Appendix

A Proof of Proposition 3.1

Proof. Since we are dealing with an objective, we can drop constants, which do not depend on the embedding $\vec{z}_1$ and $\vec{z}_2$, when they occur.

$$L = \int p(\vec{x}_1)p(\vec{x}_2)\left[w\vec{z}_1^T\vec{z}_2 - \frac{p(\vec{x}_1,\vec{x}_2)}{p(\vec{x}_1)p(\vec{x}_2)}\right]^2 d\vec{x}_1 d\vec{x}_2 \tag{3}$$

$$= \int p(\vec{x}_1)p(\vec{x}_2)\left[(w\vec{z}_1^T\vec{z}_2)^2 - 2w\vec{z}_1^T\vec{z}_2 \cdot \frac{p(\vec{x}_1,\vec{x}_2)}{p(\vec{x}_1)p(\vec{x}_2)}\right] d\vec{x}_1 d\vec{x}_2 \tag{4}$$

$$= \int p(\vec{x}_1)p(\vec{x}_2)(w\vec{z}_1^T\vec{z}_2)^2 d\vec{x}_1 d\vec{x}_2 - 2w\int p(\vec{x}_1,\vec{x}_2)(\vec{z}_1^T\vec{z}_2)d\vec{x}_1 d\vec{x}_2 \tag{5}$$

$$= \mathbb{E}_{p(\vec{x}_1,\vec{x}_2)}\left[-\vec{z}_1^T\vec{z}_2\right] + \lambda\mathbb{E}_{p(\vec{x}_1)p(\vec{x}_2)}\left(\vec{z}_1^T\vec{z}_2\right)^2 \tag{6}$$

where $\lambda = \frac{w}{2}$. □

In practice, we can simply choose $w = 2$ so that $\lambda = 1$.

B The Duality Between Contrastive and Non-Contrastive SSL

The similarity term in different joint-embedding methods is essentially the same, and we focus on the regularization term, particularly with SGD optimizer. For simplicity, we assume that the embedding $\vec{z}$ is L2-normalized, and each of the embedding dimensions also has zero mean and normalized variance. Given a minibatch with size N, the spectral regularization term $\mathbb{E}_{p(\vec{x}_1)p(\vec{x}_2)}\left(\vec{z}_1^T\vec{z}_2\right)^2$ reduces to $\left\|Z^TZ - I_d\right\|_F^2$. By Lemma 3.2 from Le et al. (2011), we have:

$$\left\|Z^TZ - I_N\right\|_F^2 = \left\|ZZ^T - I_d\right\|_F^2 = \left\|ZZ^T - \frac{N}{d}I_d\right\|_F^2 + C \tag{7}$$

where C is a constant. The third equality follows due to that each of the embedding dimensions is normalized. $\left\|ZZ^T - \frac{1}{d}I_N\right\|_F^2$ is the mini-batch version of the covariance regularization term $\mathbb{E}_{p(\vec{x})}\left[\vec{z}\vec{z}^T\right] = \frac{N}{d_{emb}} \cdot I$.

For a thorough discussion on the duality between contrastive learning and non-contrastive learning, we refer the curious readers to (Garrido et al., 2023).

C Implementation Details

C.1 CIFAR-10 and CIFAR-100

For all experiments and methods, we pretrain a ResNet-34 for 600 epochs. We use a batch size of 1024, LARS optimizer, and a weight decay of $1e-04$. The learning rate is set to 0.3 and follows a cosine decay schedule, with 10 epochs of warmup and a final value of 0. In the TCR loss, λ is set to 30.0, and ϵ is set to 0.2. In the SimCLR loss, we set the temperature to 0.1. In the VICReg loss, we set the coefficients for variance/invariance/covariance to 25, 25, and 1, respectively. For all methods, the projector network consists of 2 linear layers with respectively 4096 hidden units and 128 output units for the CIFAR-10 experiments and 512 output units for the CIFAR-100 experiments. All the layers are separated with a ReLU and a BatchNorm layers. For all the methods, the data augmentations used are identical to those described in BYOL Grill et al. (2020).

C.2 ImageNet-100

For all the experiments and methods, we pretrain a ResNet-50 for 400 epochs. We use a batch size of 1024, the LARS optimizer, and a weight decay of $1e-04$. The learning rate is set to 0.1 and follows a cosine

decay schedule, with 10 epochs of warmup and a final value of 0. In the TCR loss, λ is set to 1920.0, and ϵ is set to 0.2. In the SimCLR loss, we set the temperature to 0.1. In the VICReg loss, we set the coefficients for variance/invariance/covariance to 25, 25, and 1, respectively. For all methods, the projector network consists of 3 linear layers, with each and 8192 units separated by a ReLU and a BatchNorm layers. For all the methods, the data augmentations used are identical to those described in BYOL Grill et al. (2020).

C.3 ImageNet

For all the experiments, we pre-train a ResNet-50 for 800 epochs with VICReg Bardes et al. (2021). We use a batch size of 1024, the LARS optimizer, and a weight decay of $1e-06$. The learning rate is set to 0.1 and follows a cosine decay schedule, with 10 epochs of warmup and a final value of 0. We set the coefficients for variance/invariance/covariance to 25, 25, and 1, respectively. The projector network consists of 3 linear layers, with each and 8192 units, separated by a ReLU and a BatchNorm layers. The data augmentations used are identical to those described in BYOL Grill et al. (2020).

C.4 Implementation Detail for Local Aggregation

For all the experiments, we downloaded the checkpoints of the SOTA SSL model pretrained on the CIFAR10 dataset from solo-learn. Each method is pretrained for 1000 epochs, and the hyperparameters used for each method are described in solo-learn. The backbone model used in all these checkpoints is ResNet-18, which output a dimension $512 \times 5 \times 5$ tensor for each image. We apply spatial average pooling (stride = 2, window size = 3) to this tensor and flatten the result to obtain a feature vector of dimension 2048.

D ImageNet Visualization

In this section, we provide further visualization of the multi-scale pretrained VICReg model, and the results are shown in Fig 6. Here we use image patches of scale 0.1 to calculate the cosine similarity heatmaps. The query patch is marked by the red-dash boxes. The embedding space contains more localized information, whereas the projection space is relatively more invariant, especially when the patch has enough information to determine the category.

E CIFAR10 kNN Visualization

This section continues additional visualization of the model pretrained with 14×14 patches on CIFAR-10. In this visualization, we use cosine similarity to find the closest neighbors for the query patches, marked in the red-dash boxes. Again, green boxes indicate that the patches are from other images of the same category; red boxes indicate that the patches are from other images of a different category. Patches that do not have a colored box are from the same image. In the following, we discuss several aspects of the problem.

Additional Projection and Embedding Spaces Comparison. As we can see in Figure 7, the embedding space has a lesser degree of information collapse. The projection space tends to collapse different "parts" of a class to similar vectors, whereas the embedding space preserves more information about the locality or details in a patch. This is manifested by higher visual similarity between neighboring patches.

Embedding Space with 256 kNN. In the previous CIFAR visualization, we only show kNN with 119 neighbors. In Figure 8 and Figure 9, we provide kNN with 255 neighbors, and the same conclusions hold.

Different "Parts" in the Embedding Space. In Figure 10, we provide more typical patches of "parts" and show their embedding neighbors. While many parts are shared by different instances, we also find several less ideal cases, e.g., Figure 10(4a)(2d), where the closest neighbors are nearly all from the same image.

As we discussed earlier, the objective is modeling patches' co-occurrence statistics. It is relatively uninformative if the same patch is not "shared" by different instances. While the same patch might not be "shared", the color augmentation and deep image prior embedded in the network design may create approximate sharing. In Figure 11 and Figure 12, we provide two examples of the compositional structure of instances.

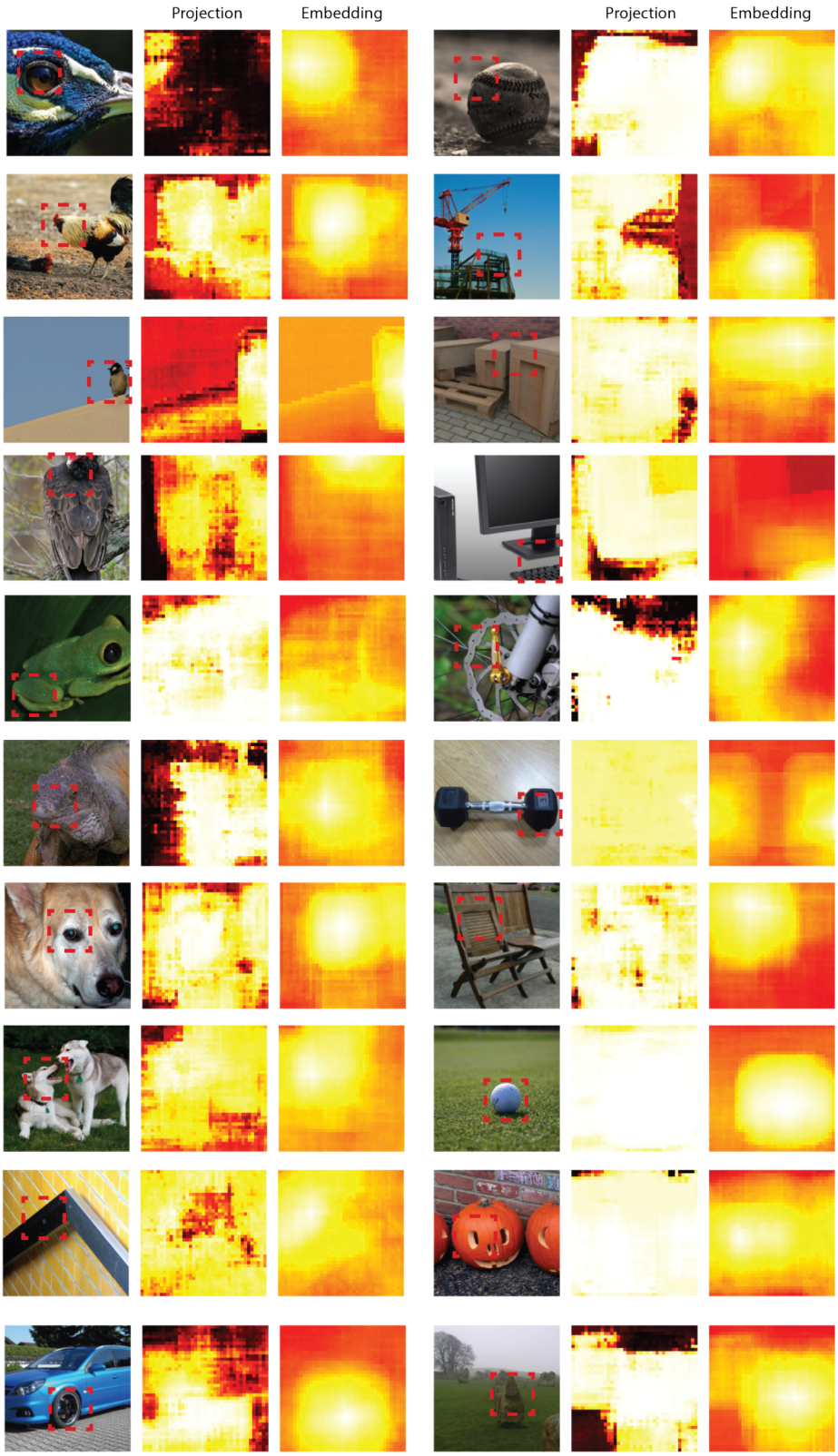

Figure 6: **More visualization of cosine similarity heatmaps in the projection space and the embedding space.** Here, the query patch is marked by the red-dash boxes, and its size is 71×71, and the instance image size is 224×224.

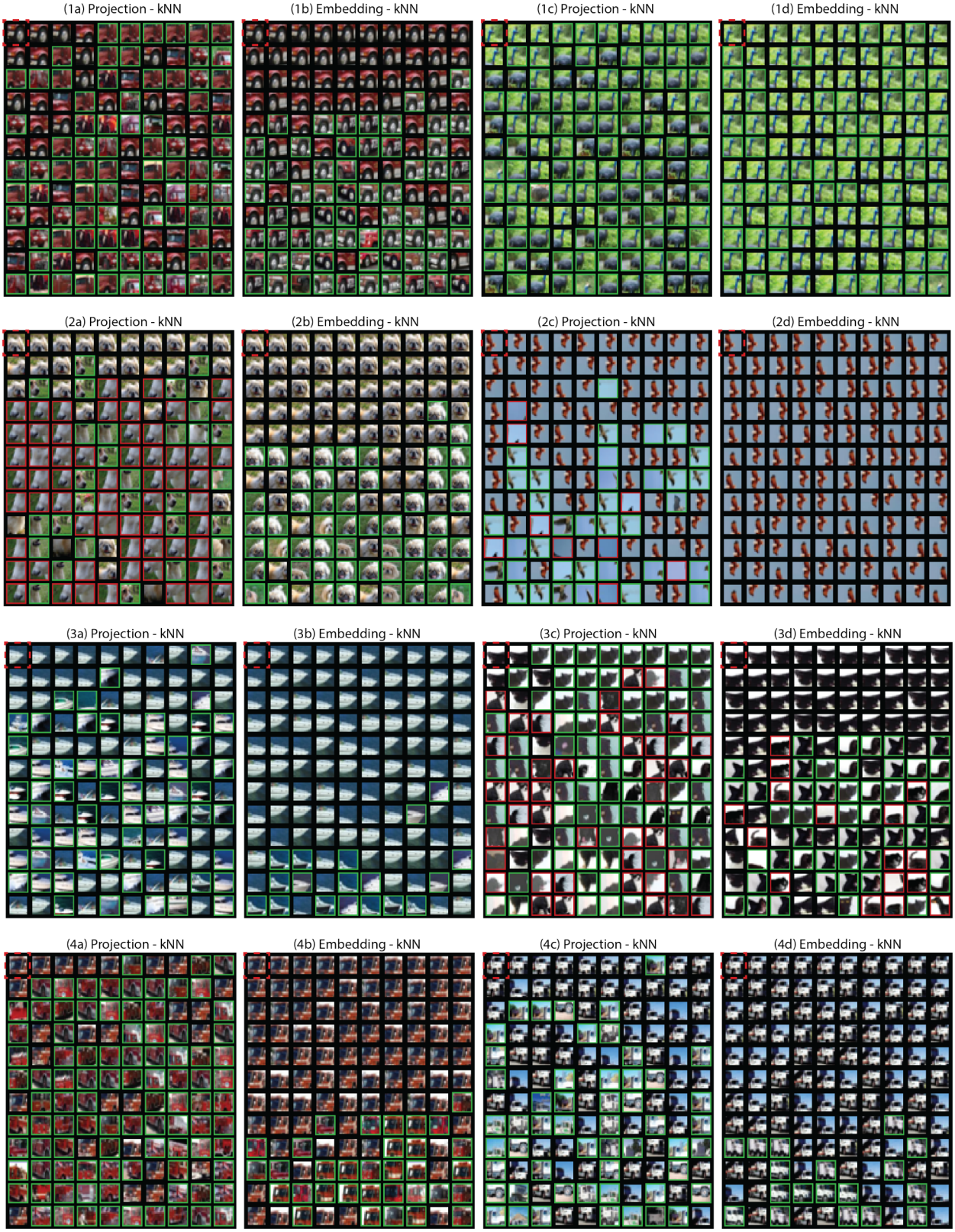

Figure 7: **Additional comparison between the projection space and the embedding space.**

Figure 8: **kNN in the embedding Space with 255 neighbors.**

Figure 9: **kNN in the embedding Space with 255 neighbors.**

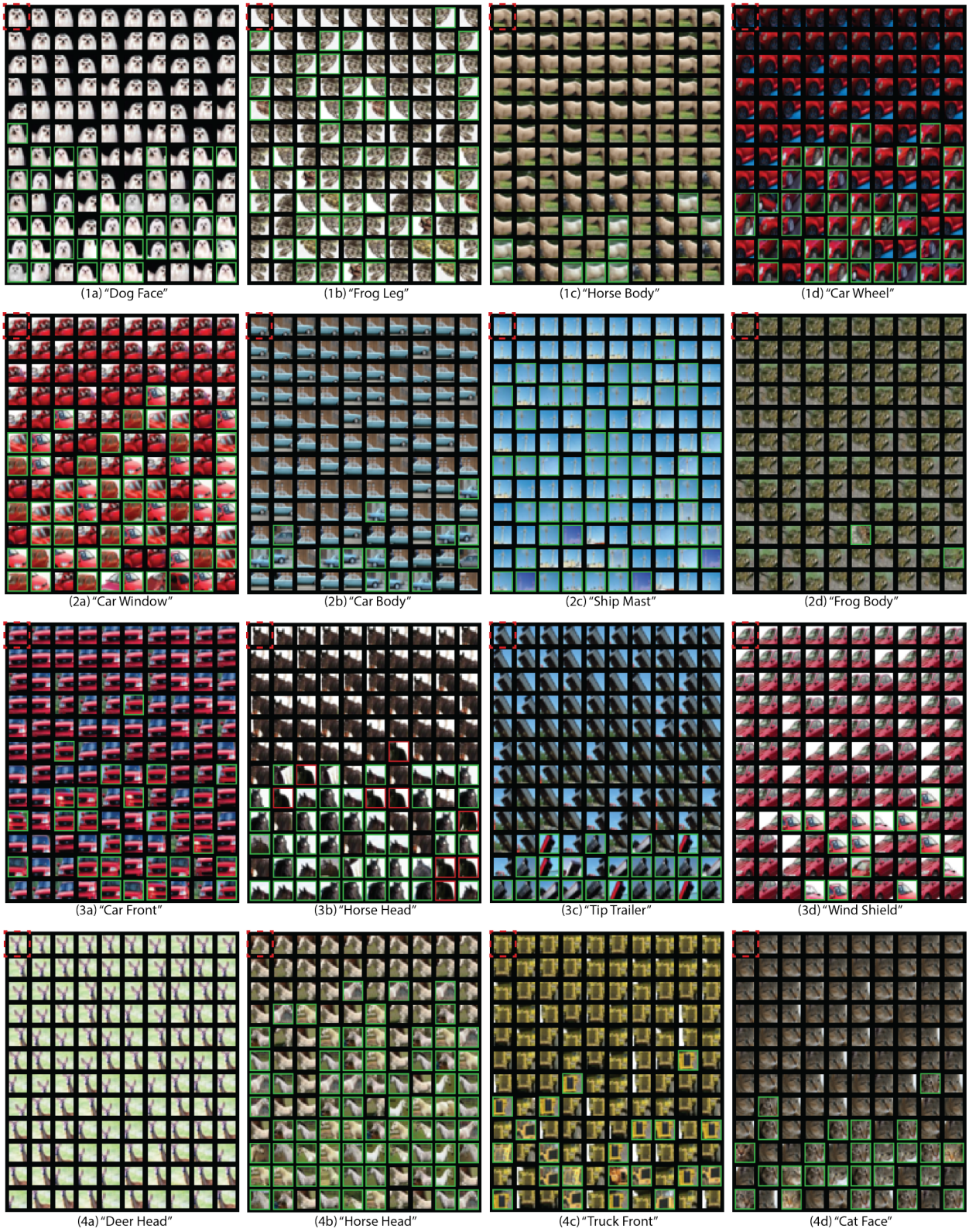

Figure 10: **Different "parts" in the embedding space.**

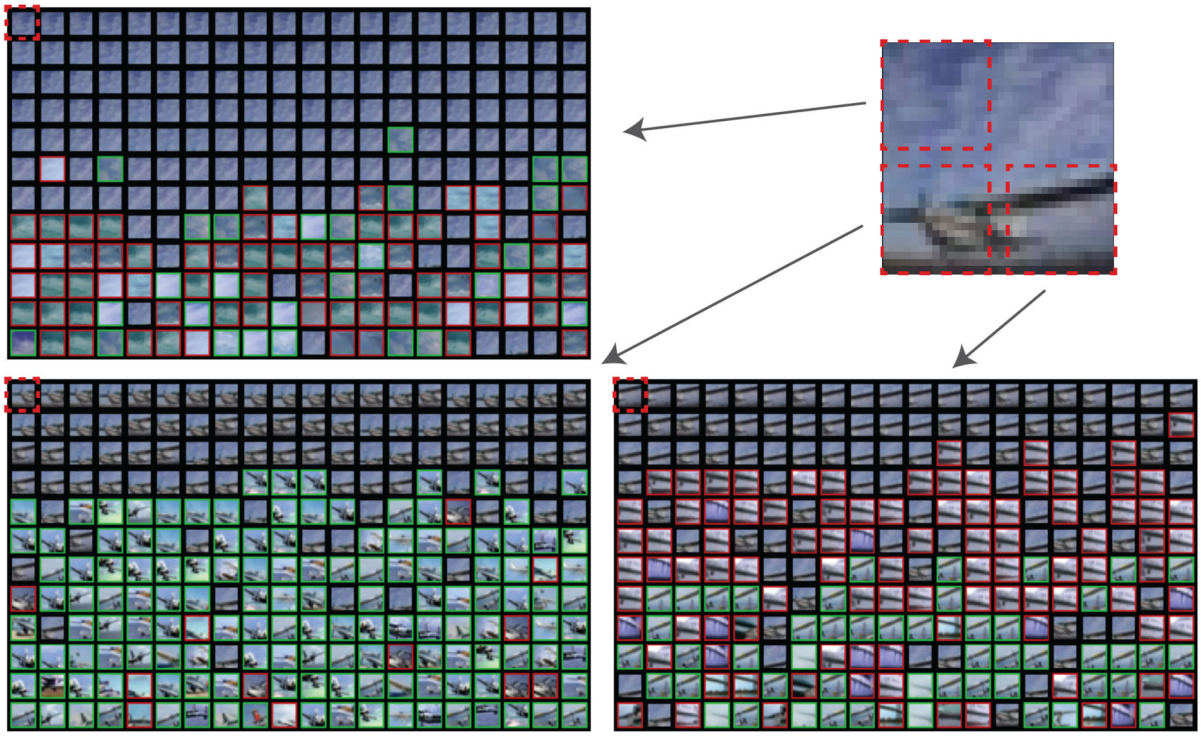

Figure 11: **The compositional structure of an airplane.** The "sky" part is shared by ships, birds, etc. The "wing" resembles the silhouettes of ships and is also shared by flying birds. The airscrew part is primarily shared by the other airplanes.

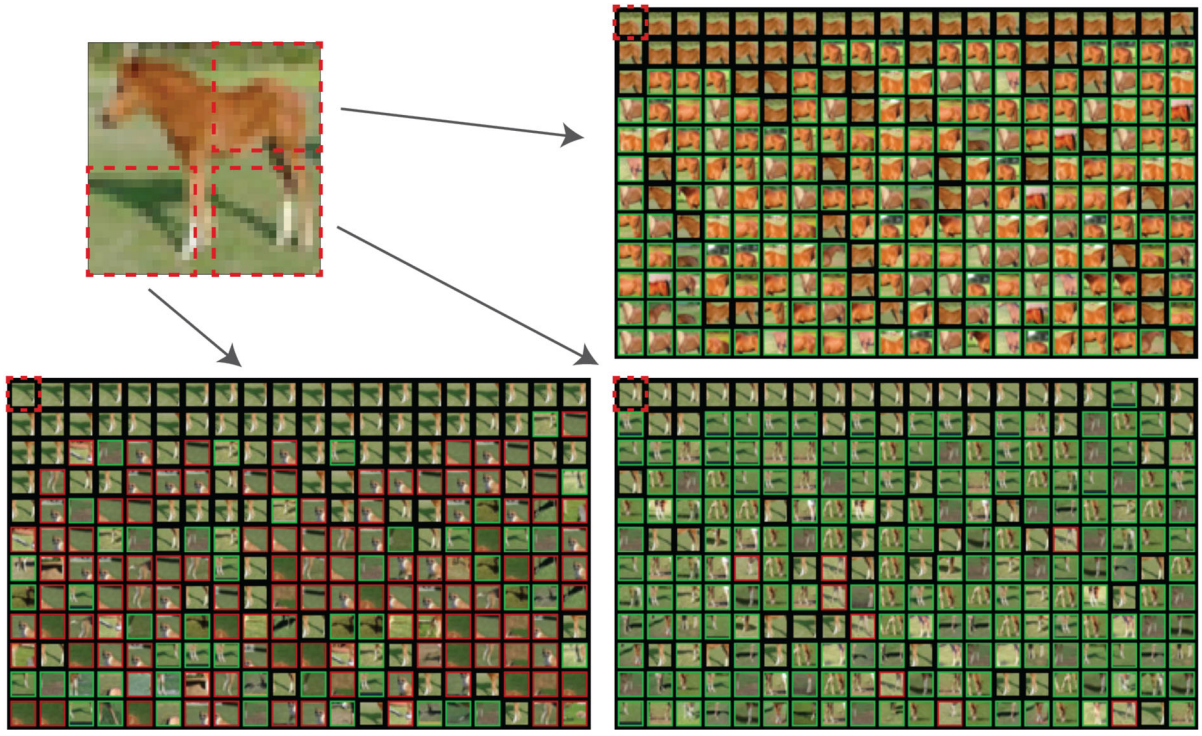

Figure 12: **The compositional structure of a horse.** The bottom left corner contains "shadow", and similar shadows are shared by deer and dogs. The bottom right part contains "legs", which are also shared by deer and dogs. However, from the back to the thigh is shared by primarily other horses.