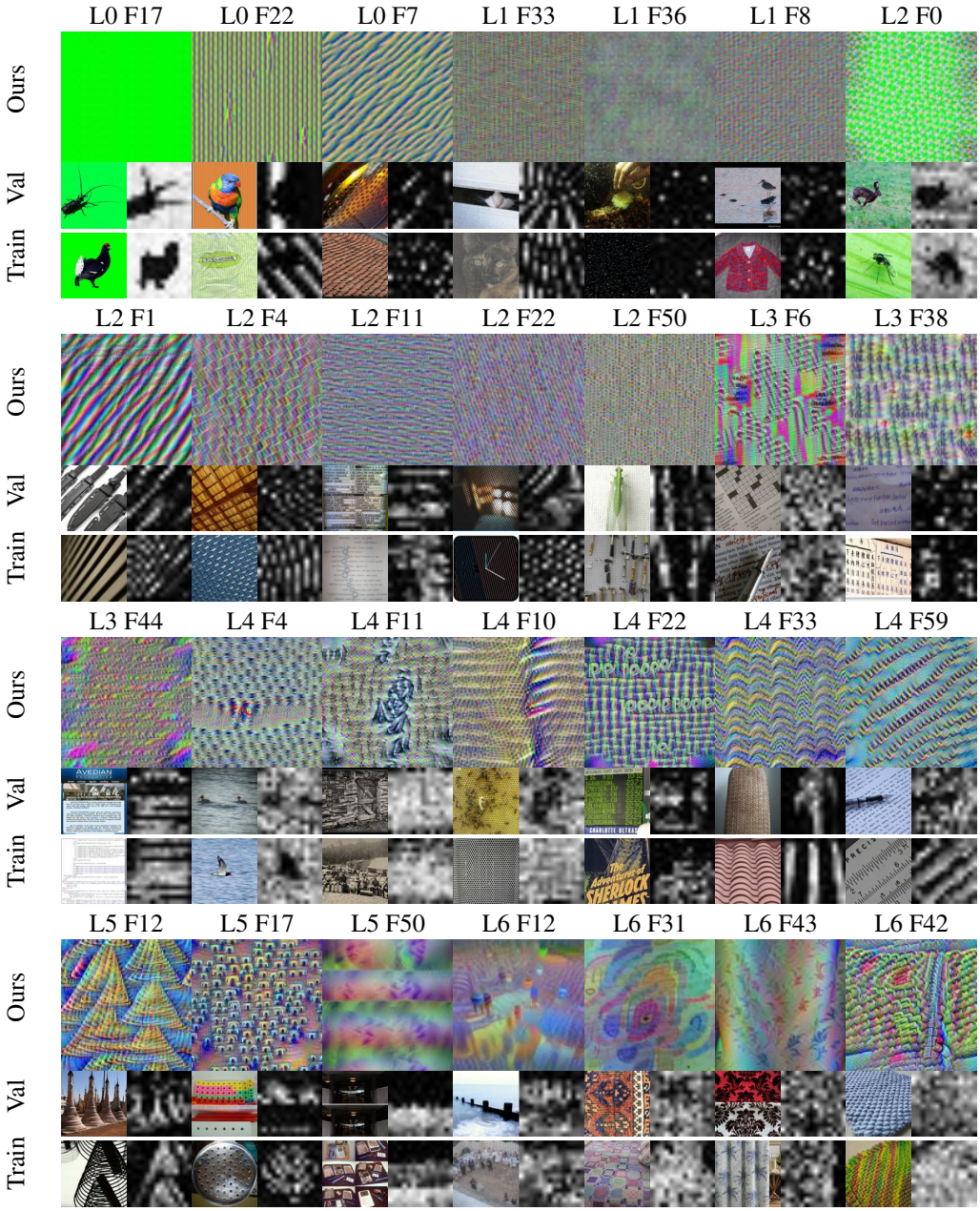


Figure 17: Visualization of a CLIP model with ViT-base-patch16 as its visual part.

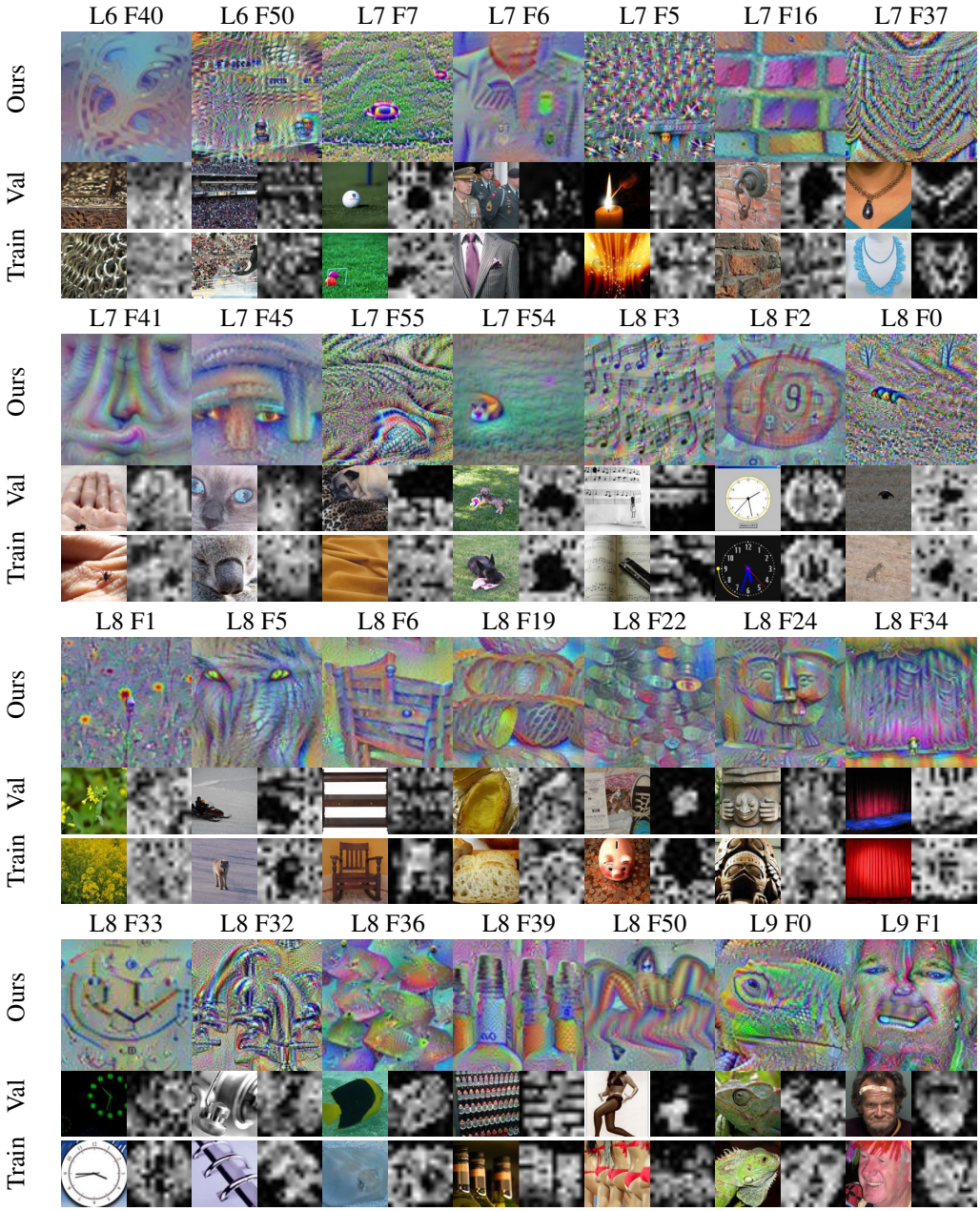


Figure 18: (Cont.) Visualization of a CLIP model with ViT-base-patch16 as its visual part.

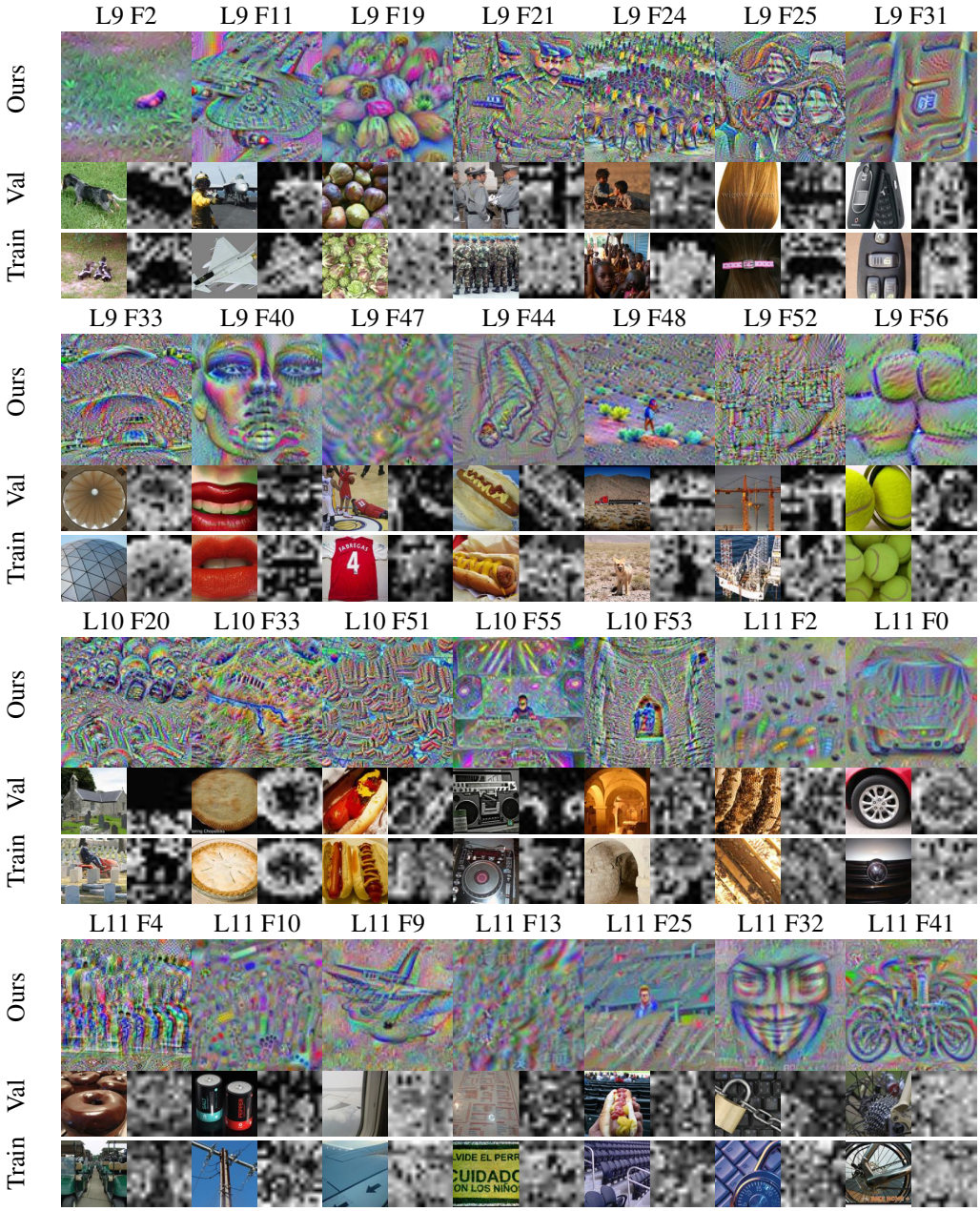


Figure 19: (Cont.) Visualization of a CLIP model with ViT-base-patch16 as its visual part.

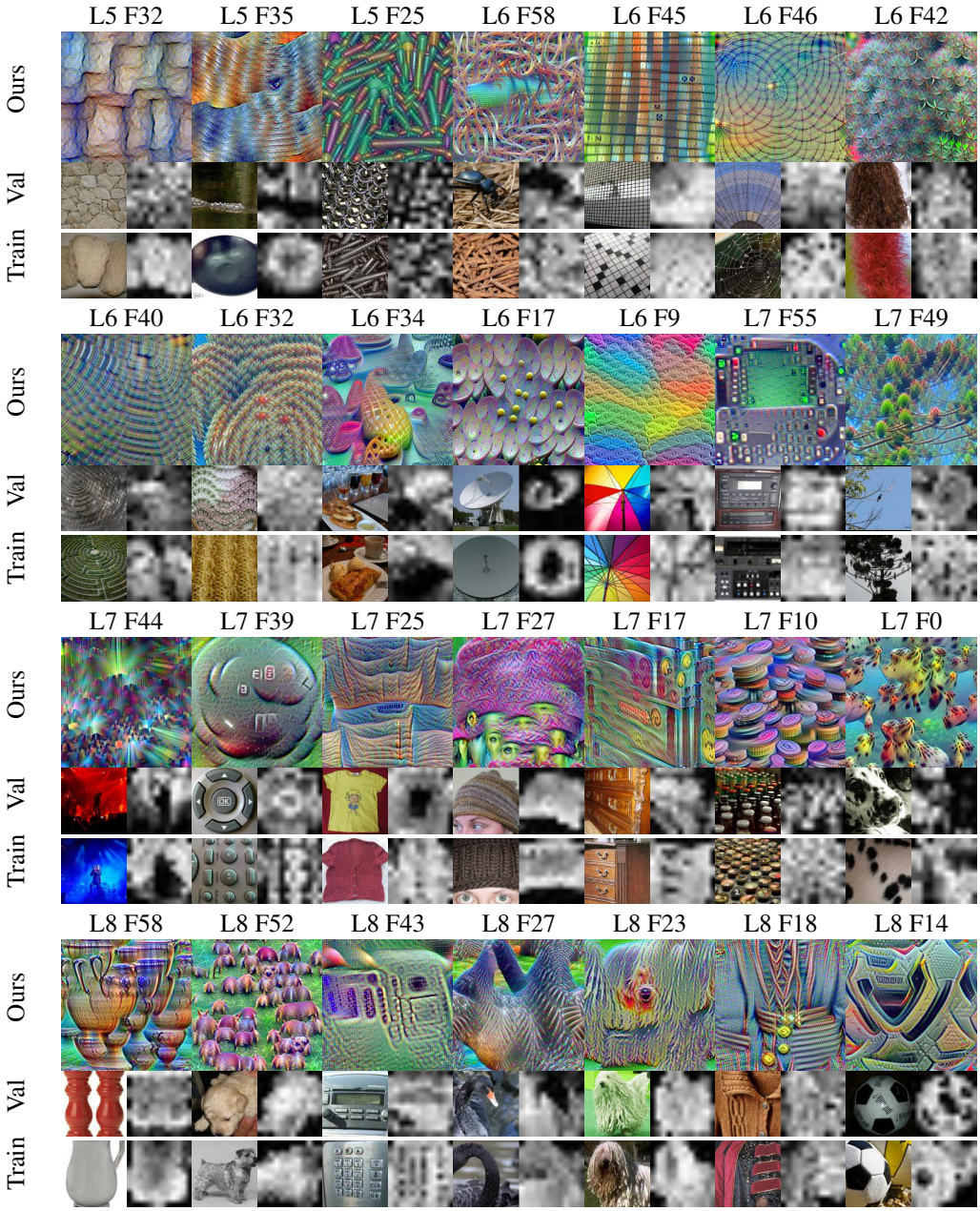


Figure 20: Visualization of ViT-base-patch32

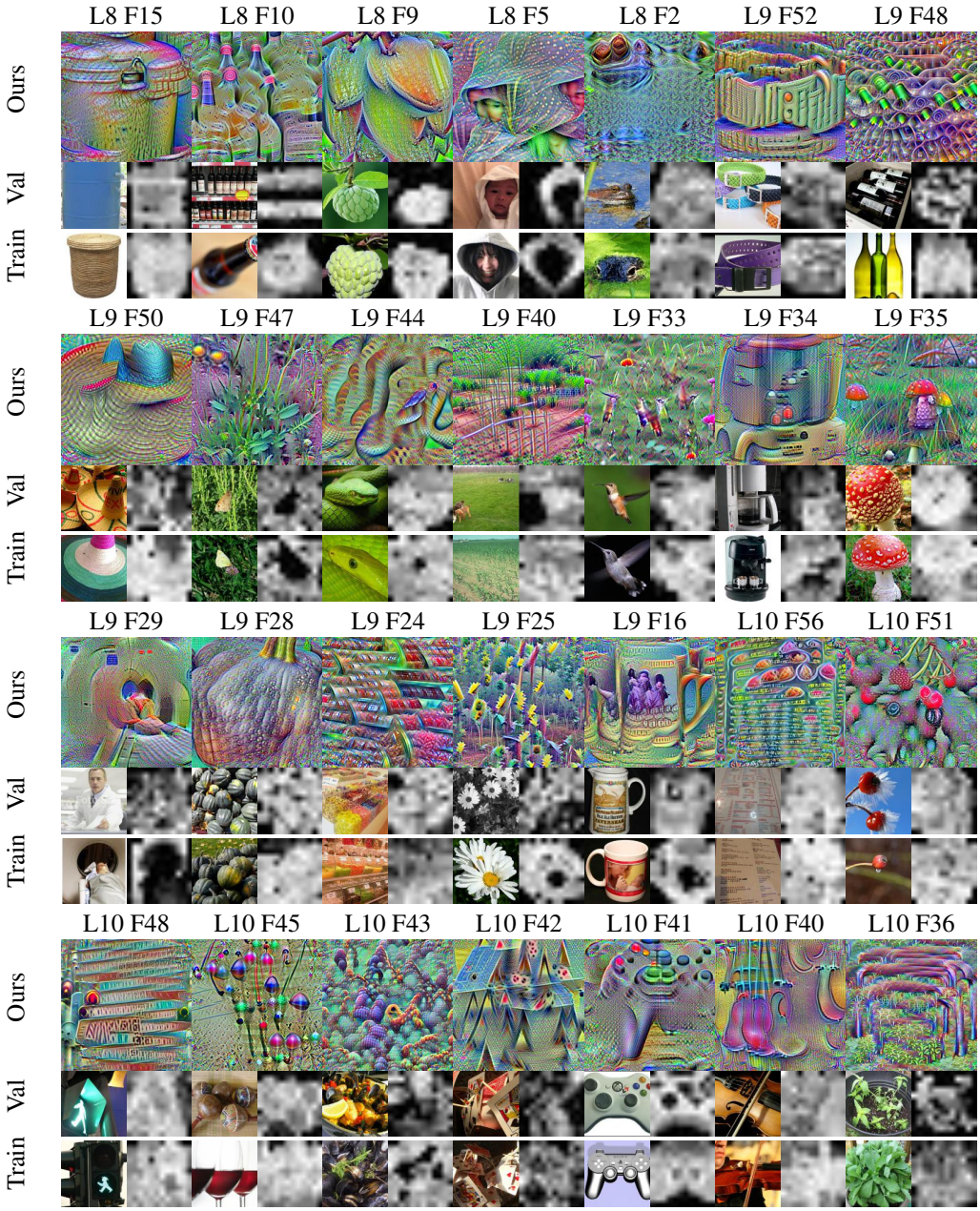


Figure 21: (Cont.) Visualization of ViT-base-patch32

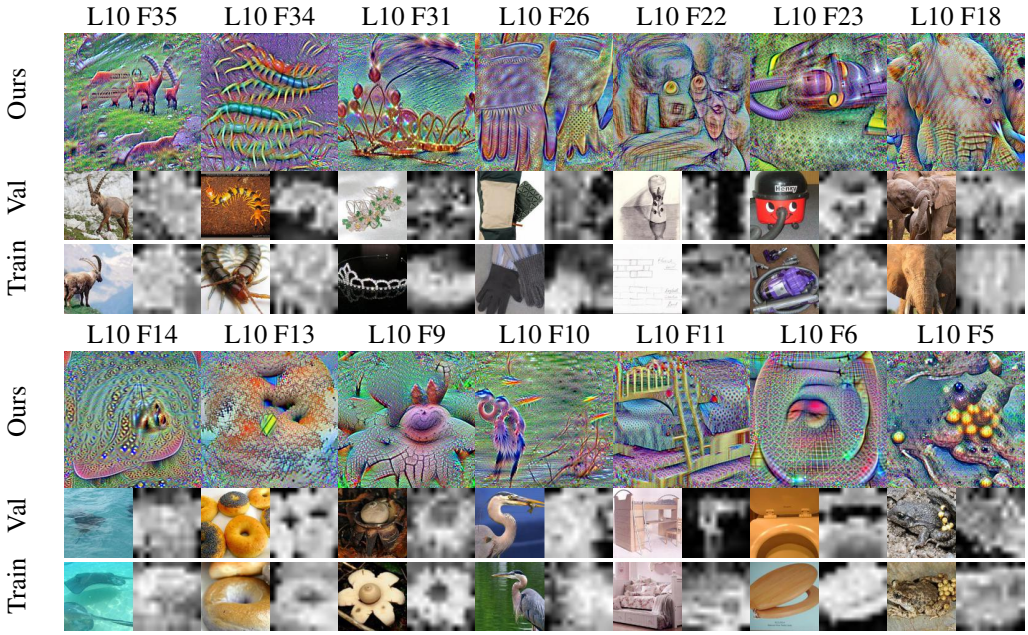


Figure 22: (Cont.) Visualization of ViT-base-patch32

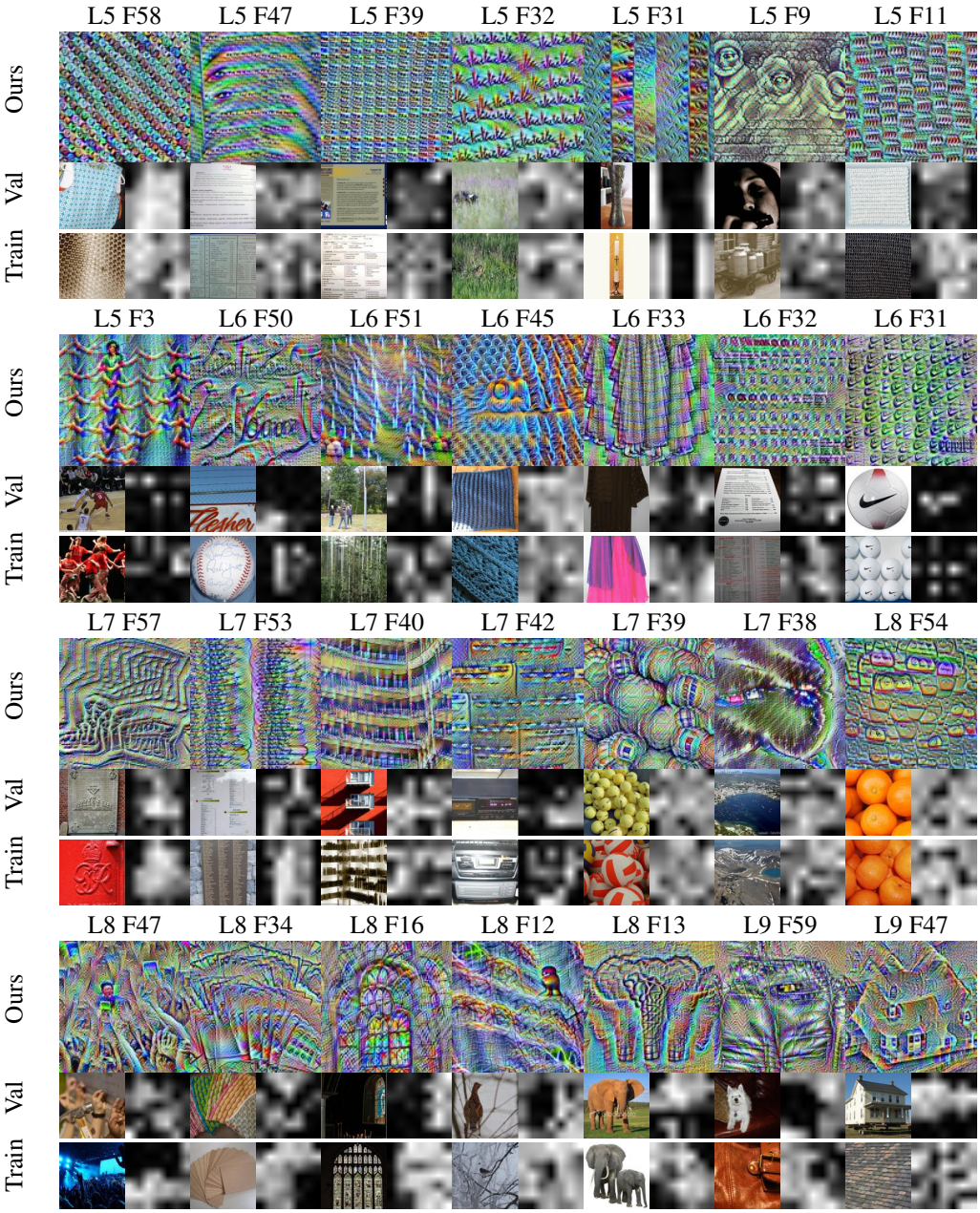


Figure 23: Visualization of a CLIP model with ViT-base-patch32 as its visual part.

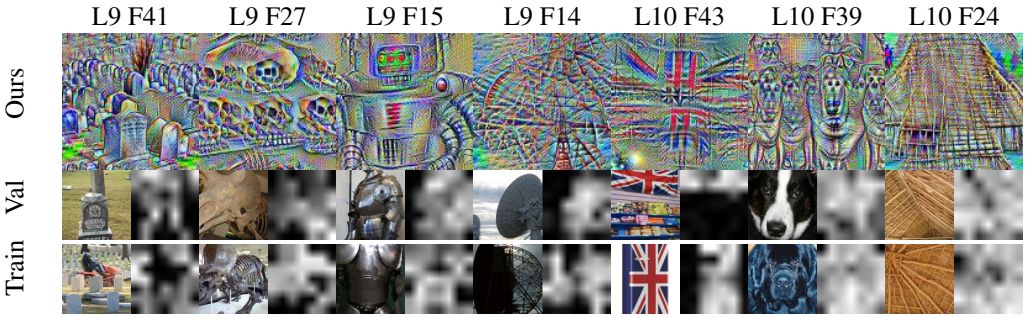


Figure 24: (Cont.) Visualization of a CLIP model with ViT-base-patch32 as its visual part.

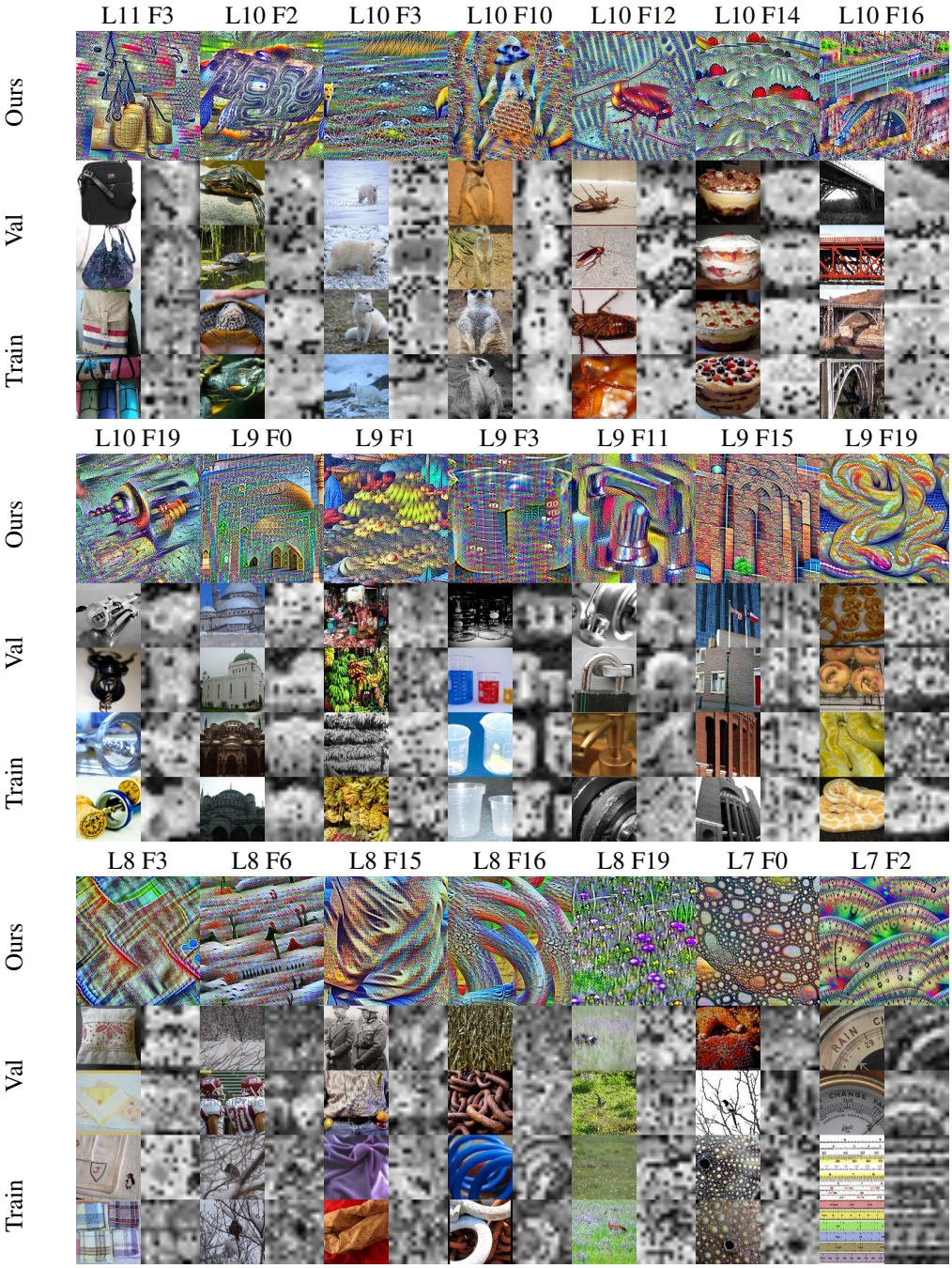


Figure 25: Visualization of features in Deit base p-16 im-224

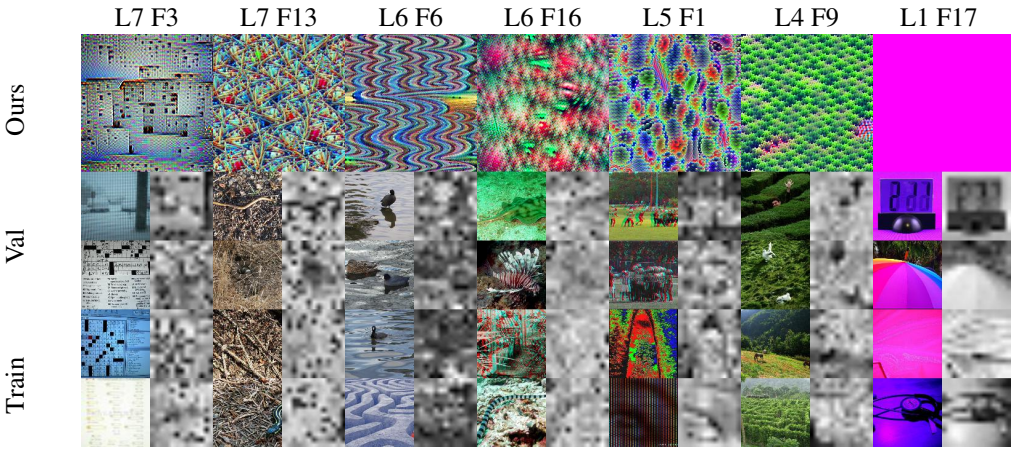


Figure 26: (Cont.) Visualization of features in Deit base p-16 im-224

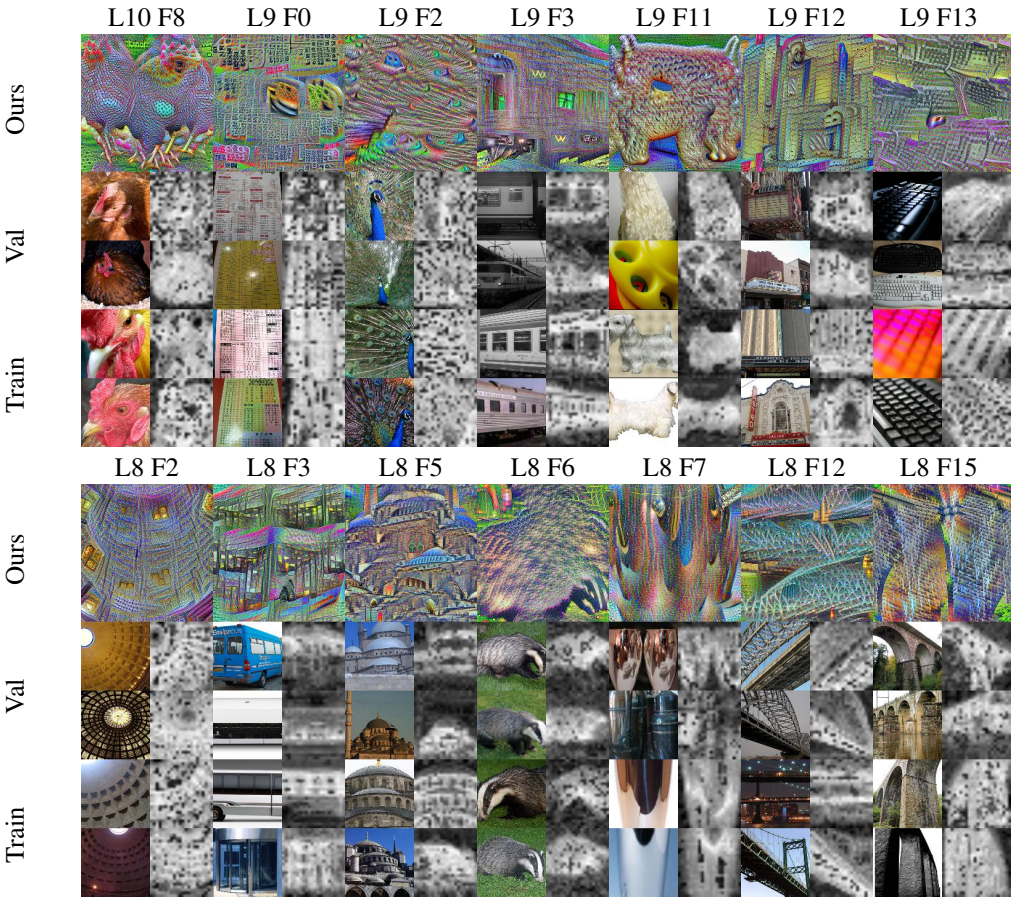


Figure 27: Visualization of features in DeiT base p-16 im-384

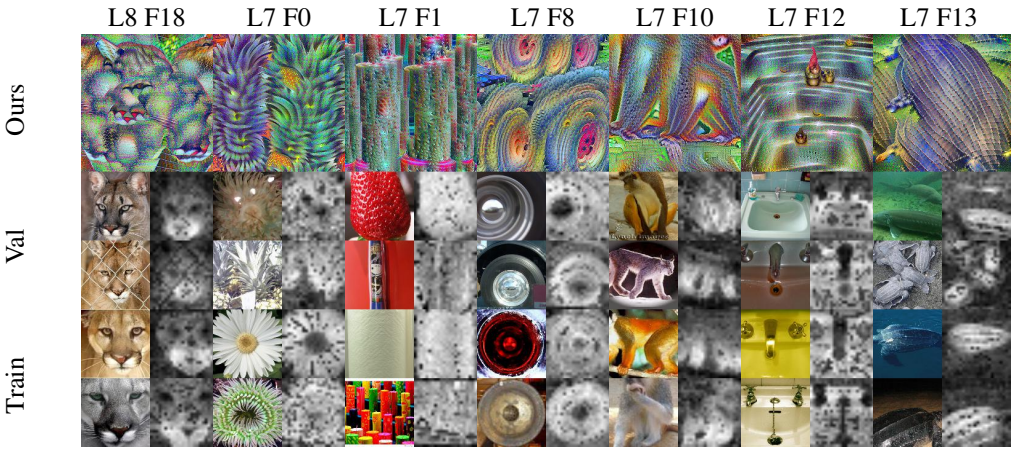


Figure 28: (Cont.) Visualization of features in DeiT base p-16 im-384

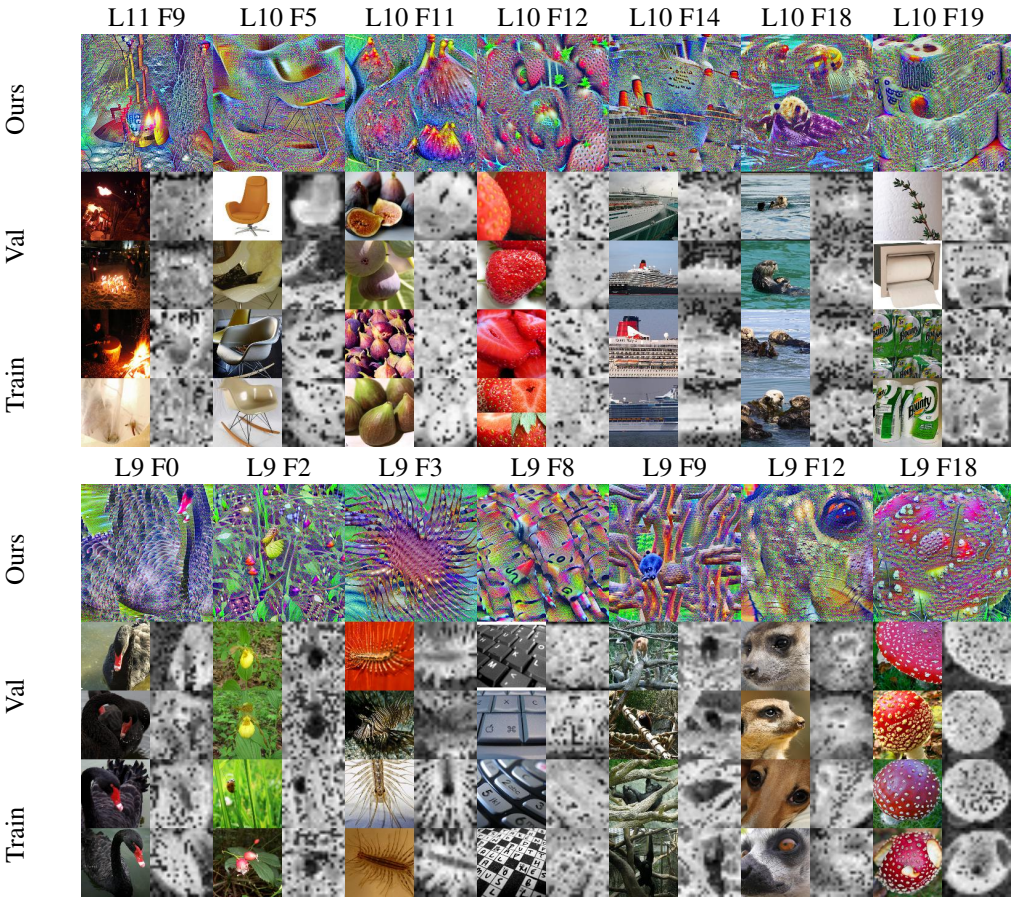


Figure 29: Visualization of features in DeiT base p-16 im-384

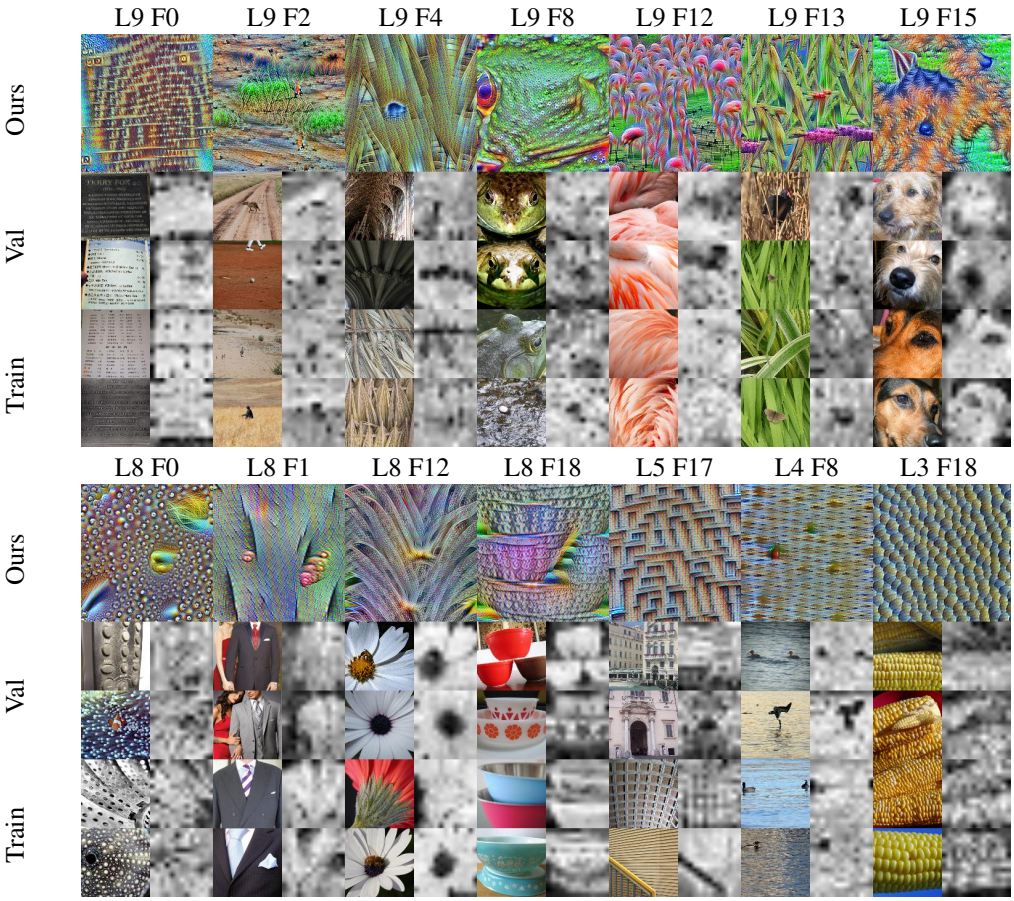


Figure 30: Visualization of features in DeiT tiny distilled p-16 im-224

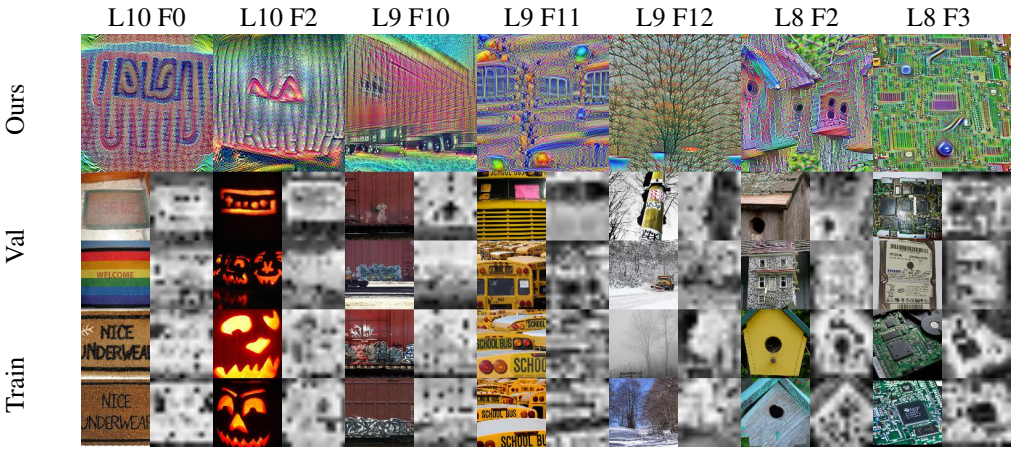


Figure 31: Visualization of features in DeiT small distilled p-16 im-224

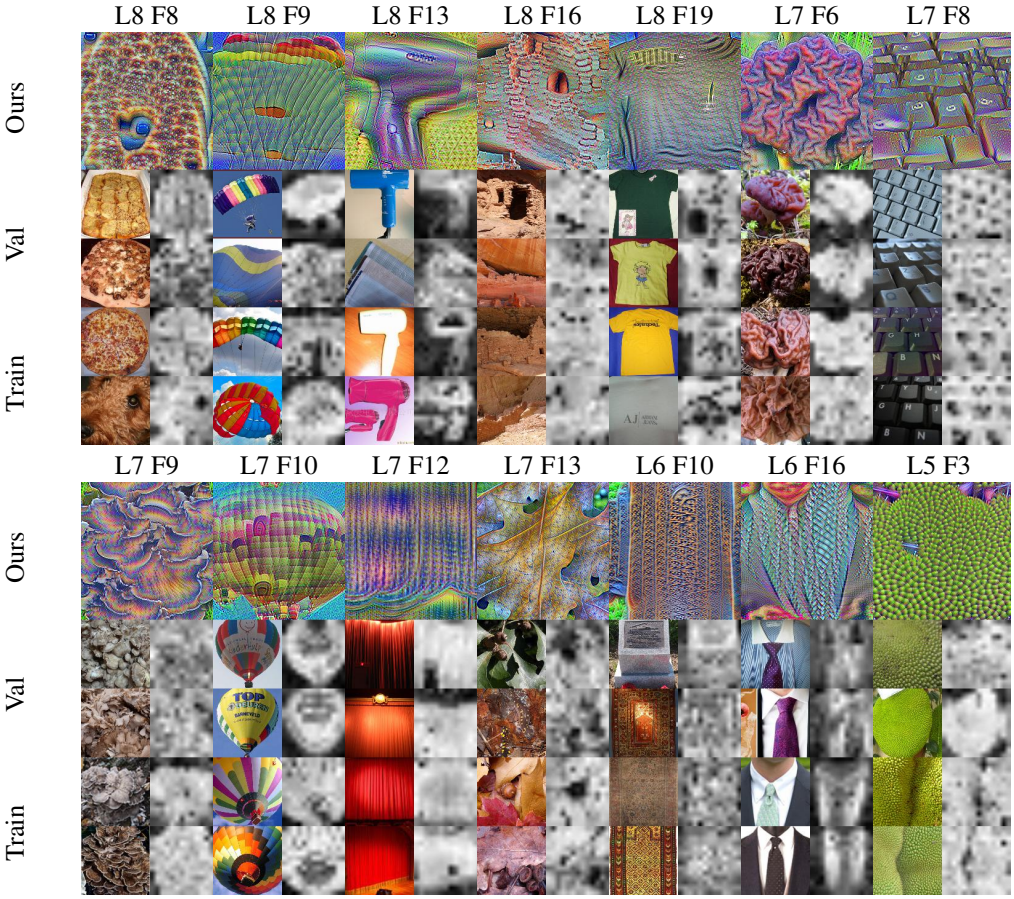


Figure 32: Visualization of features in DeiT small distilled p-16 im-224

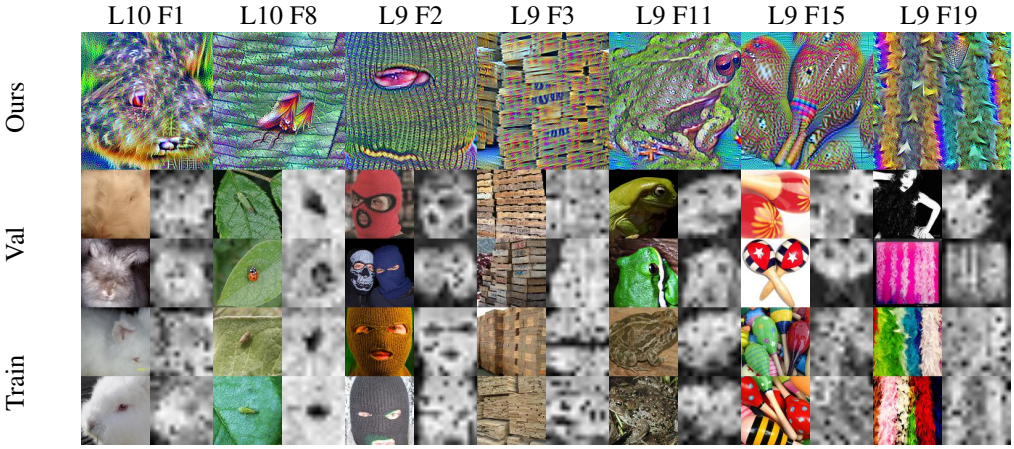


Figure 33: Visualization of features in DeiT base distilled p-16 im-224

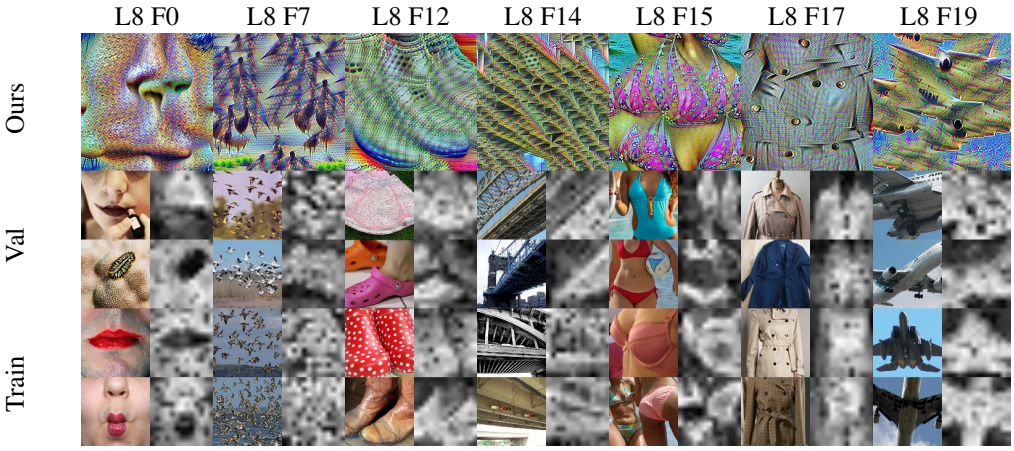


Figure 34: (Cont.) Visualization of features in DeiT base distilled p-16 im-224

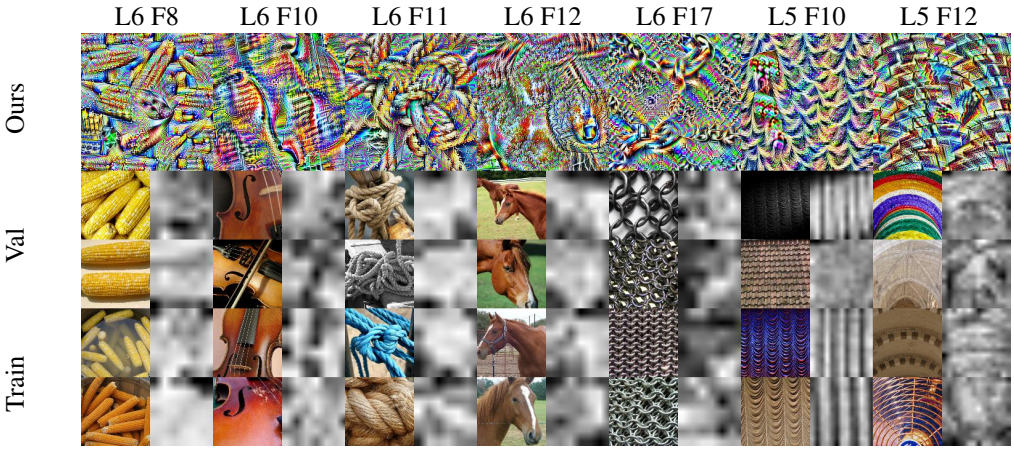


Figure 35: Visualization of features in Coat lite mini

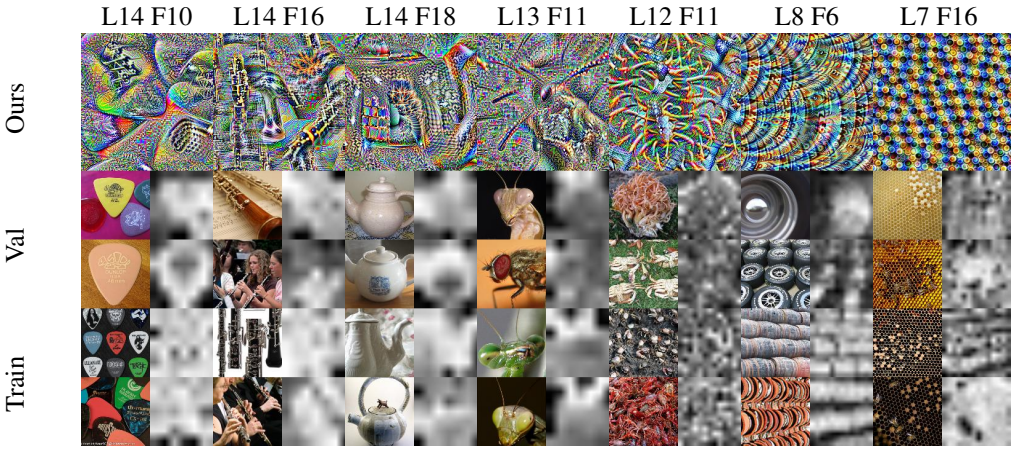


Figure 36: Visualization of features in Coat lite small

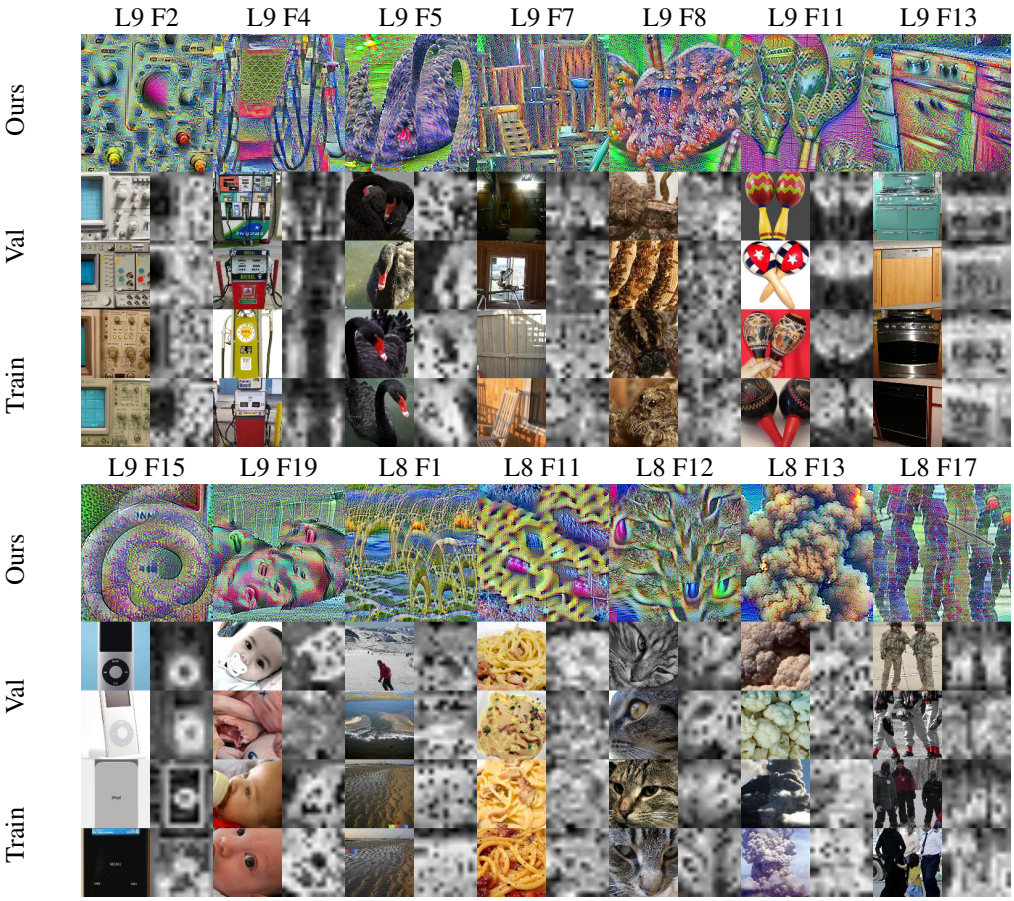


Figure 37: Visualization of features in ConViT base

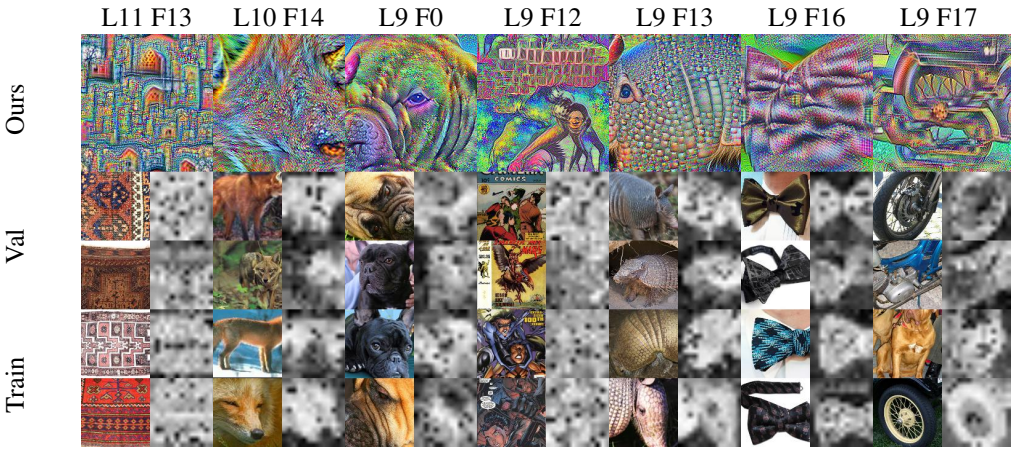


Figure 38: Visualization of features in ConViT small.

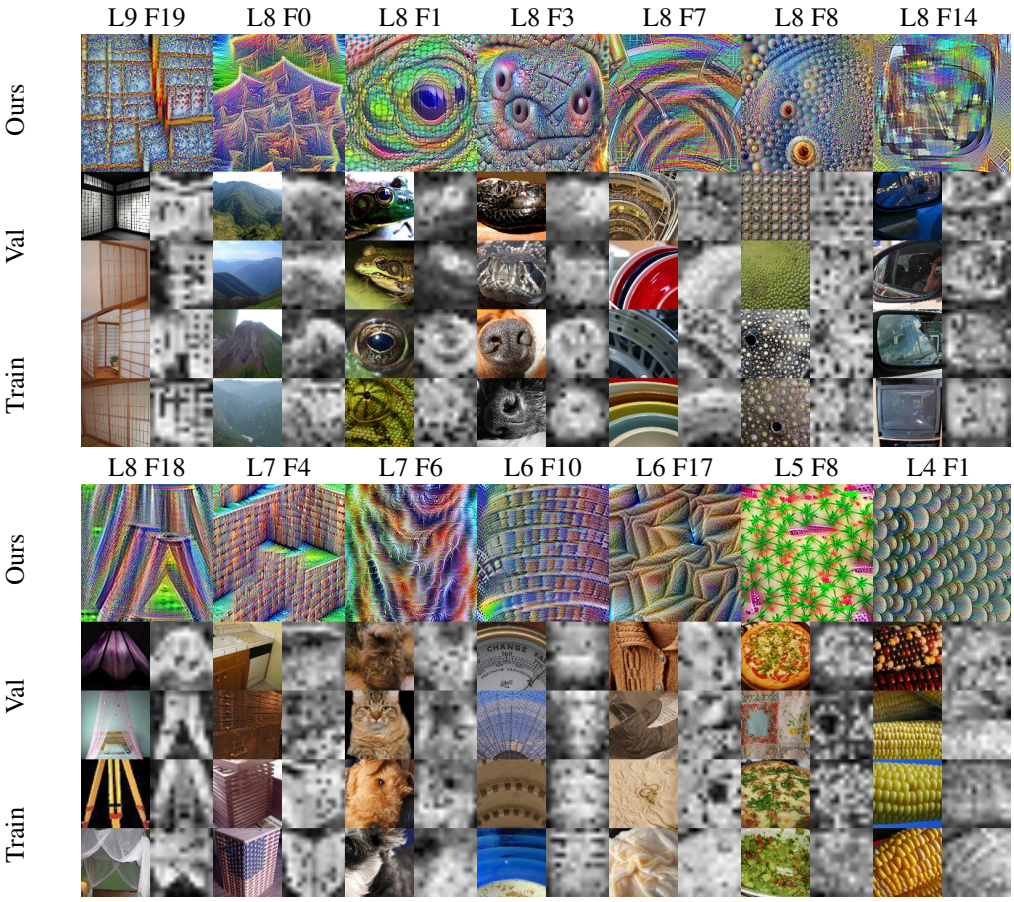


Figure 39: (Cont.) Visualization of features in ConViT small.

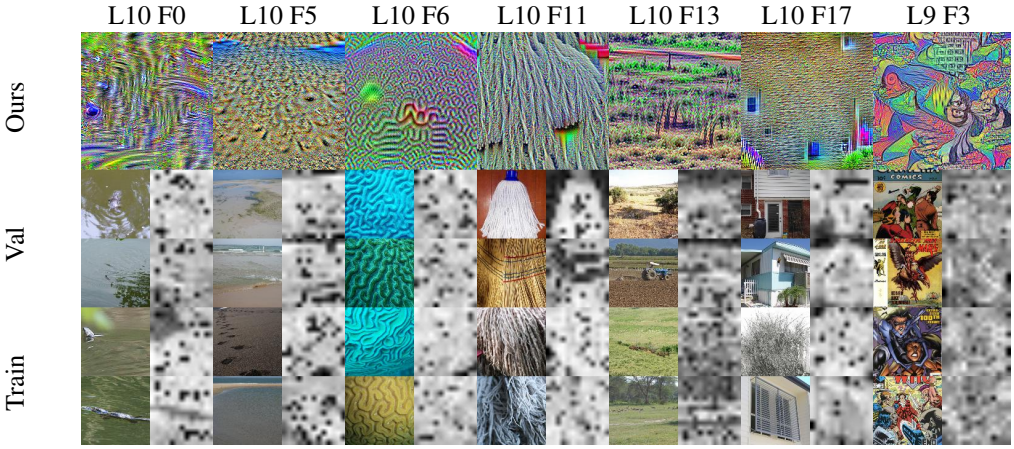


Figure 40: Visualization of features in ConViT tiny.

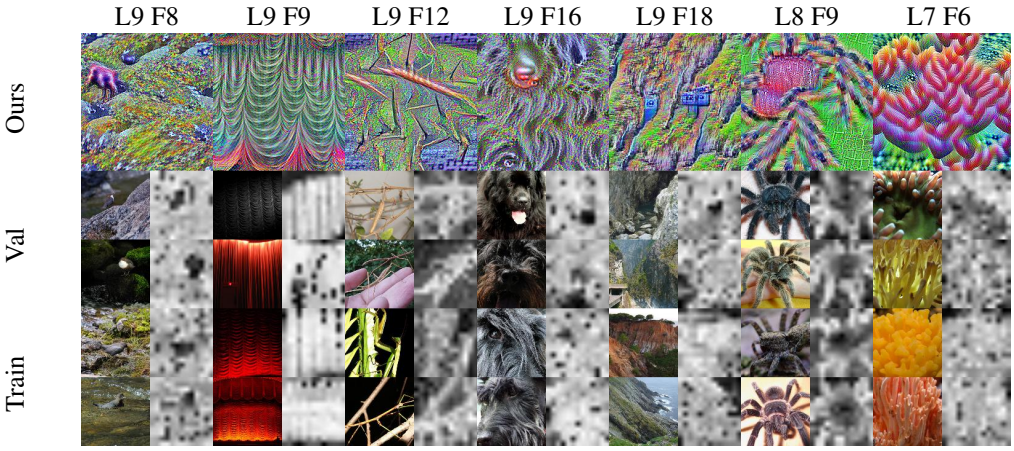


Figure 41: (Cont.) Visualization of features in ConViT tiny.

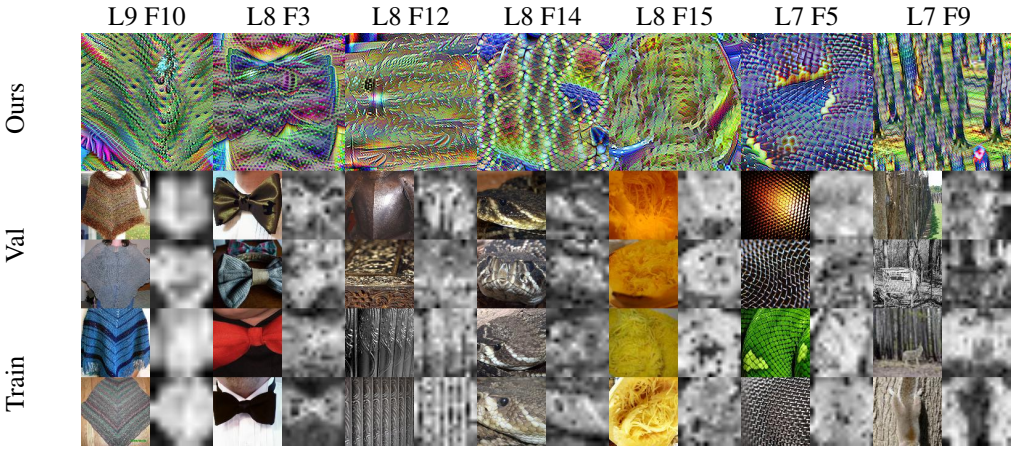


Figure 42: Visualization of features in Pit base im-224

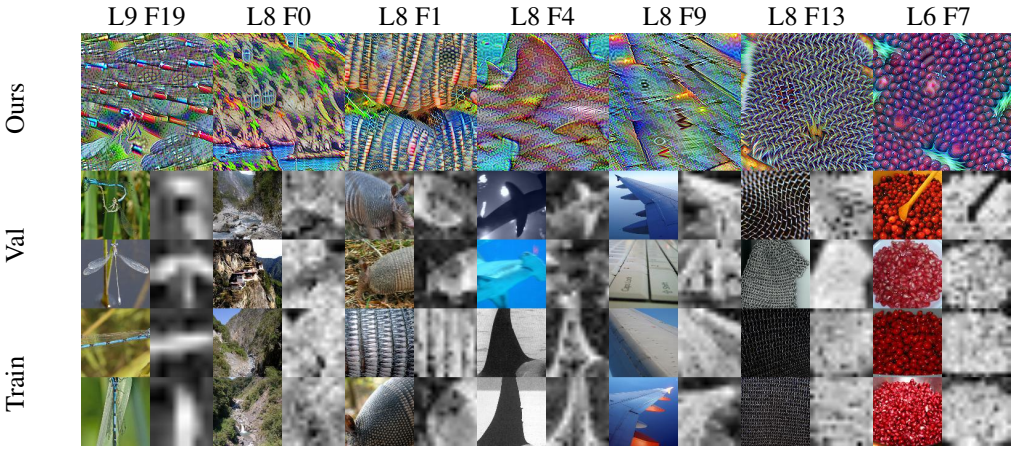


Figure 43: Visualization of features in Pit base distilled im-224.

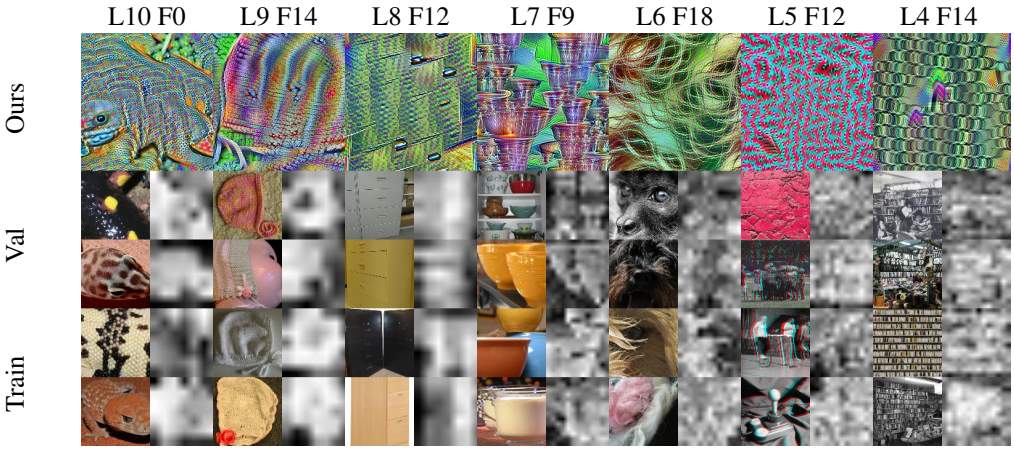


Figure 44: Visualization of features in Pit small im-224.

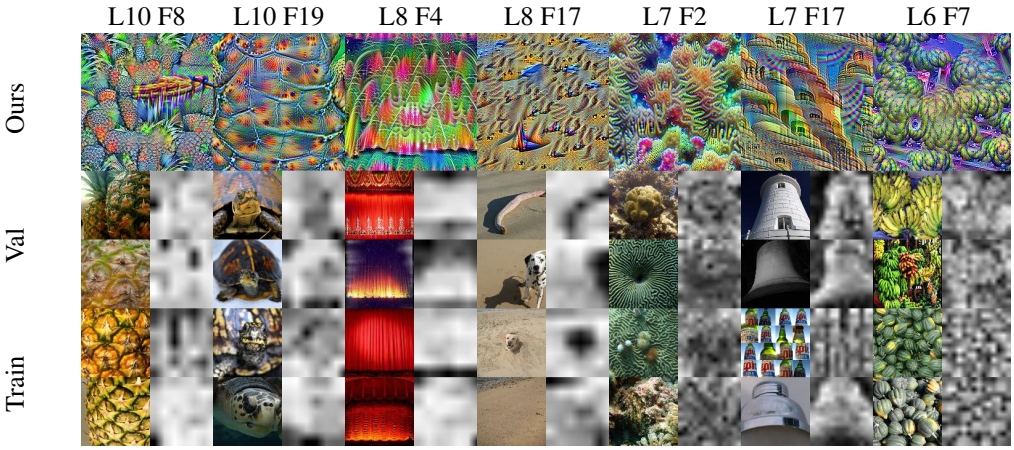


Figure 45: Visualization of features in Pit small distilled im-224.

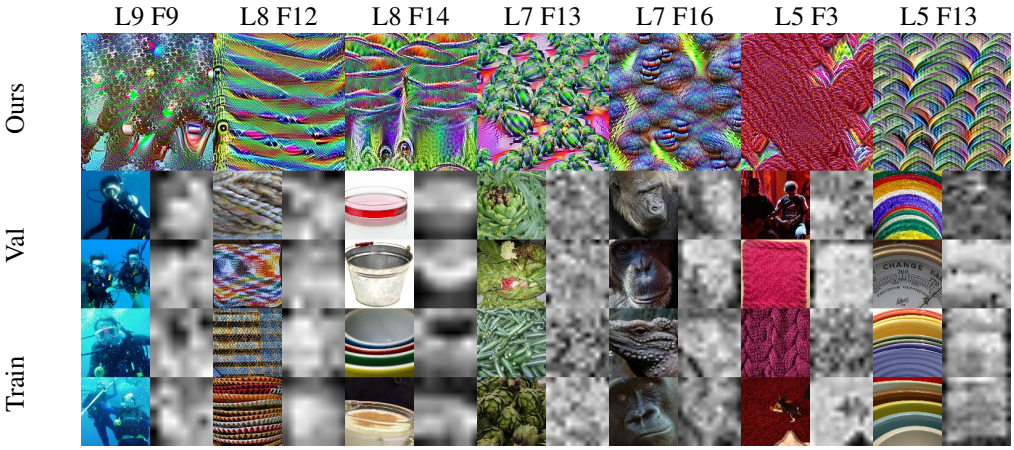


Figure 46: Visualization of features in Pit tiny im-224.

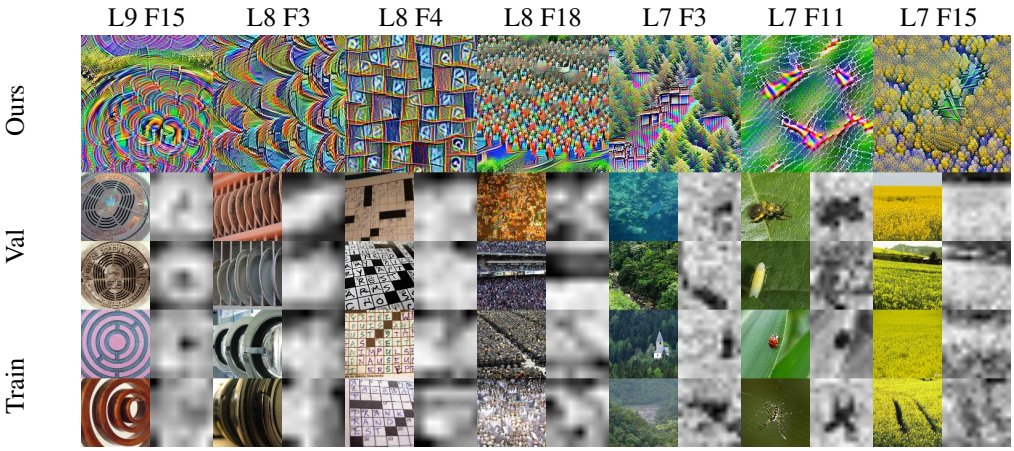


Figure 47: Visualization of features in Pit tiny distilled im-224.

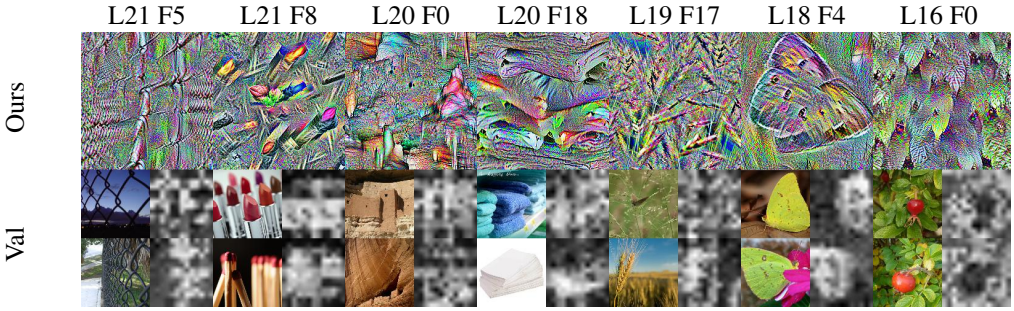


Figure 48: Visualization of features in Swin base p-4 w-7 im-224.

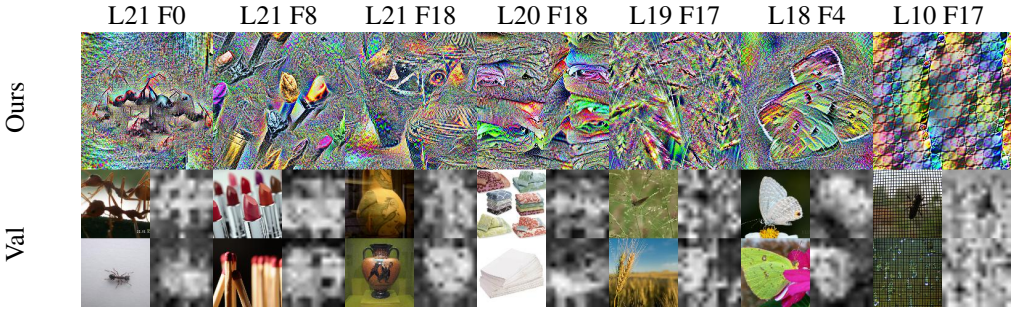


Figure 49: Visualization of features in Swing base base p-4 w-7 im-224 imagenet 22k.

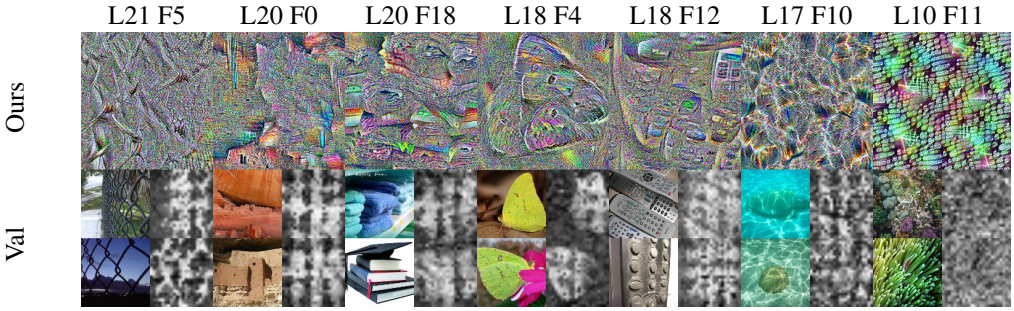


Figure 50: Visualization of features in Swin base p-4 w-12 im-384.

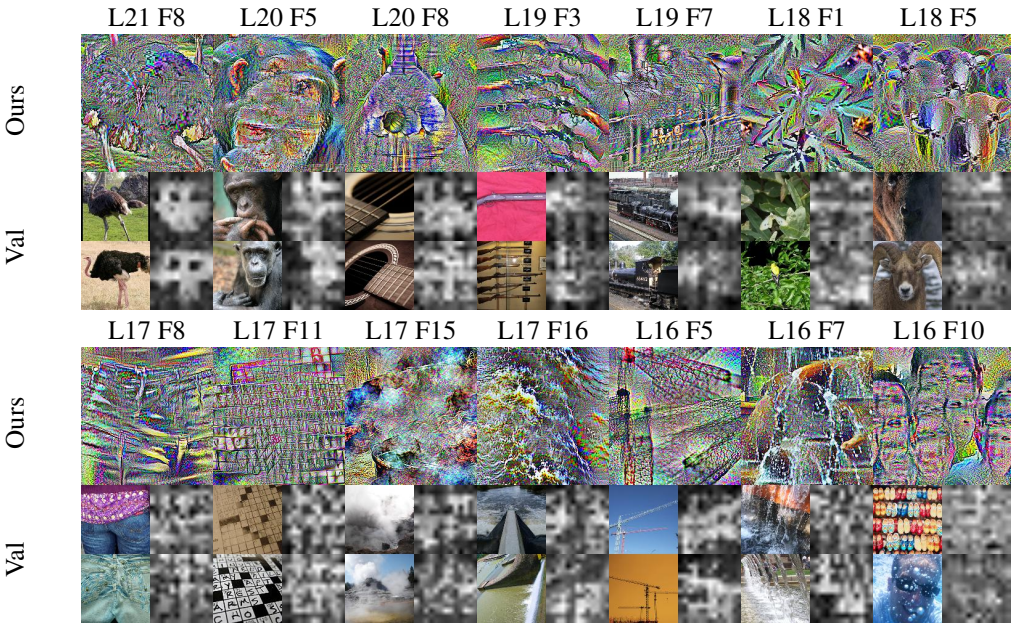


Figure 51: Visualization of features in Swin large p-4 w-7 im-224.

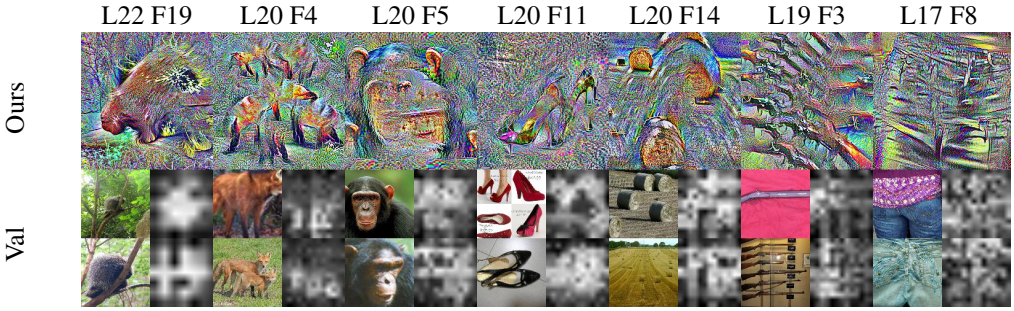


Figure 52: Visualization of features in Swin large p-4 w-7 im-224 imagenet 22k.

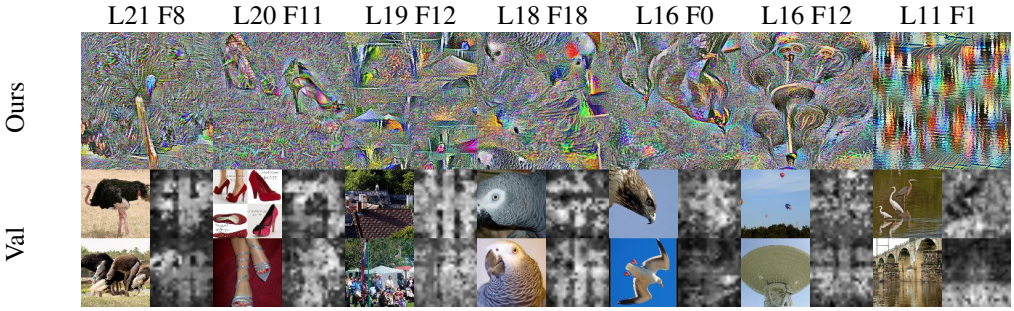


Figure 53: Visualization of features in Swin large p-4 w-12 im-384.

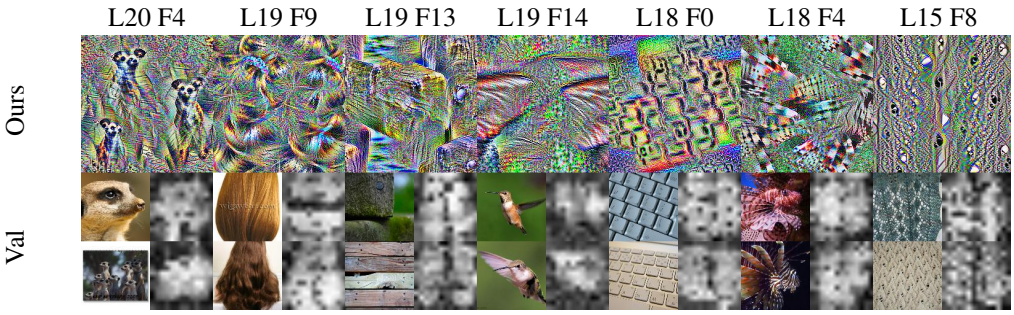


Figure 54: Visualization of features in Swin small p-4 w-7 im-224.

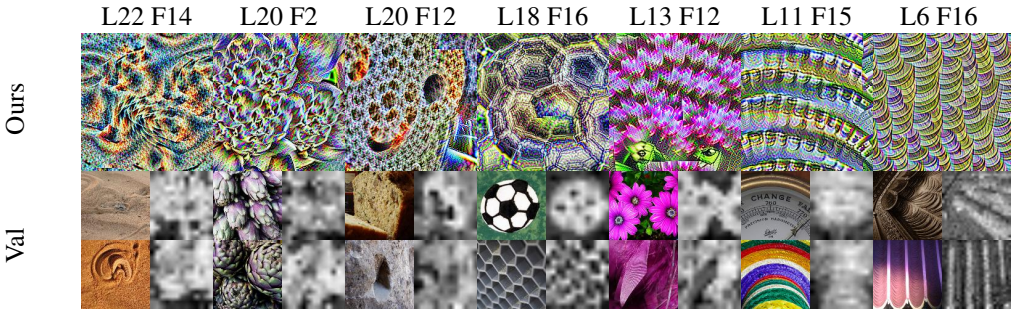


Figure 55: Visualization of features in Twins pcpvt base.

Figure 56: Visualization of features in Twins pcpvt large.

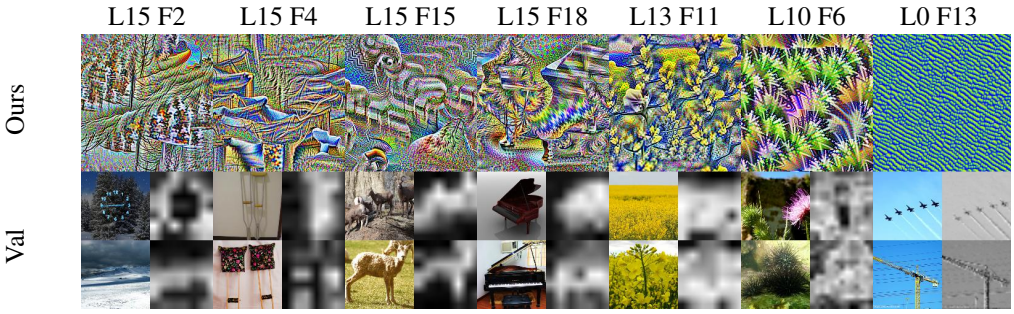


Figure 57: Visualization of features in Twins pcpvt small.

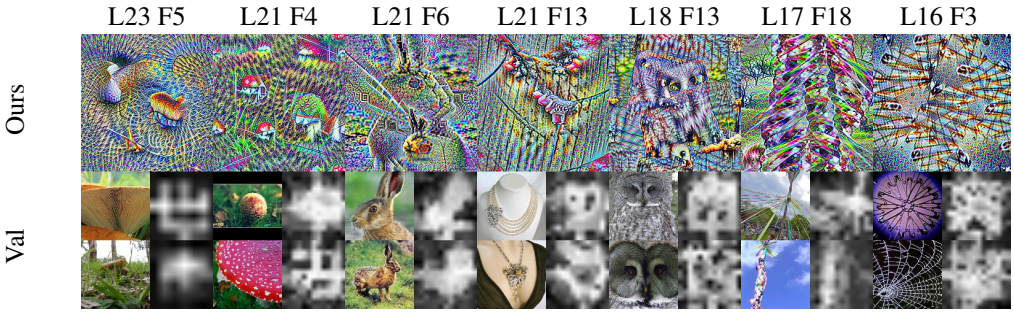


Figure 58: Visualization of features in Twins svt base.

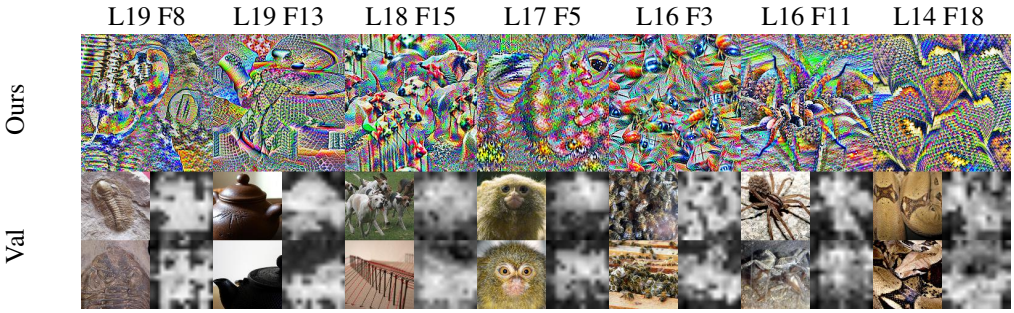


Figure 59: Visualization of features in Twins svt large.

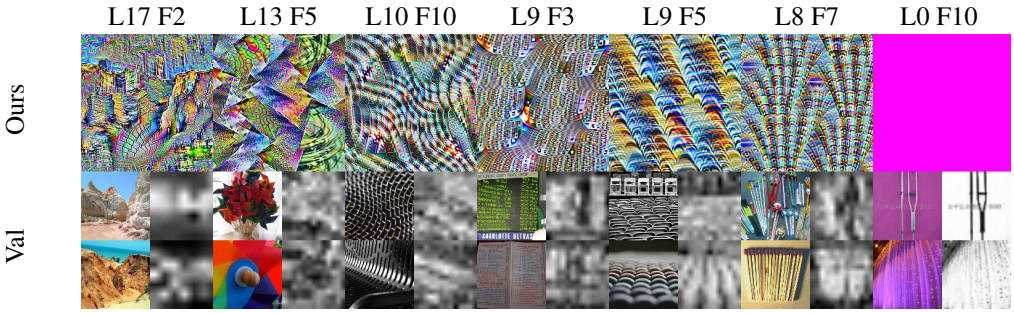


Figure 60: Visualization of features in Twins svt small.

Table 3: **ViTs more effectively correlate background information with correct class.** Both foreground and background data are normalized by full image top-1 accuracy.

Normalized Top-1 ImageNet Accuracy			
Architecture	Full Image	Foreground	Background
ViT-B32	89.25	91.53	15.04
ViT-L16	95.00	93.88	**19.08**
ViT-L32	94.64	90.63	17.67
ViT-B16	92.37	93.70	16.98
ResNet-50	87.67	85.59	9.25
ResNet-152	82.92	82.03	8.24
MobileNetv2	83.77	85.58	8.75
DenseNet121	90.58	86.53	**9.72**