

# A Geometric Analysis of Logit Embeddings for Out-of-Distribution Detection

## APPENDIX: TABLE OF CONTENTS

• Appendix A: Initialized network	13
• Appendix B: In-distribution during training	13
• Appendix C: Models for toy example	16
• Appendix D: Experiments on ImageNet-1K	17
• Appendix E: Experimentation on CIFAR-100 (ID) vs CIFAR-10 (OOD)	19
• Appendix F: Detailed visualization of density plots on individual logit cells	20
• Appendix G: Experiments on different dropout rate	30
• Appendix H: Experiments on different classifiers	40
• Appendix I: Experiments on different vision transformers	60
• Appendix J: Additional experimentation on grayscale image	69

## A INITIALIZED NETWORK

**Assumption 1** (Distributional Separation). *Let input data  $x \sim \mathcal{P}_x$  and model initial weights  $\omega \sim \mathcal{P}_\omega$  be drawn from distinct distributions with*

$$\text{supp}(\mathcal{P}_x) \cap \text{supp}(\mathcal{P}_\omega) = \emptyset \quad (3)$$

**Proposition 1** (Covariance Bound). *Under Assumption 1, the covariance satisfies:*

$$|\text{Cov}(x, \omega)| \leq \epsilon$$

for small  $\epsilon > 0$ .

*Proof.* The disjoint support in eq. (3) almost surely makes their covariance small.  $\square$

**Corollary 1.** *For zero-centered initialization ( $\mathbb{E}[\omega] = 0$  as in Glorot & Bengio (2010)):*

$$\begin{aligned} |\text{Cov}(x, \omega)| &\leq \epsilon \\ \left| \mathbb{E}[\langle x, \omega \rangle] - \mathbb{E}[x] \underbrace{\mathbb{E}[\omega]}_{\mathbb{E}[\omega]=0} \right| &\leq \epsilon \\ |\mathbb{E}[\langle x, \omega \rangle]| &\leq \epsilon \end{aligned}$$

Thus, the expected logit magnitude for an initialized DL classifier is  $\mathcal{O}(\epsilon)$  small.

## B IN-DISTRIBUTION DURING TRAINING

**Proposition 2.** *Consider a classification model with a linear transformation from a non-negative embedding space  $\vec{\mathcal{E}} \in \mathbb{R}_+^d$  to the logit space  $\vec{\mathbb{L}} \in \mathbb{R}^C$  such that  $\vec{\mathbb{L}} = \vec{\mathcal{E}}\mathcal{W}$ , where  $\mathcal{W} \in \mathbb{R}^{d \times C}$  is the weight matrix and  $C$  is the number of classes. Under the following conditions:*

(i) **Non-negativity Constraint:** *Embeddings  $\vec{\mathcal{E}}(i) \geq 0 \forall i$ .*

(ii) **Objective:** *The optimizer maximizes the logit  $\vec{\mathbb{L}}[i]$  for the correct class  $i$  while minimizing  $\vec{\mathbb{L}}[j \neq i]$  for all other classes,*

Then:

- (a) **Alignment of Correct Class:** The optimization  $\arg \max_{\mathcal{W}[i,:]} \langle \vec{\mathcal{E}}, \mathcal{W}[i,:] \rangle$  minimizes the angle  $\theta_i$  between  $\mathcal{W}[i,:]$  and the data cluster  $\vec{\mathcal{E}}$  belonging to class  $i$  (see fig. 10a), i.e.,

$$\theta_i = \angle(\mathcal{W}[i,:], \vec{\mathcal{E}}) \rightarrow 0, \quad \text{s.t. } \vec{\mathcal{E}} \geq 0 \quad (4)$$

- (b) **Suppression of Incorrect Classes:** The optimization  $\arg \min_{\mathcal{W}[j \neq i,:]} \langle \vec{\mathcal{E}}, \mathcal{W}[j \neq i,:] \rangle$  maximizes the angle  $\theta_j$  between  $\mathcal{W}[j \neq i,:]$  and  $\vec{\mathcal{E}}$  (see fig. 10a), approaching orthogonality asymptotically:

$$\theta_j \rightarrow \pi/2, \quad \text{and thus } \vec{\mathbb{L}}[j \neq i] \rightarrow 0 \quad (5)$$

- (c) **Cluster Geometry:** The logit-space configuration yields compact class clusters with maximized inter-class angular separation and minimized intra-class variance, pushing these class-wise clusters toward the positive region of the logit space (see fig. 10b).

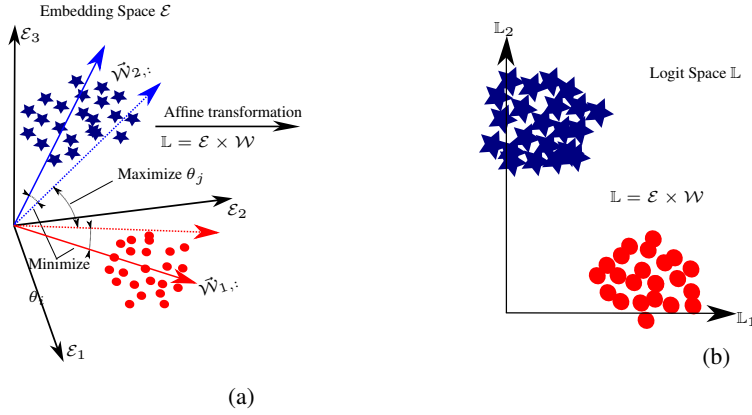


Figure 10: This toy example shows the separation of ID in a binary classification task. Figure 10a contains the embeddings ( $\vec{\mathcal{E}}$ ) rectified with a ReLU. Figure 10b shows the linear separation of class-wise clustering of ID data logits ( $\vec{\mathbb{L}}$ ). The smaller the angle between red cluster  $\vec{\mathcal{E}}$  and  $\vec{\mathcal{W}}_{1,:}$ , the higher the dot product  $\langle \mathcal{W}_{1,i}, \mathcal{E}_i \rangle$ ; thus, the more distant from the center the ID logits are (see fig. 10b). The bigger the angle between red cluster ( $\vec{\mathcal{E}}$ ) and  $\vec{\mathcal{W}}_{2,:}$ , the higher the dot product  $\langle \vec{\mathcal{W}}_{2,i}, \vec{\mathcal{E}}_i \rangle$  (see fig. 10a), the more compact the ID logits are. During training  $\vec{\mathcal{W}}_{2,i}$  is pushed into the blue cluster and away from both  $\vec{\mathcal{W}}_{1,i}$  and red cluster (see fig. 10a). Simultaneously,  $\vec{\mathcal{W}}_{1,i}$  is pushed into the red cluster and away from both  $\vec{\mathcal{W}}_{2,i}$  and blue cluster (see fig. 10a).

*Proof sketch.* We formalize the asymptotic behavior of logits under non-negative embedding constraints.

#### Preliminaries:

- Embeddings  $\vec{\mathcal{E}} \in \mathbb{R}_+^d$  (non-negative orthant, per fig. 10a)
- Logit computation:  $\vec{\mathbb{L}} = \vec{\mathcal{E}}\mathcal{W}$  where  $\mathcal{W} \in \mathbb{R}^{d \times C}$  where  $C$  is the number of classes.
- For class  $i$ , the target logit is  $\vec{\mathbb{L}}[i] = \langle \vec{\mathcal{E}}, \mathcal{W}[i,:] \rangle$

#### Optimization Dynamics:

- *Correct Class ( $i$ )*: The optimizer maximizes:

$$\max_{\mathcal{W}[i,:]} \langle \vec{\mathcal{E}}, \mathcal{W}[i,:] \rangle \quad \text{s.t.} \quad \vec{\mathcal{E}} \geq 0 \quad (6)$$

This minimizes the angle  $\theta_i$  between  $\vec{\mathcal{E}}$  belonging to class  $i$  and  $\mathcal{W}[i,:]$ , aligning them in the non-negative orthant (i.e., blue arrow inside the blue cluster in fig. 10a).

- *Incorrect Classes ( $j \neq i$ )*: The optimizer minimizes:

$$\min_{\mathcal{W}[j,:]} \langle \vec{\mathcal{E}}, \mathcal{W}[j,:] \rangle \quad \text{s.t.} \quad \vec{\mathcal{E}} \geq 0 \quad (7)$$

This maximizes the angle  $\theta_j$  toward  $90^\circ$ , making  $\vec{\mathcal{E}}$  belonging to class  $i$  and  $\mathcal{W}[j,:]$  (i.e., red arrow away from the blue cluster in fig. 10a) asymptotically orthogonal (see Corollary 2).

### Logit Constraints:

- As  $\theta_j \rightarrow 90^\circ$ , by dot product properties:

$$\vec{\mathbb{L}}[j] = \|\vec{\mathcal{E}}\| \|\mathcal{W}[j,:]\| \cos \theta_j \rightarrow 0 \quad (8)$$

- For  $\theta_i \rightarrow 0^\circ$ ,  $\vec{\mathbb{L}}[i]$  is maximized as  $\cos \theta_i \rightarrow 1$ .

### Geometric Interpretation:

- Cluster tightness: Intra-class  $\vec{\mathcal{E}}$  embeddings cluster around  $\mathcal{W}[i,:]$  (minimized  $\theta_i$ )
- Inter-class separation:  $\mathcal{W}[j,:]$  vectors are pushed toward the orthogonal complement of  $\{\vec{\mathcal{E}}\}_{i \in \text{class}}$  (see fig. 10a)

**Conclusion:** The non-negativity constraint forces incorrect-class logits to vanish asymptotically ( $\vec{\mathbb{L}}[j \neq i] \rightarrow 0$ ) while promoting maximal separation between class clusters in logit space.  $\square$

**Corollary 2.** *In the non-negative orthant of  $\mathbb{R}_+^N$ , the maximum attainable angle between any two vectors is  $90^\circ$  (perpendicularity).*

*Proof.* Consider two vectors  $X, Y \in \mathbb{R}_+^N$  where  $X_i, Y_i \geq 0$  for all  $i \in [1, N]$ . The cosine similarity between them is:

$$\cos \theta_{X,Y} = \frac{\langle X, Y \rangle}{\|X\| \|Y\|} = \frac{\sum_{i=1}^N X_i Y_i}{\|X\| \|Y\|} \geq 0 \quad (9)$$

Since  $X_i, Y_i \geq 0$ , their dot product  $\langle X, Y \rangle$  is non-negative. Thus:

- $\cos \theta_{X,Y} \geq 0$  implies  $\theta_{X,Y} \in [0^\circ, 90^\circ]$
- The angle is maximized when  $\cos \theta_{X,Y} = 0$ , which occurs iff  $\langle X, Y \rangle = 0$  (orthogonality)

$\square$

Note that non-negative embedding constraints do not necessitate strictly non-negative weights post-training. Negative activations yield negligible gradients, leaving negative-weight updates sparse. Consequently, positive weights predominantly influence training dynamics, aligning with established insights into activation sparsity and gradient updates.

## C MODELS FOR TOY EXAMPLE

The training configuration for the model outlined in tables 1 and 2 includes a batch size of 64, a learning rate of 0.001, and 30 training epochs. To combat overfitting, a dropout rate of 0.8 is employed.

Table 1: Architecture of the MLP model without batch normalization.

Layer Type	Output Size	Additional Information
Linear	2048	in_features=3
ReLU	2048	-
Dropout	2048	p=0.8
Linear	2048	in_features=2048
ReLU	2048	-
Dropout	2048	p=0.8
Linear	2	in_features=2048

Table 2: Architecture of the MLP model with batch normalization.

Layer Type	Output Size	Additional Information
Linear	2048	in_features=3
BatchNorm	2048	in_features=3
ReLU	2048	-
Dropout	2048	p=0.8
Linear	2048	in_features=2048
BatchNorm	2048	in_features=2048
ReLU	2048	-
Dropout	2048	p=0.8
Linear	2	in_features=2048

## D EXPERIMENTS ON IMAGENET-1K

We evaluate several ViT (see fig. 11) and CNN (see fig. 12) variants pretrained on ImageNet-1K by generating logits for ID-in, ID-out, and OOD samples across all 1,000 classes, then overlay their kernel density estimate (KDE) curves for direct comparison. For the OOD dataset, we utilize the Places and Texture.

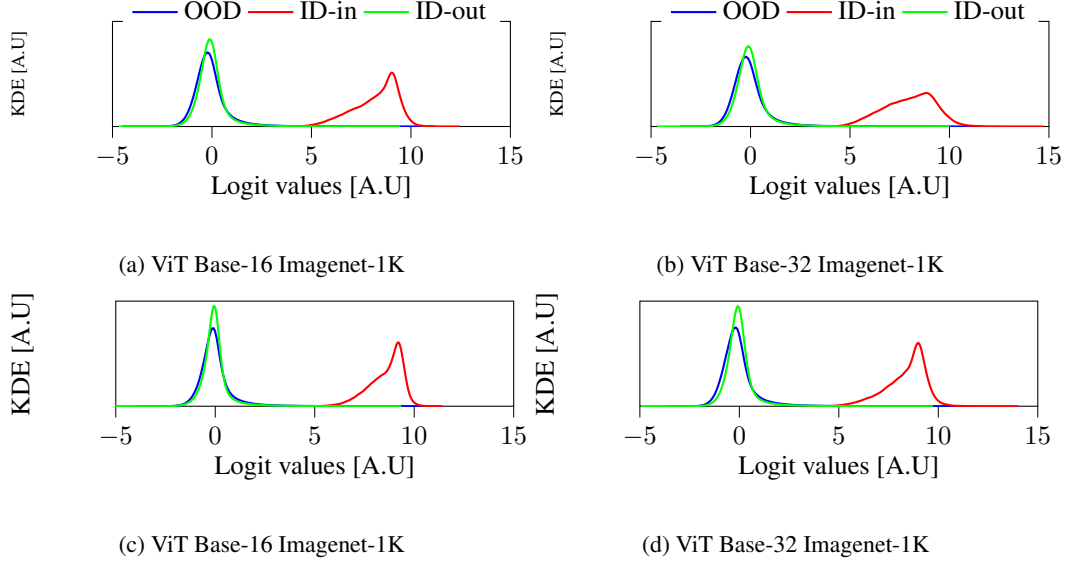


Figure 11: Experiments on different vision transformers using Imagenet-1K

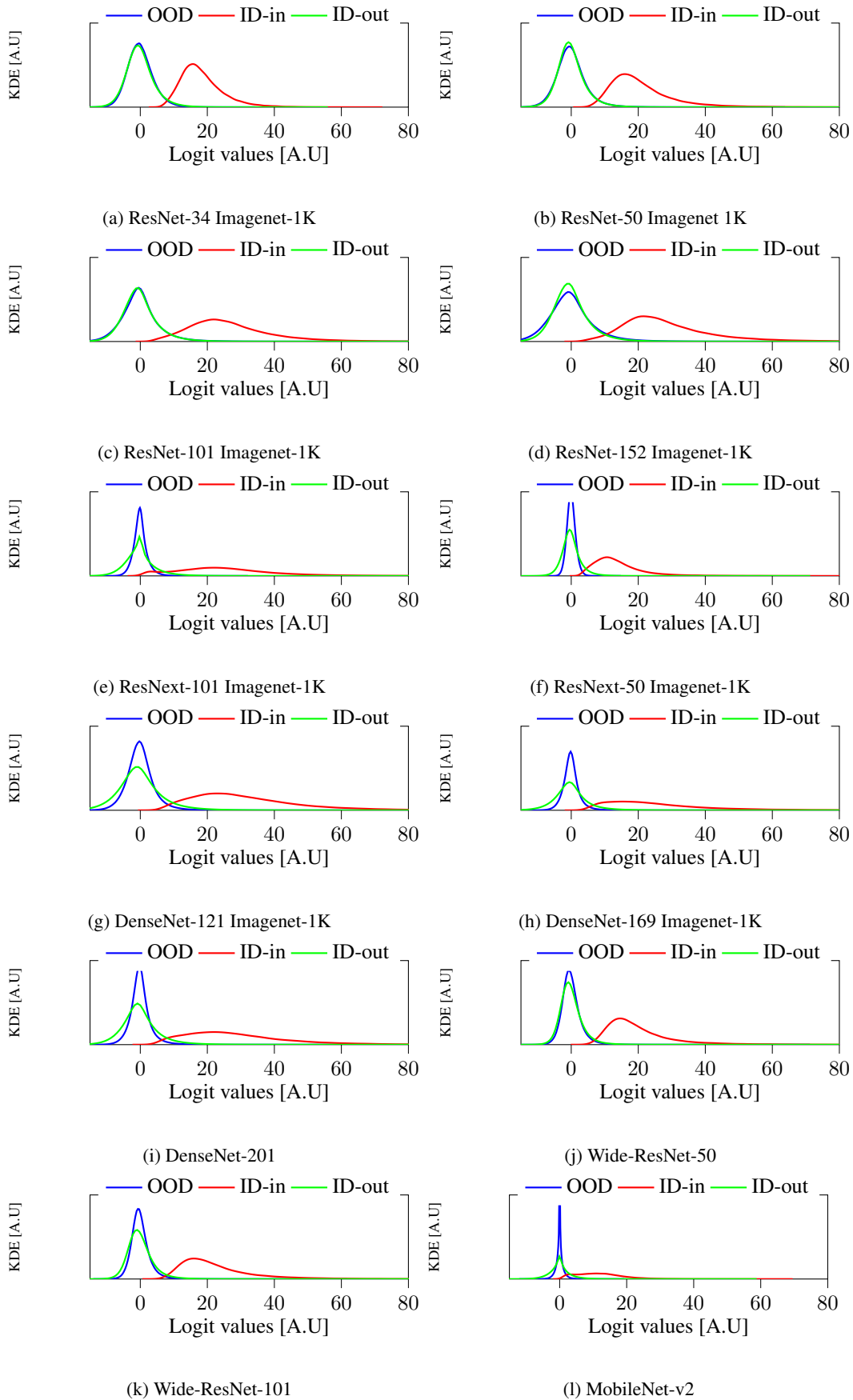


Figure 12: Experiments on different CNN using Imagenet-1K

## E EXPERIMENTATION ON CIFAR-100 (ID) vs CIFAR-10 (OOD)

Resnet-9 is trained using stochastic gradient descent (SGD) with a learning rate starting at  $lr = 10^{-1}$ . The batch size is 256, and the number of epochs is 200. The learning rate is decreased every quarter of epochs. Within each quarter, the learning rate is scheduled using a 1-cycle learning rate. ReLU is utilized as an activation function for every layer. No regularization is applied to the training process, while the training data are augmented with random flipping and cropping. In figs. 13a and 13b, we present the distributions of logit values for ID samples, with the former displaying densities corresponding to the ID-in and the latter for the ID-out. OOD densities are depicted for each logit in fig. 13c.

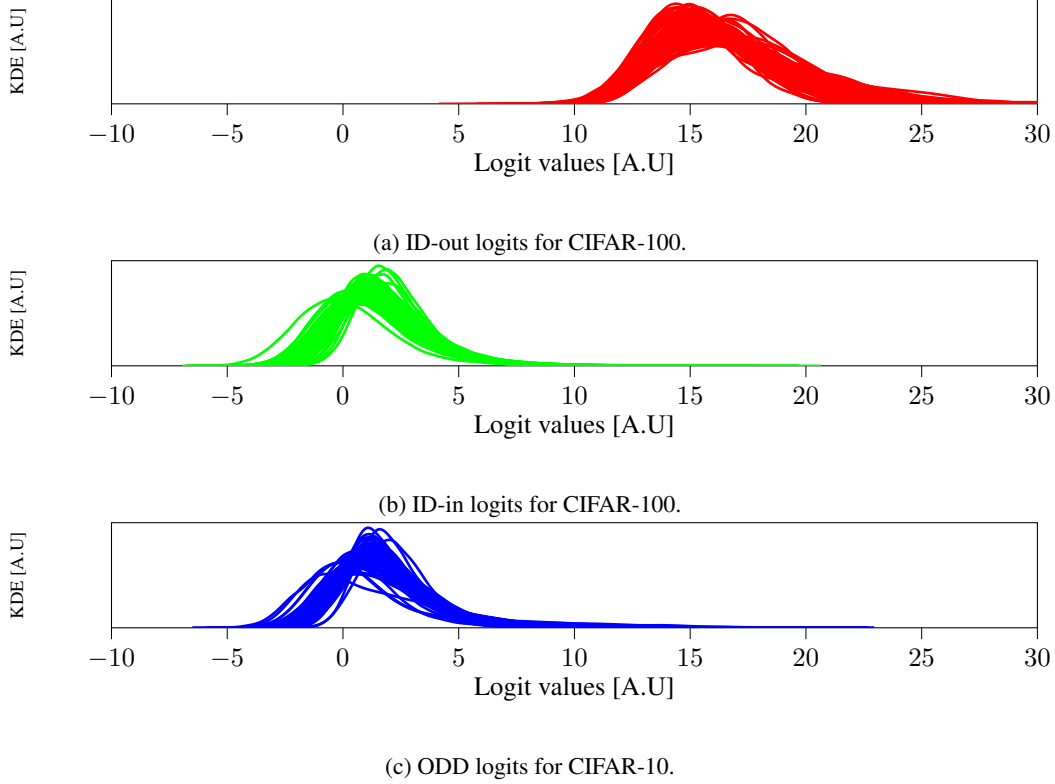


Figure 13: KDE response CIFAR-100 (ID) vs CIFAR-10 (OOD) while using Resnet-9 with ReLU activation function.

## F EFFECT OF ACTIVATION FUNCTION:

To further understand the configuration of ID and OOD logits, we investigate the impact of various activation functions on a ResNet-34 model. Specifically, we empirically observed this impact by utilizing a selection of activation functions known for their inherent suppression of negative values, including Celu, Elu, Gelu, Selu, Silu, Relu, Leaky-Relu, and Mish.

A ResNet-34 model was trained on the SVHN dataset Goodfellow et al. (2014) (i.e., ID data), utilizing each activation function (i.e., Celu, Elu, Gelu, Mish, Selu, Silu, ReLU, Leaky ReLU). The model is trained using stochastic gradient descent (SGD) with a cyclical learning rate starting at  $lr = 10^{-3}$  with a cosine annealing operation with a periodicity of 200. Furthermore, the momentum is 0.9 while the weight decay is  $5 \cdot 10^{-4}$ . A batch size of 256 is applied for both test and train data. No regularization is applied to the training process, while the training data are augmented with random flipping and cropping. Simultaneously, the CIFAR-10 dataset was used as OOD data.

One can notice that ID-in logits maintain a tendency towards high positive values across all the activation functions (see fig. 14). On the other hand, ID-out and OOD logits are predominantly centralized around the center (see fig. 14). Consequently, despite the application of varying nonlinearities, the relative configuration of ID and OOD logits remains similar. However, different activation functions yield varying degrees of separation and overlap between ID and OOD logits. ReLU produces the most compact logit distributions with minimal overlap, as it preserves positive activations while suppressing negative ones. In contrast, activation functions with nonlinear modulations (e.g., CeLU, ELU, SELU) induce a broader spread in logits and exhibit greater overlap between ID and OOD samples. This increased dispersion in ID logits from these non-linearities may stem from the introduction of spurious discriminative features parameterized by the DL classifier despite their absence in the training data. These spurious features contribute to higher variability in OOD logits, thereby increasing their overlap with ID distributions. Thus, unlike ReLU, transcendental activation functions (e.g., CeLU, ELU, SELU) increase computational overhead in training and inference while simultaneously diminishing the distinctiveness of ID and OOD logit distributions, potentially degrading OOD detection performance. Our heuristic explanation for the observed behavior relies on three main insights. First, at model initialization, the choice of activation has minimal effect, as all input samples initially produce logits concentrated near zero. Second, throughout training, the explicit objective to maximize the correct class logit naturally elevates its value while implicitly suppressing incorrect class logits toward lower magnitudes. Third, while negative activations can occur for some activations such as Leaky-ReLU, their magnitudes are disproportionately reduced, resulting in weaker gradient contributions towards negative logits. Consequently, logits for ID samples remain closely clustered around zero unless actively driven upward by training. Conversely, OOD logits maintain a similar centralized tendency, but for different reasons. Specifically, during training, the model explicitly learns translation-invariant, class-specific discriminative features inherent to ID data. When such critical features are absent in OOD samples, the learned features are not strongly activated. Hence, such OOD samples exhibit minimal covariance with class weights, consistently producing low-magnitude logits across all classes.

Figures 15 to 22 showcase a detailed visualization of the ID and OOD logits for each cell across different types of activation functions.

$$\text{Relu}: f(x) = \max(0, x)$$

$$\text{Celu}: f(x) = \max(0, x) + \min(0, \alpha(e^{x/\alpha} - 1))$$

$$\text{Elu}: f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases}$$

$$\text{GELU}: f(x) = x\Phi(x)$$

where  $\Phi(x)$  is the cumulative distribution function of the standard Gaussian distribution:

$$\Phi(x) = \frac{1}{2} \left[ 1 + \operatorname{erf} \left( \frac{x}{\sqrt{2}} \right) \right]$$



$$\text{Selu}: f(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha e^x - \alpha & \text{if } x \leq 0 \end{cases}$$

$$\text{Silu}: f(x) = \frac{x}{1 + e^{-x}}$$

$$\text{Leaky-Relu}: f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}$$

$$\text{Mish}: f(x) = x \tanh(\ln(1 + e^x))$$

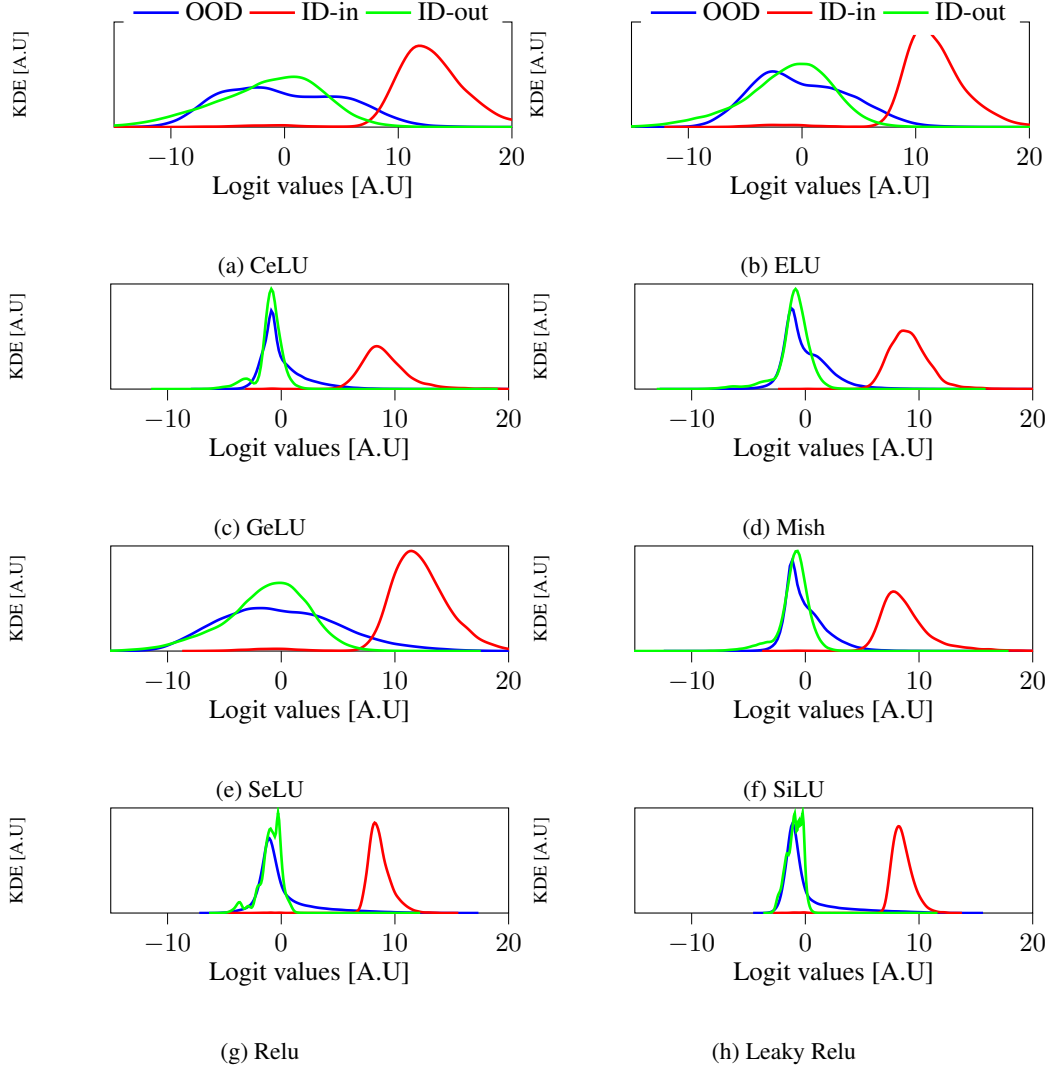


Figure 14: An analysis of the density over logits across eight distinct activation functions that suppress negative values is presented. The ResNet-34 architecture is utilized and trained on the SVHN dataset as the ID data, while the OOD includes CIFAR-10. For a detailed visualization of density plots on individual logit cells, see figs. 15 to 22

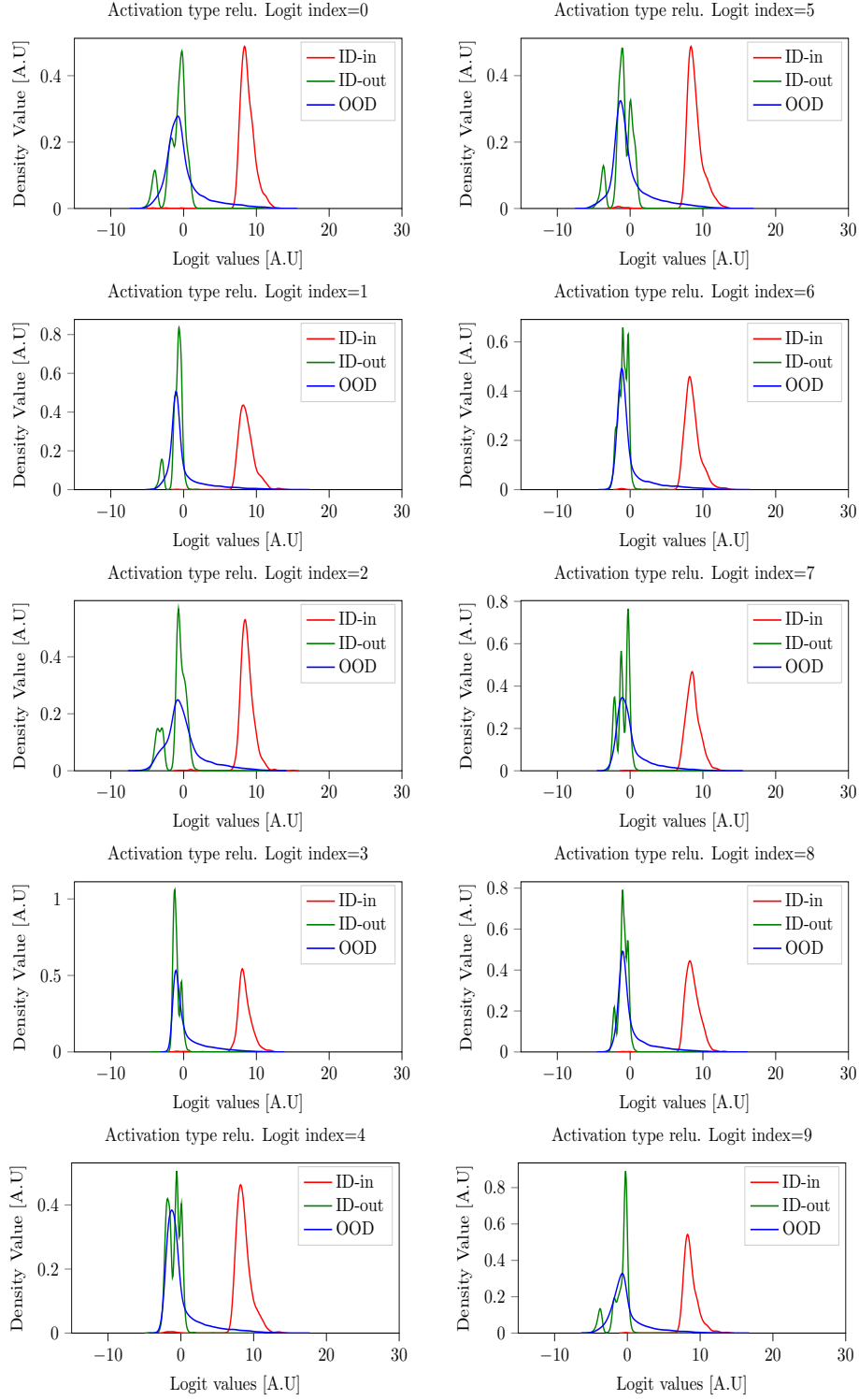


Figure 15: Densities over each logit cell from a Resnet-34 classifier with Relu activation.

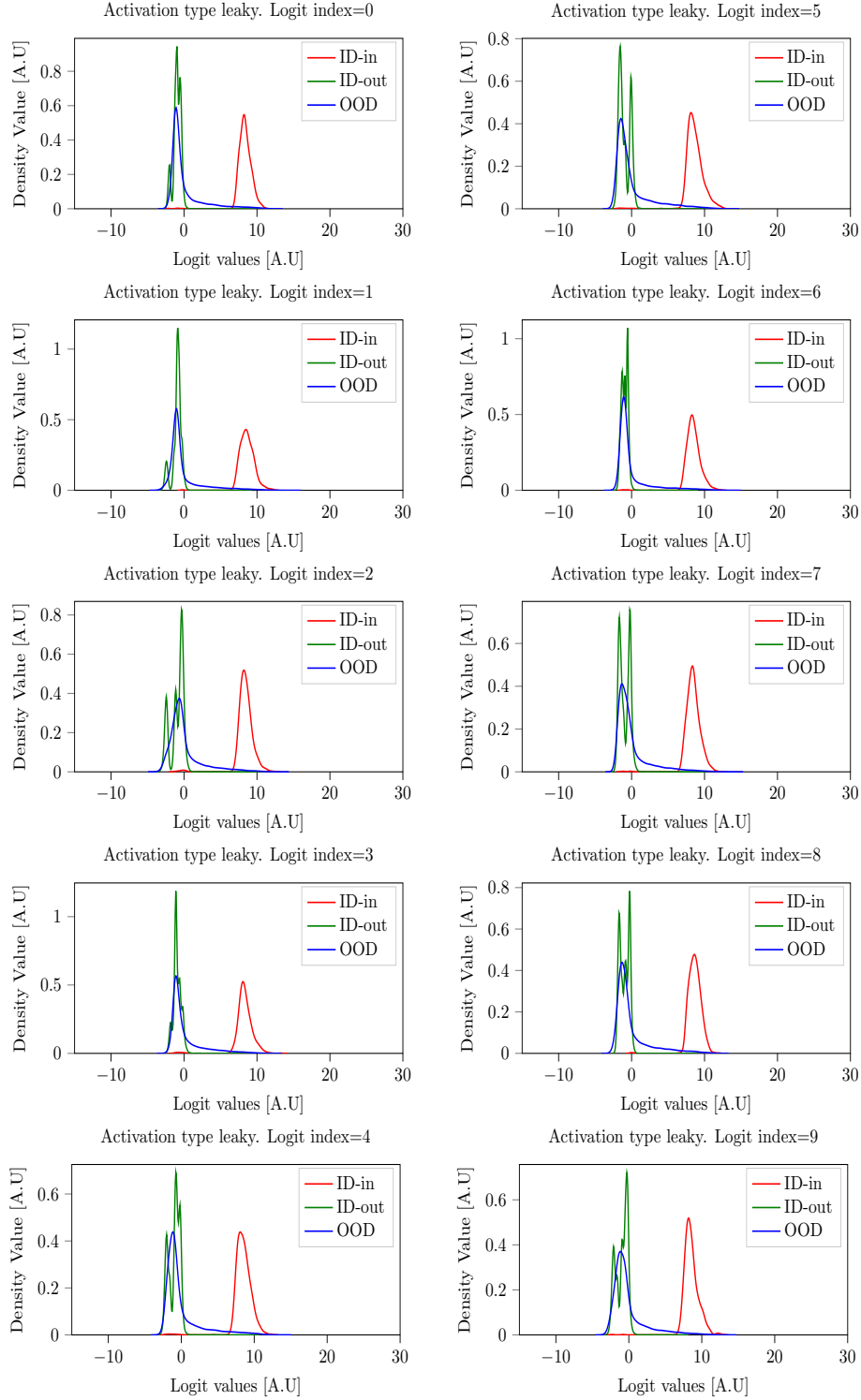


Figure 16: Densities over each logit cell from a Resnet-34 classifier with Leaky Relu activation.

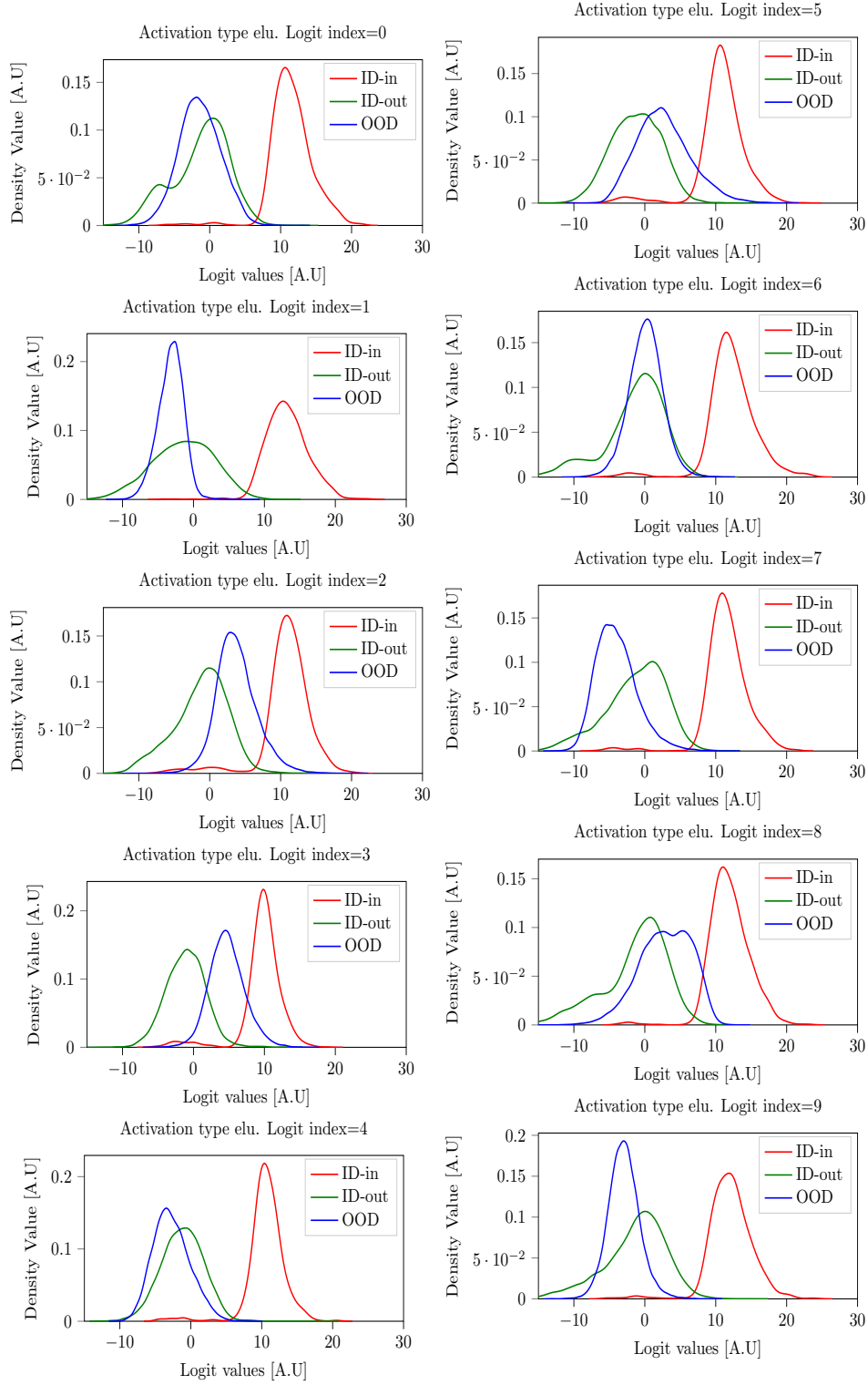


Figure 17: Densities over each logit cell from a Resnet-34 classifier with Elu activation.

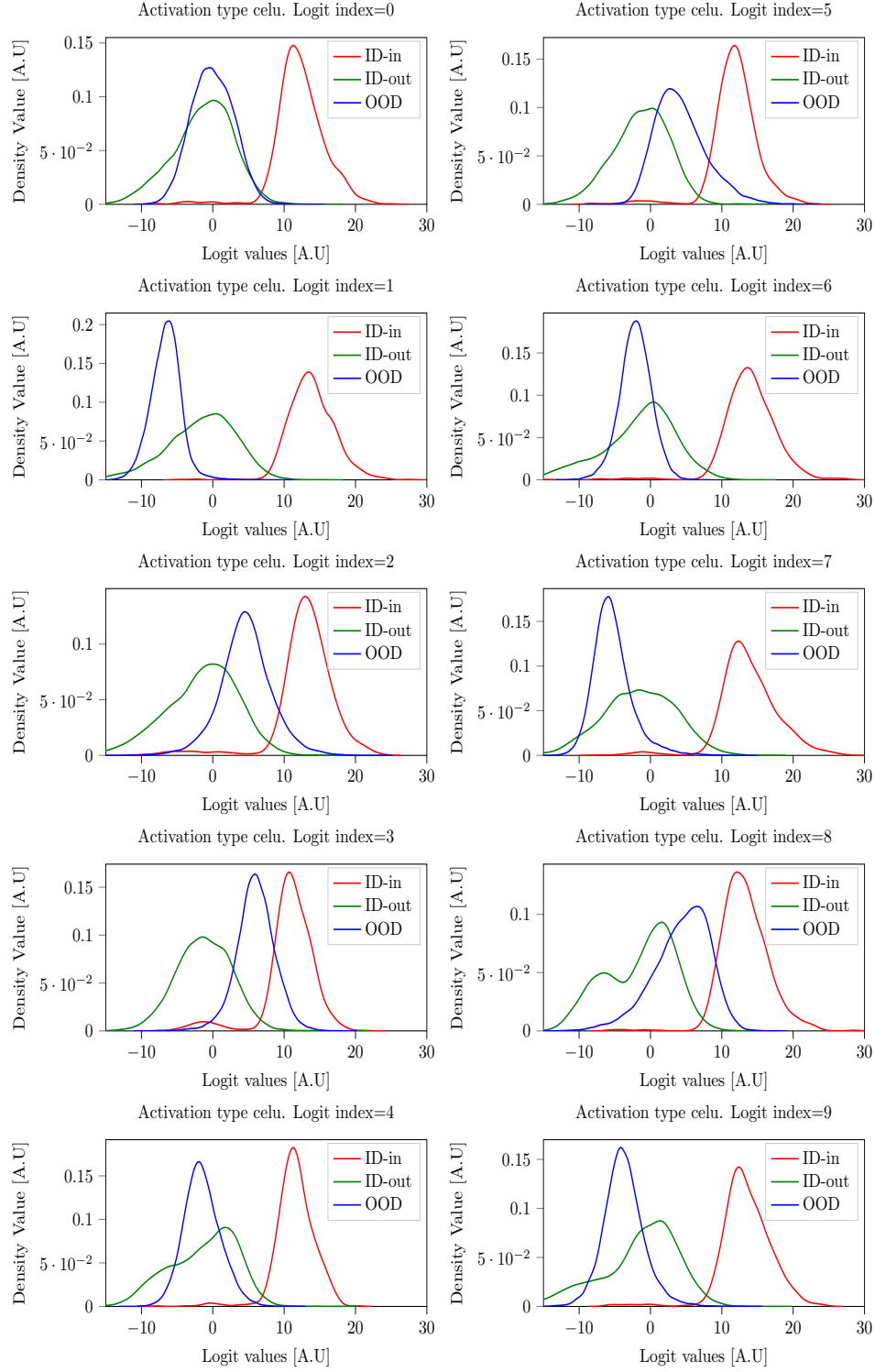


Figure 18: Densities over each logit cell from a Resnet-34 classifier with Celu activation.

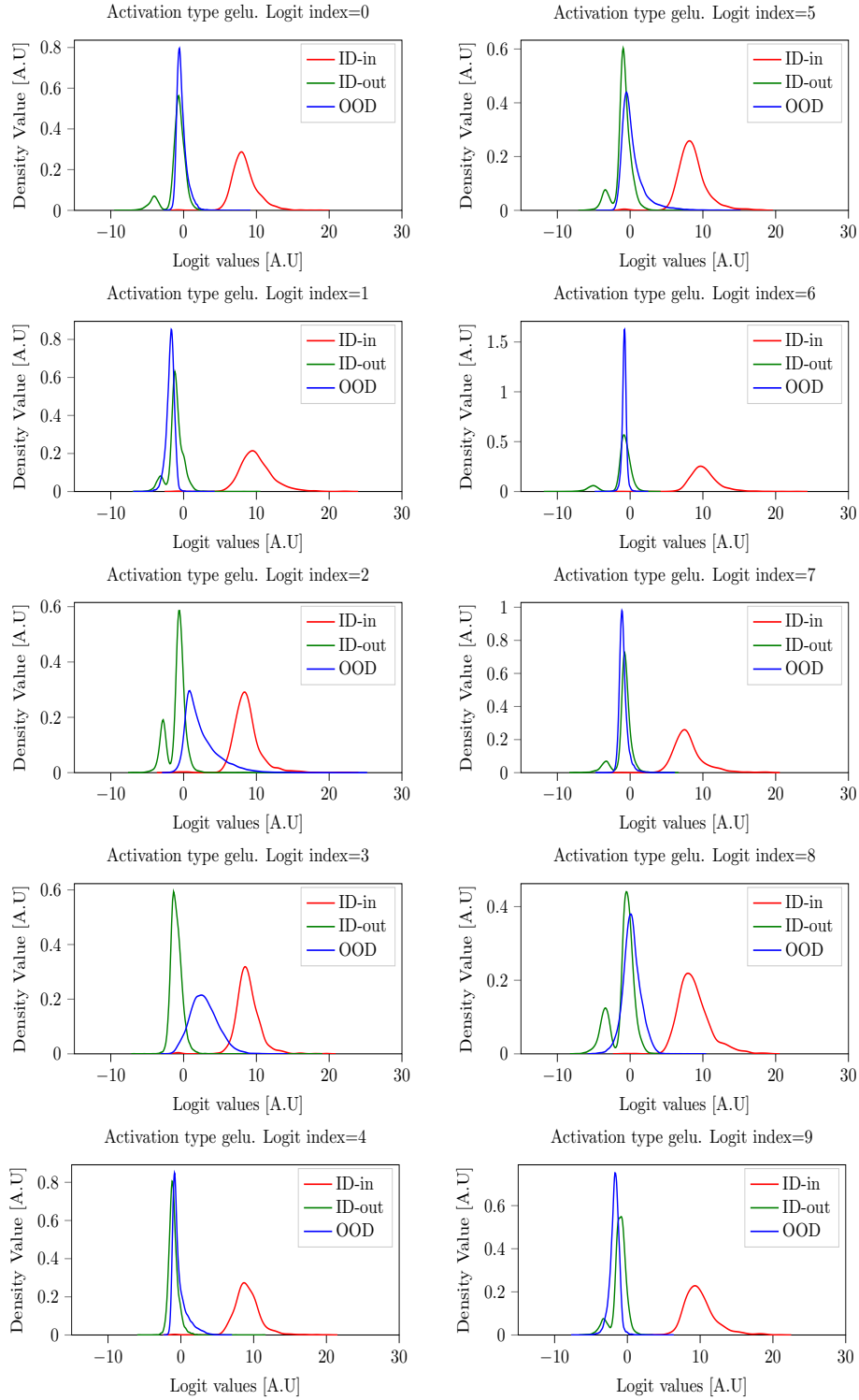


Figure 19: Densities over each logit cell from a Resnet-34 classifier with Gelu activation.

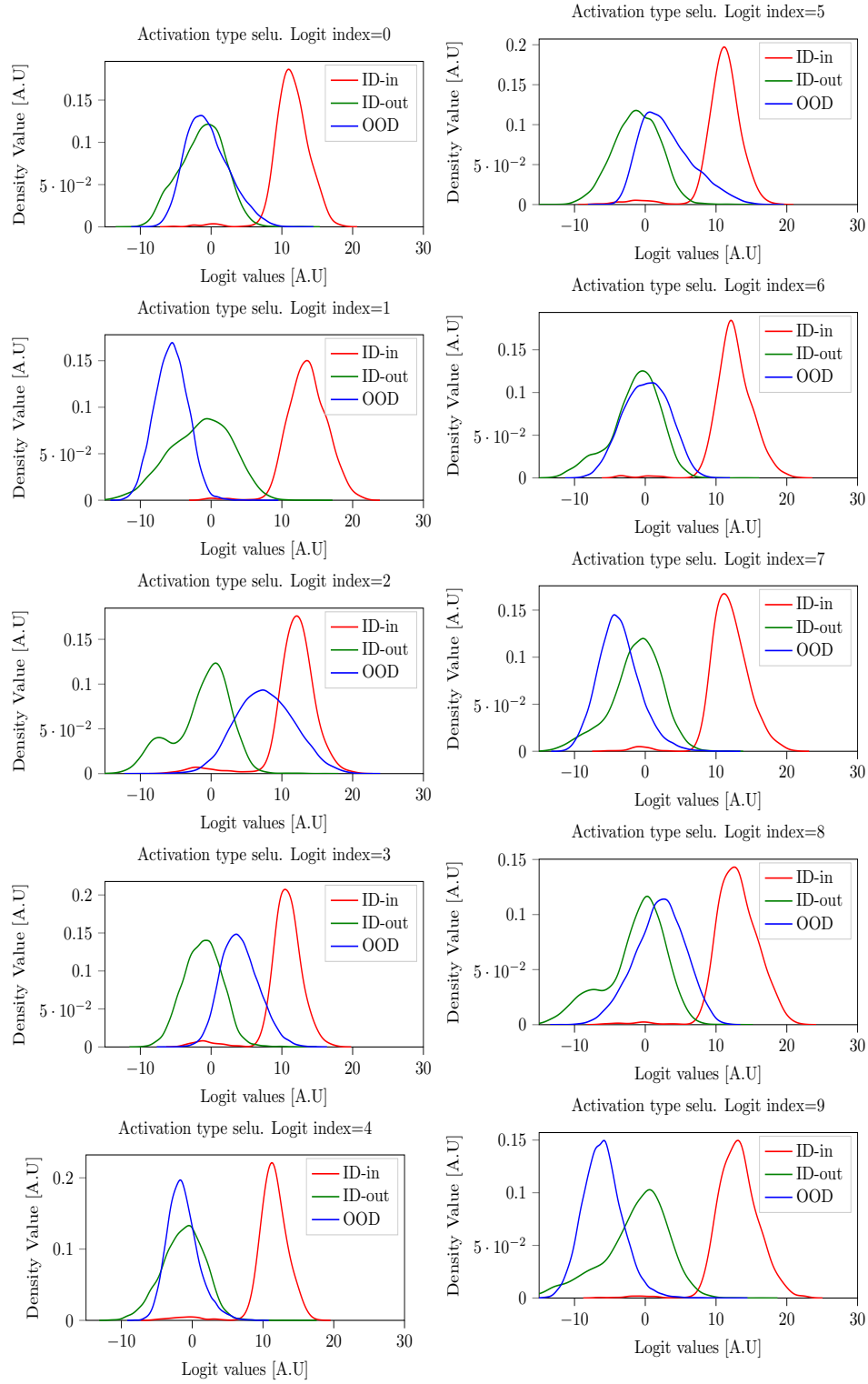


Figure 20: Densities over each logit cell from a Resnet-34 classifier with Selu activation.

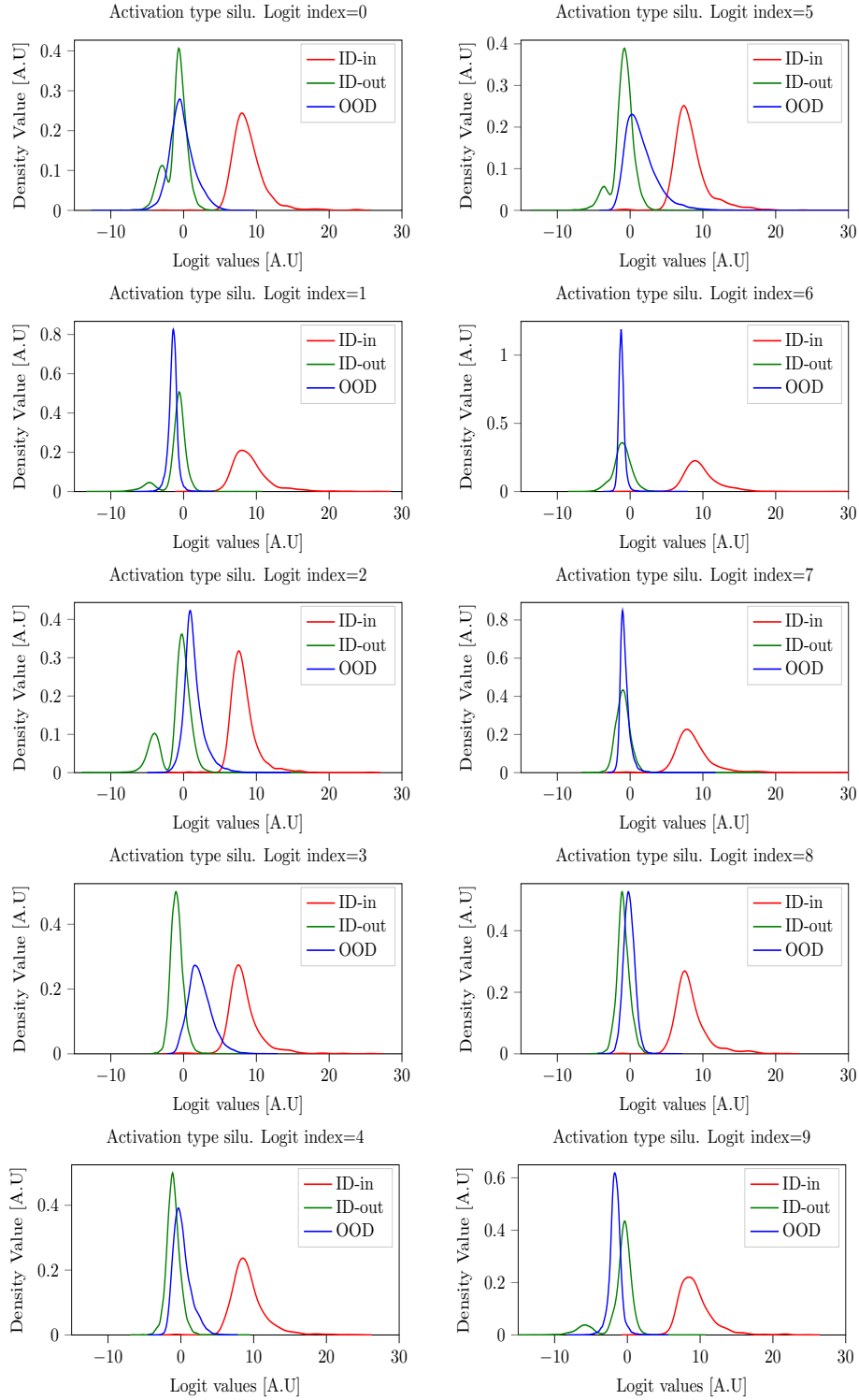


Figure 21: Densities over each logit cell from a Resnet-34 classifier with Silu activation.



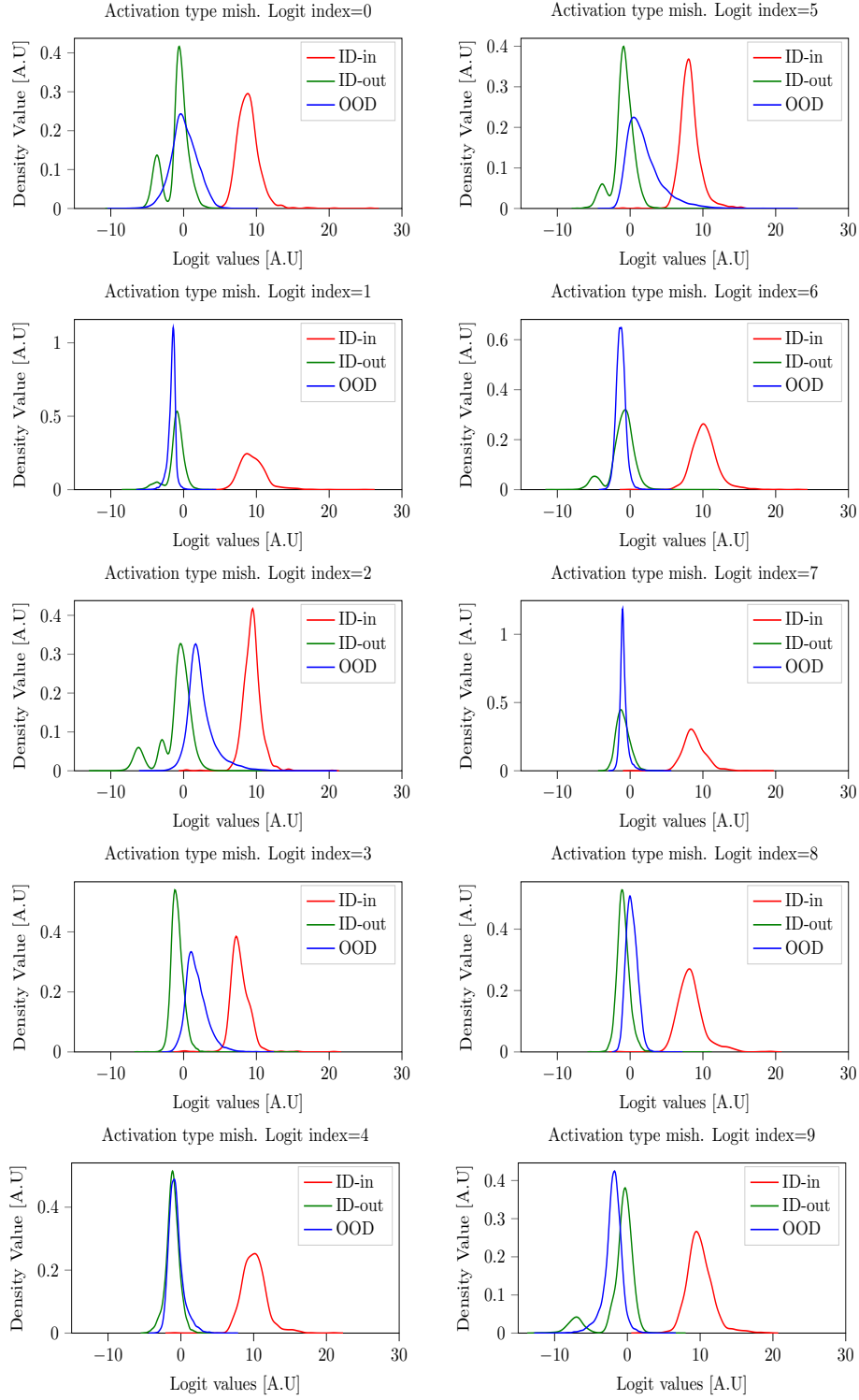


Figure 22: Densities over each logit cell from a Resnet-34 classifier with Mish activation.

## G EXPERIMENTS ON DIFFERENT DROPOUT RATE

Dropout Srivastava et al. (2014) is a widely-used regularization technique that improves generalization by randomly deactivating neurons during training. While its benefits for ID accuracy are well-established, its effects on OOD detection remain understudied. We investigate how dropout influences the separability between ID and OOD data, revealing a critical dichotomy: OOD discriminability differs significantly between training (dropout active) and inference (dropout inactive) phases.

### EXPERIMENTAL SETUP

We evaluate a ResNet-34 model trained on CIFAR-10 (ID data) with dropout rates  $p \in \{0.2, 0.4, 0.6, 0.8\}$  applied after each convolutional layer. For OOD evaluation, we use six standard benchmarks:

$$\mathcal{D}_{OOD} = \{\text{SVHN, CIFAR-100, Tiny ImageNet, iSUN, LSUN}\} \quad (10)$$

### KEY OBSERVATIONS

Our analysis yields three principal findings:

- **Logit Distribution Overlap:** Higher dropout rates increase the overlap between ID and OOD logit distributions, as shown in fig. 23.
- **Mode Dependency:** The overlap is more pronounced during inference (dropout inactive) than training (dropout active), suggesting that dropout’s stochasticity partially preserves OOD discriminability during training.
- **Generalization Trade-off:** Excessive dropout weakens ID-discriminative features, degrading OOD detection performance. This indicates an unintended correlation between dropout-induced weight sparsity and OOD feature sensitivity.

In addition to the density plots shown in fig. 23, which illustrate the aggregated distribution of ID-in, ID-out, and OOD across all logit cells, a more detailed visualization of density plots on individual logit cells can be found in figs. 24 to 31.

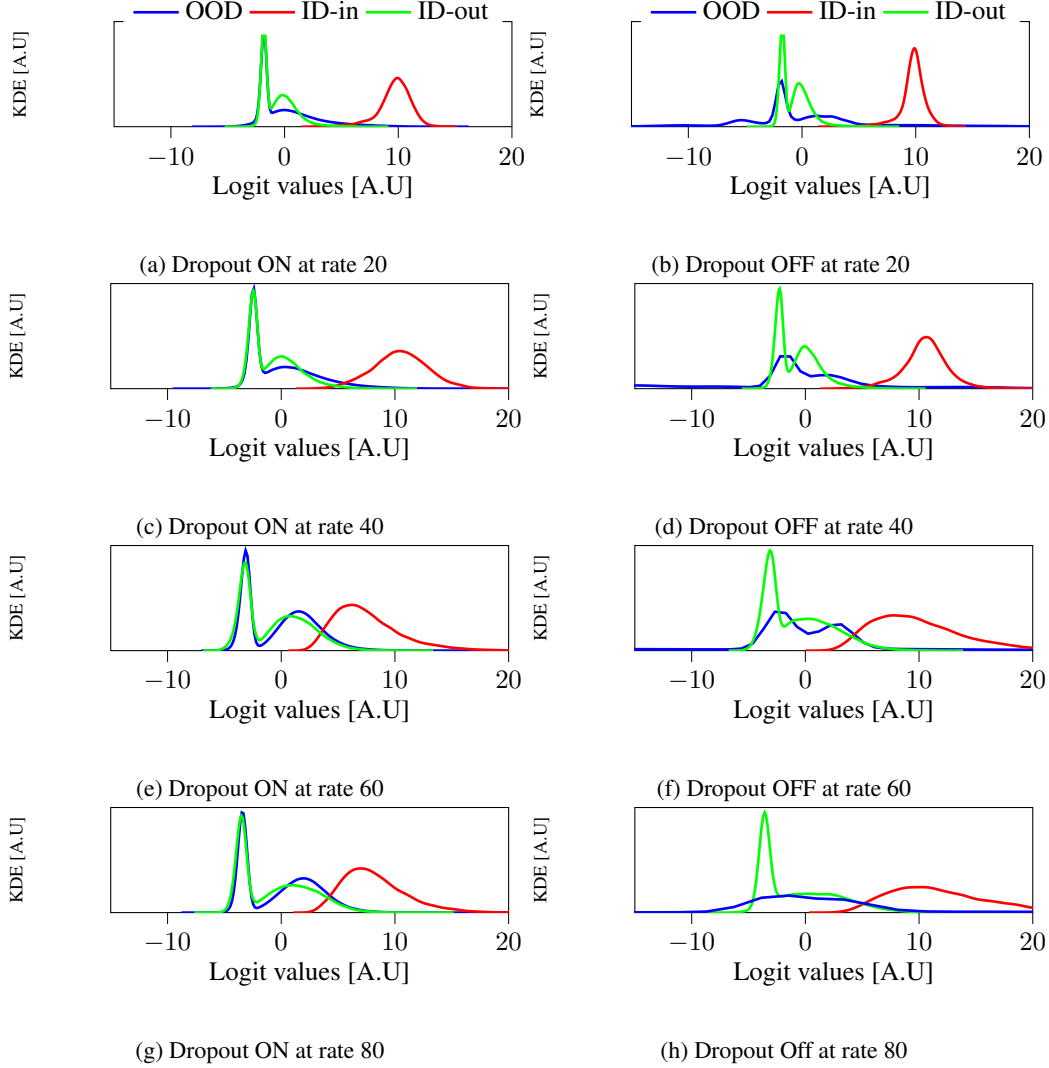


Figure 23: An analysis of the density over logits across eight distinct dropout rates is presented. The ResNet-34 architecture is utilized and trained on the CIFAR-10 dataset as the ID data, while the OOD includes  $\{D \setminus \text{CIFAR-10}\}$ .

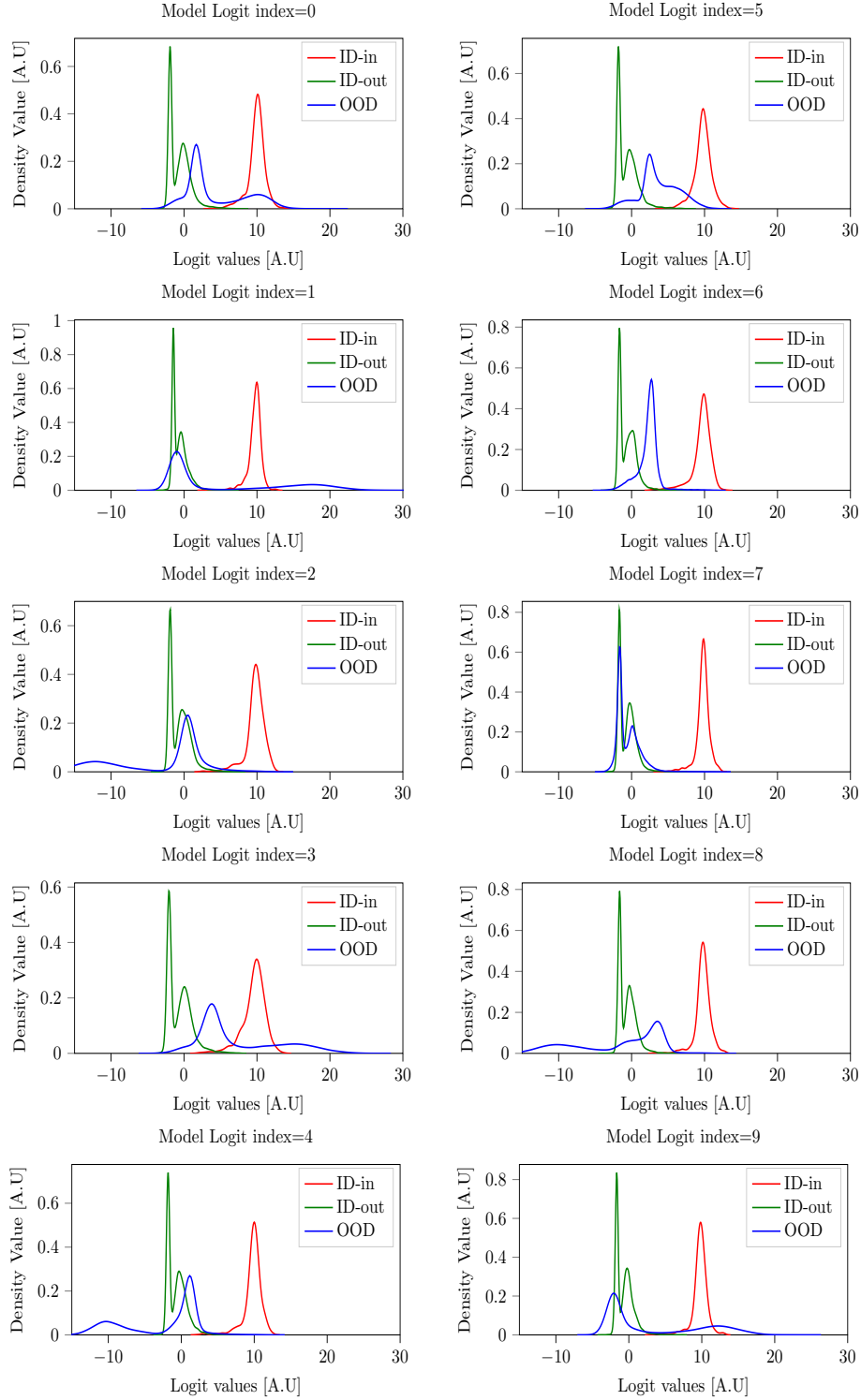


Figure 24: Densities over each logit cell from a Resnet-34 with a dropout of 20% which is deactivated post-train.

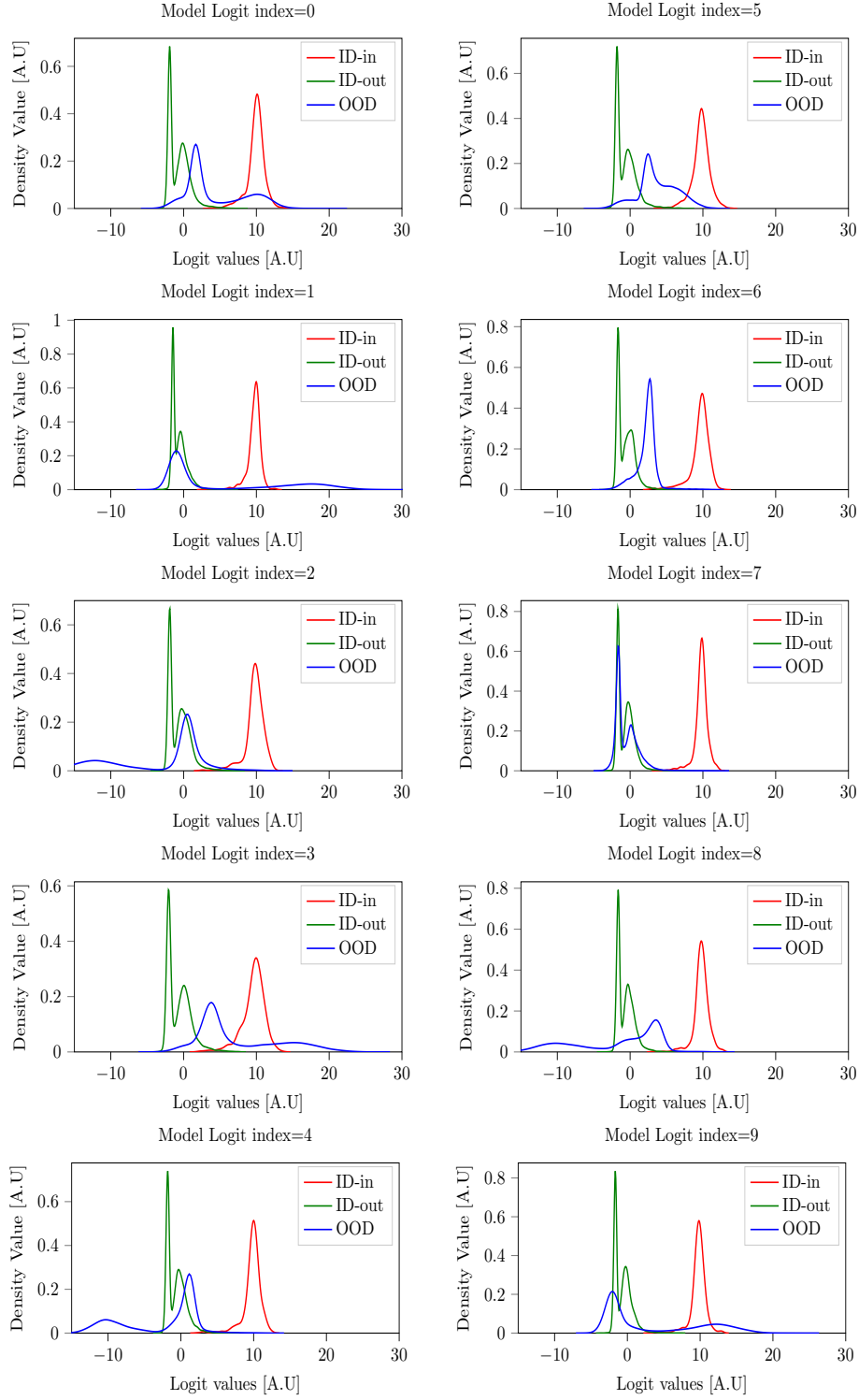


Figure 25: Densities over each logit cell from a Resnet-34 with a dropout of 40% which is deactivated post-train.

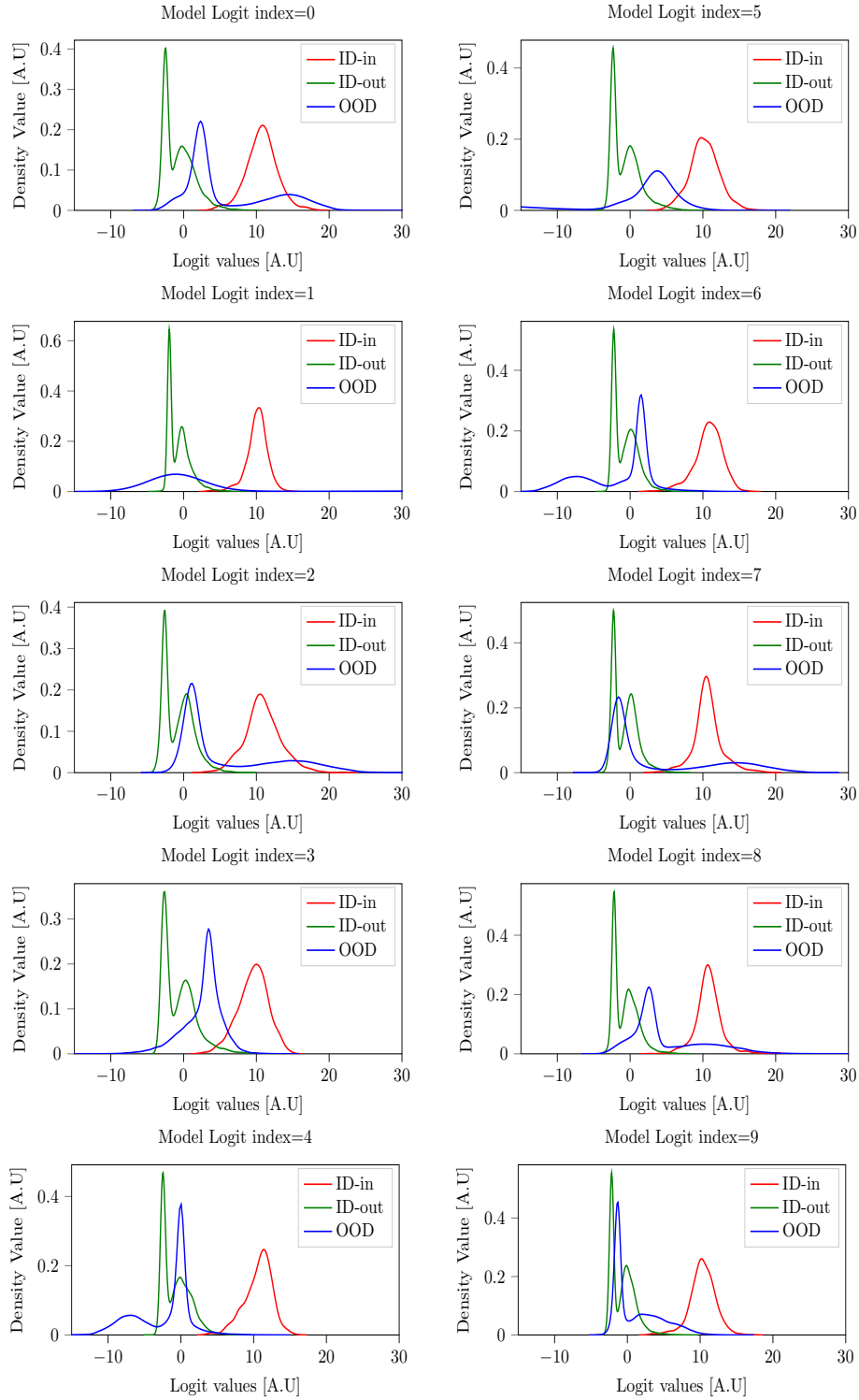


Figure 26: Densities over each logit cell from a Resnet-34 with a dropout of 60% which is deactivated post-train.

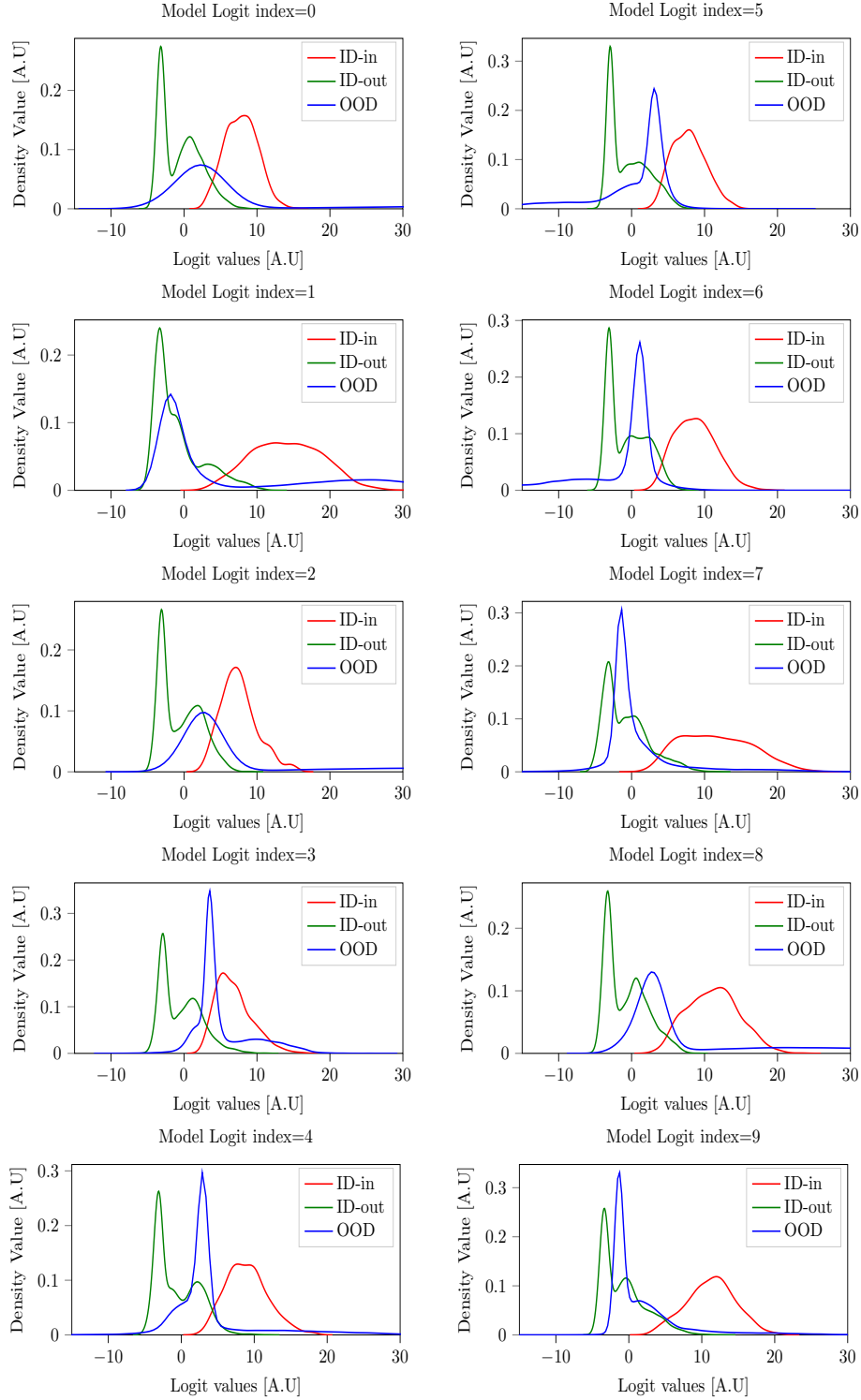


Figure 27: Densities over each logit cell from a Resnet-34 with a dropout of 80% which is deactivated post-train.

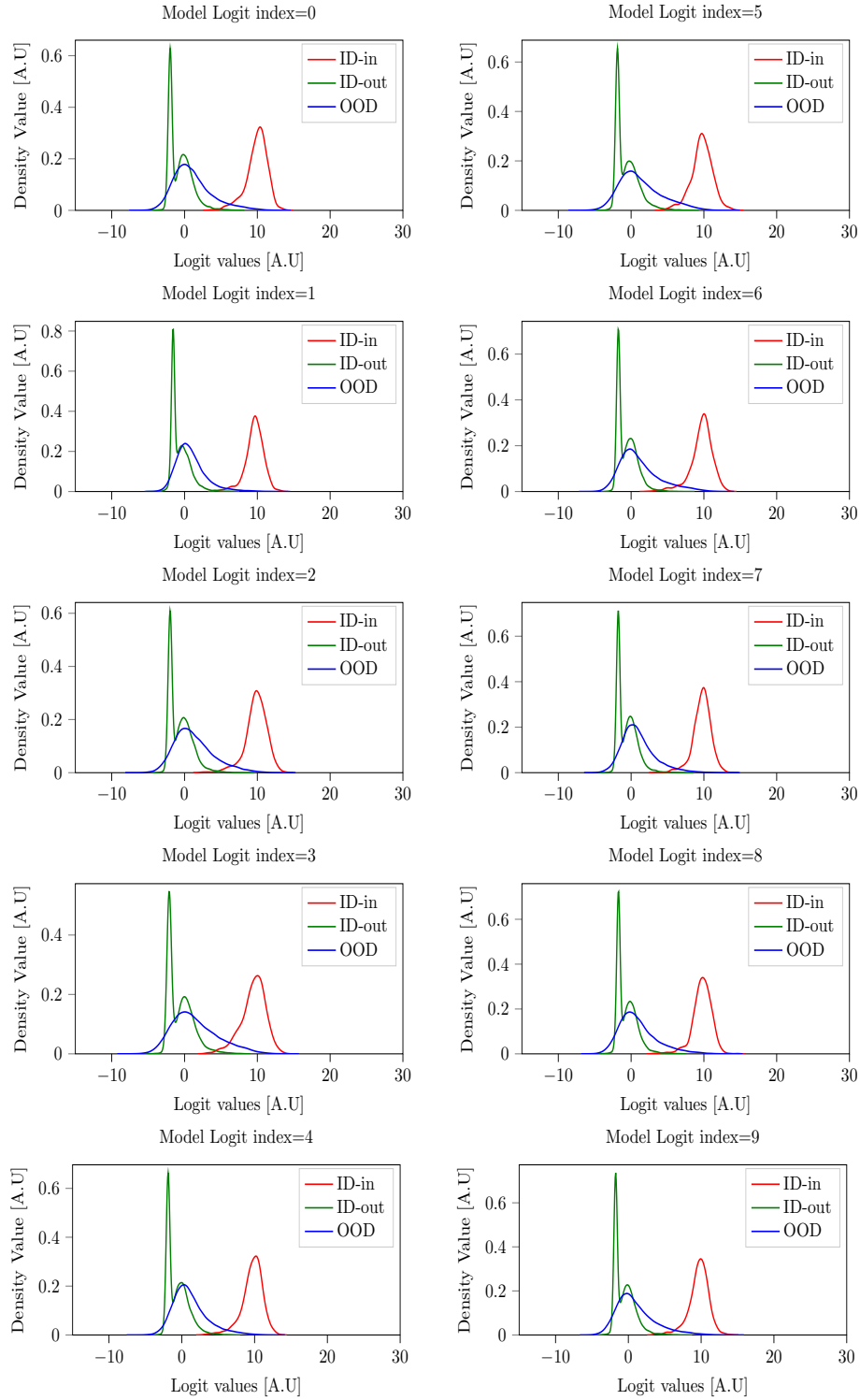


Figure 28: Densities over each logit cell from a Resnet-34 with a dropout of 20%, which remains activated post-train.



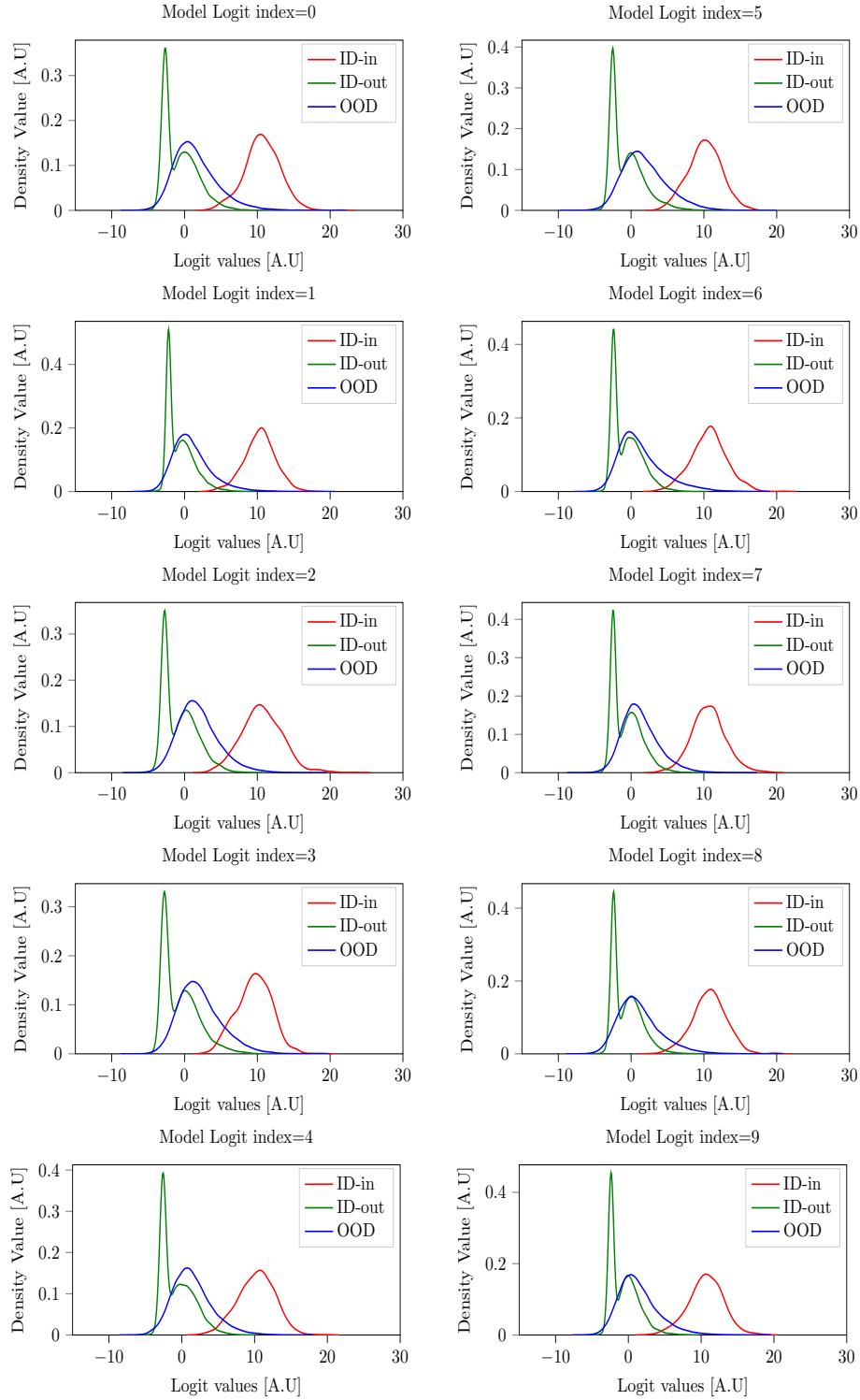


Figure 29: Densities over each logit cell from a Resnet-34 with a dropout of 40%, which remains activated post-train.

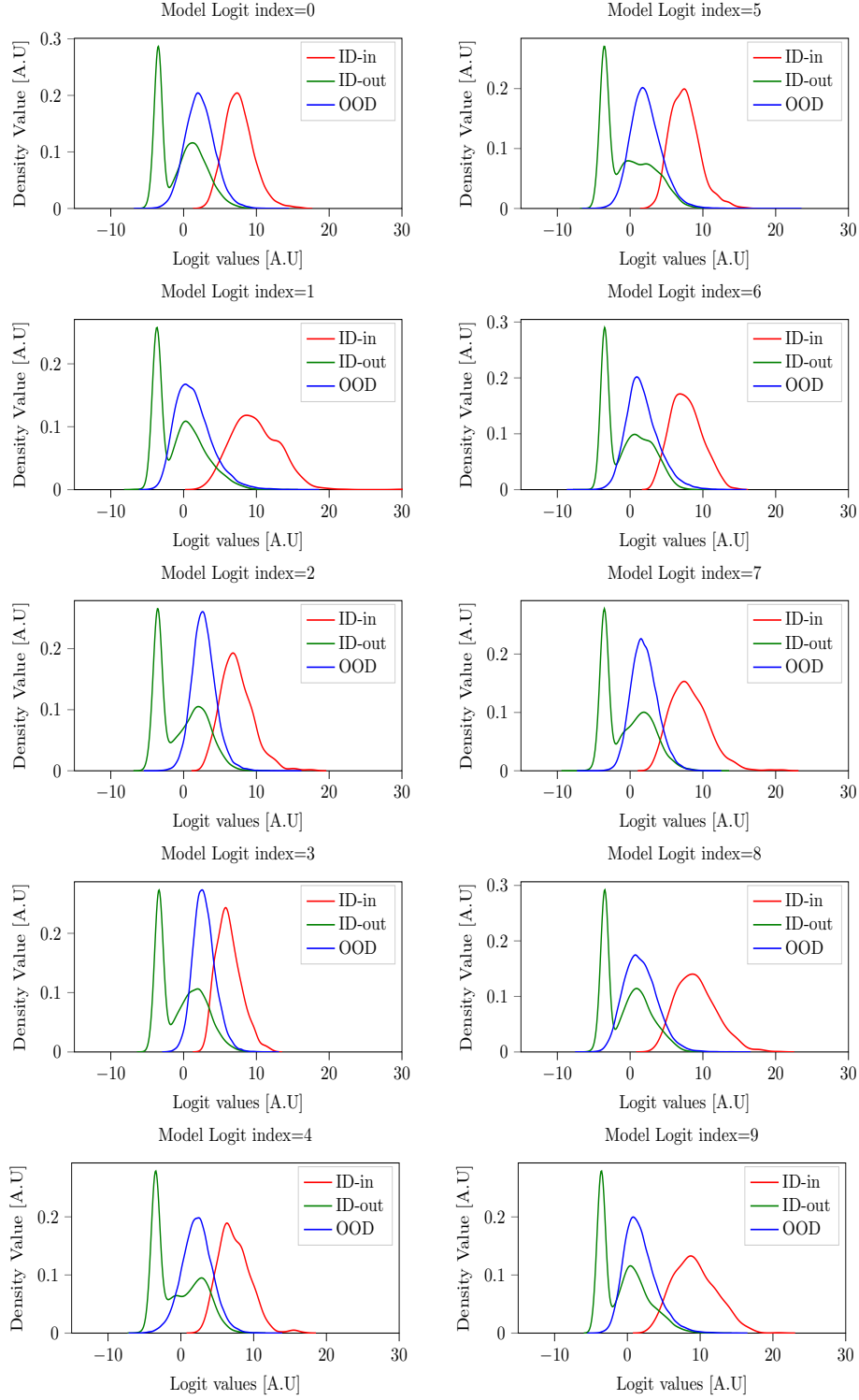


Figure 30: Densities over each logit cell from a Resnet-34 with a dropout of 60%, which remains activated post-train.

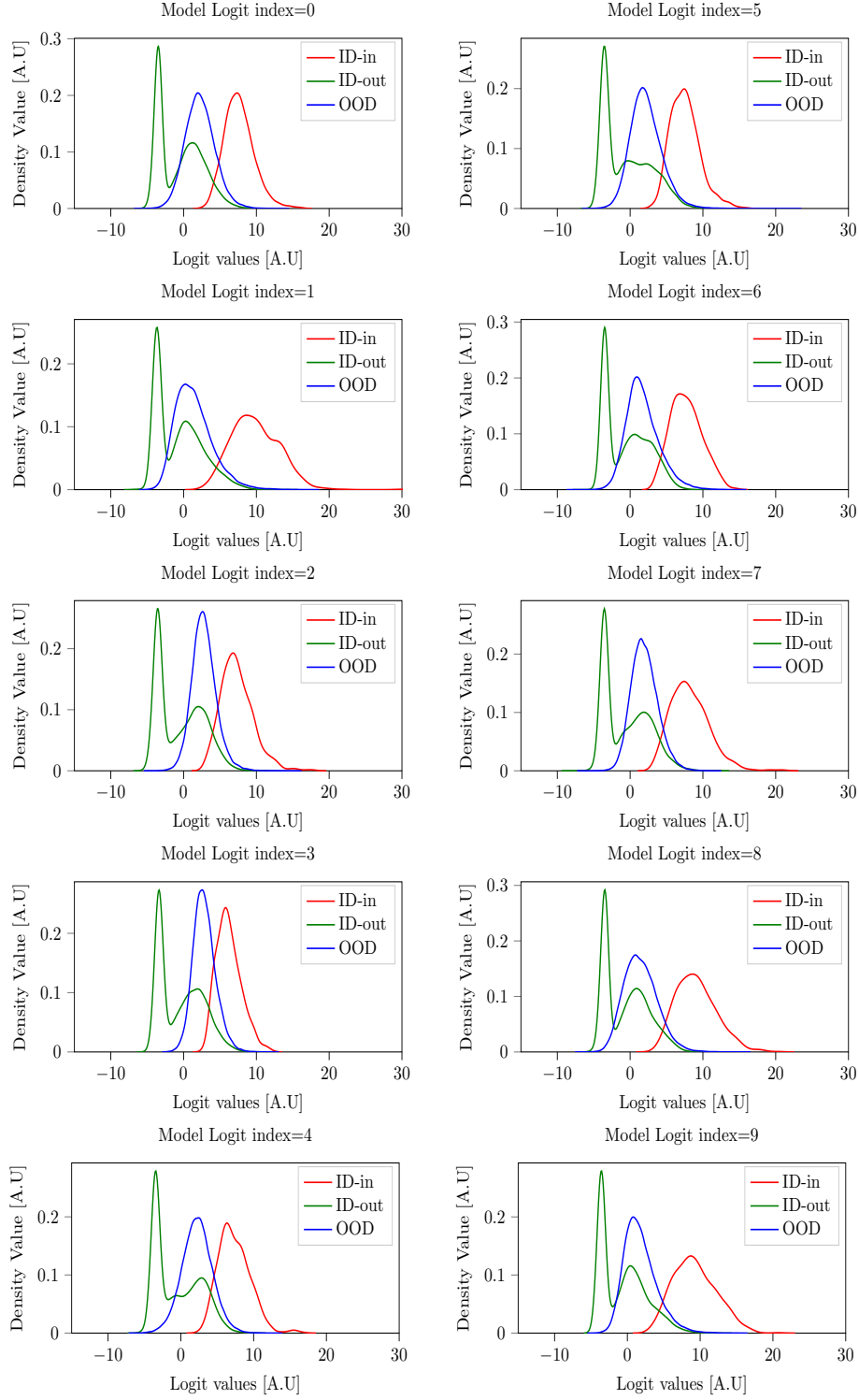


Figure 31: Densities over each logit cell from a Resnet-34 with a dropout of 80%, which remains activated post-train.

## H EXPERIMENTS ON DIFFERENT CLASSIFIERS

Table 3: Distribution mode for the ID and OOD logits over different versions of ResNet.

Data	Setup	ResNet18	ResNet34	ResNet50	ResNet101	ResNet152
SVHN	ID-in	11.5	11.3	9.4	10.0	13.5
	ID-out	1.4	0.7	-2.2	-0.4	1.7
	OOD ( $\mathcal{D} \setminus \text{SVHN}$ )	0.2	0.2	-1.5	-1.8	-0.5
CIFAR-10	ID-in	11.6	11.3	9.5	10.0	13.5
	ID-out	-4.5	-2.6	1.5	2.1	-3.2
	OOD ( $\mathcal{D} \setminus \text{CIFAR-10}$ )	-3.0	-2.3	0.6	1.3	-2.6

Table 4: Distribution mode for the ID and OOD logits over different versions of DenseNet.

Data	Setup	DenseNet201	DenseNet161	DenseNet121	DenseNet169
SVHN	ID-in	5.7	5.5	5.7	6.1
	ID-out	-1.0	-1.2	-1.7	-1.5
	OOD ( $\mathcal{D} \setminus \text{SVHN}$ )	-1.8	-1.7	-1.8	-2.0
CIFAR-10	ID-in	12.5	12.7	12.7	13.3
	ID-out	-1.9	-1.7	-1.8	-2.2
	OOD ( $\mathcal{D} \setminus \text{CIFAR-10}$ )	-1.1	-1.0	-1.6	-1.6

Apart from using CIFAR-10 as ID training data, we rerun the same experiments on DenseNet and ResNet using SVHN as ID and  $\{\mathcal{D}\} \setminus \text{SVHN}$  as OOD (see figs. 32 and 33)

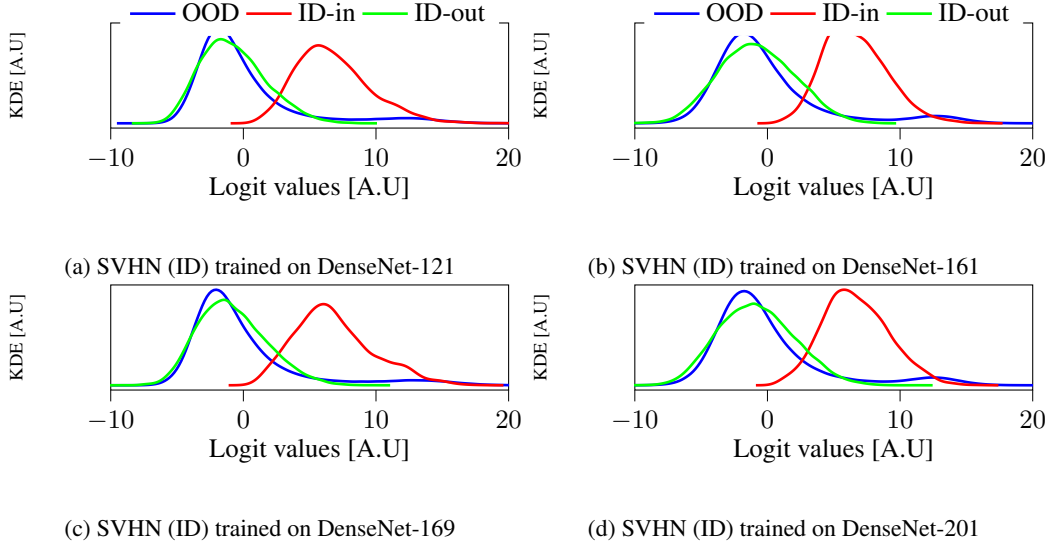


Figure 32: An analysis of the density over aggregated logits across distinct ResNet architectures trained on the SVHN dataset as the ID data, while the OOD includes  $\{\mathcal{D}\} \setminus \text{SVHN}$ . For a more detailed comparison, check figs. 47 to 51

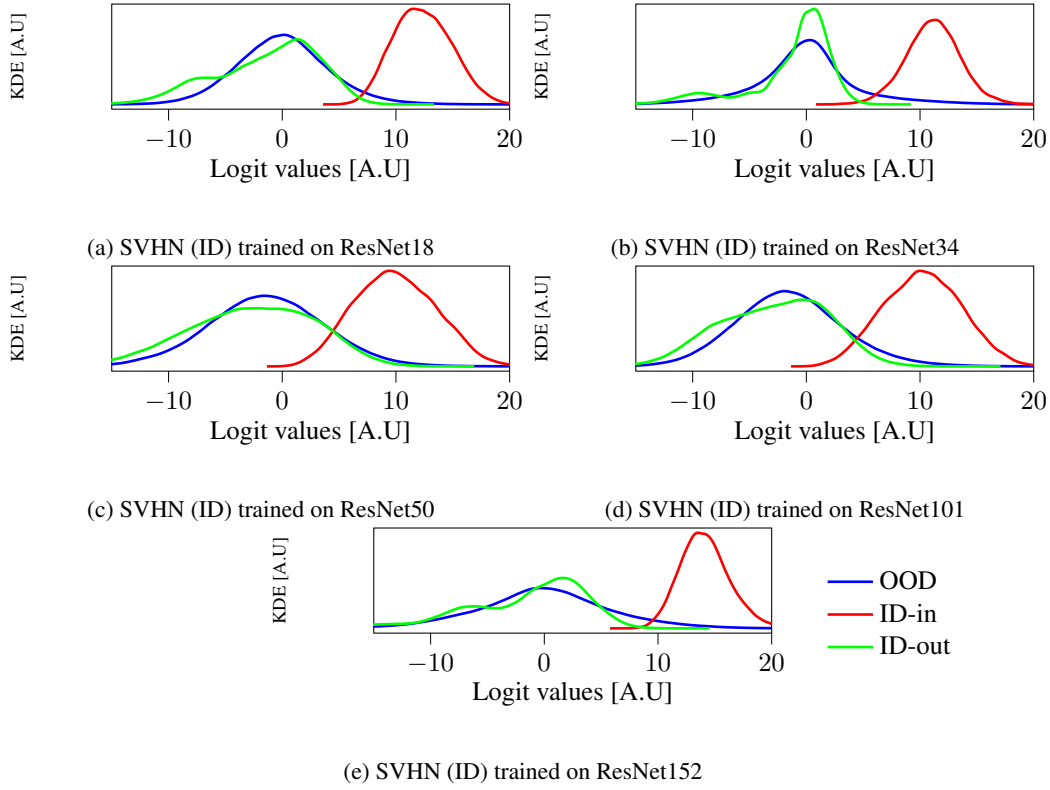


Figure 33: An analysis of the density over aggregated logits across distinct DenseNet architecture trained on the CIFAR-10 dataset as the ID data, while the OOD includes  $\{\mathcal{D}\} \setminus \text{CIFAR-10}$ . For a more detailed comparison, check figs. 34 to 41.

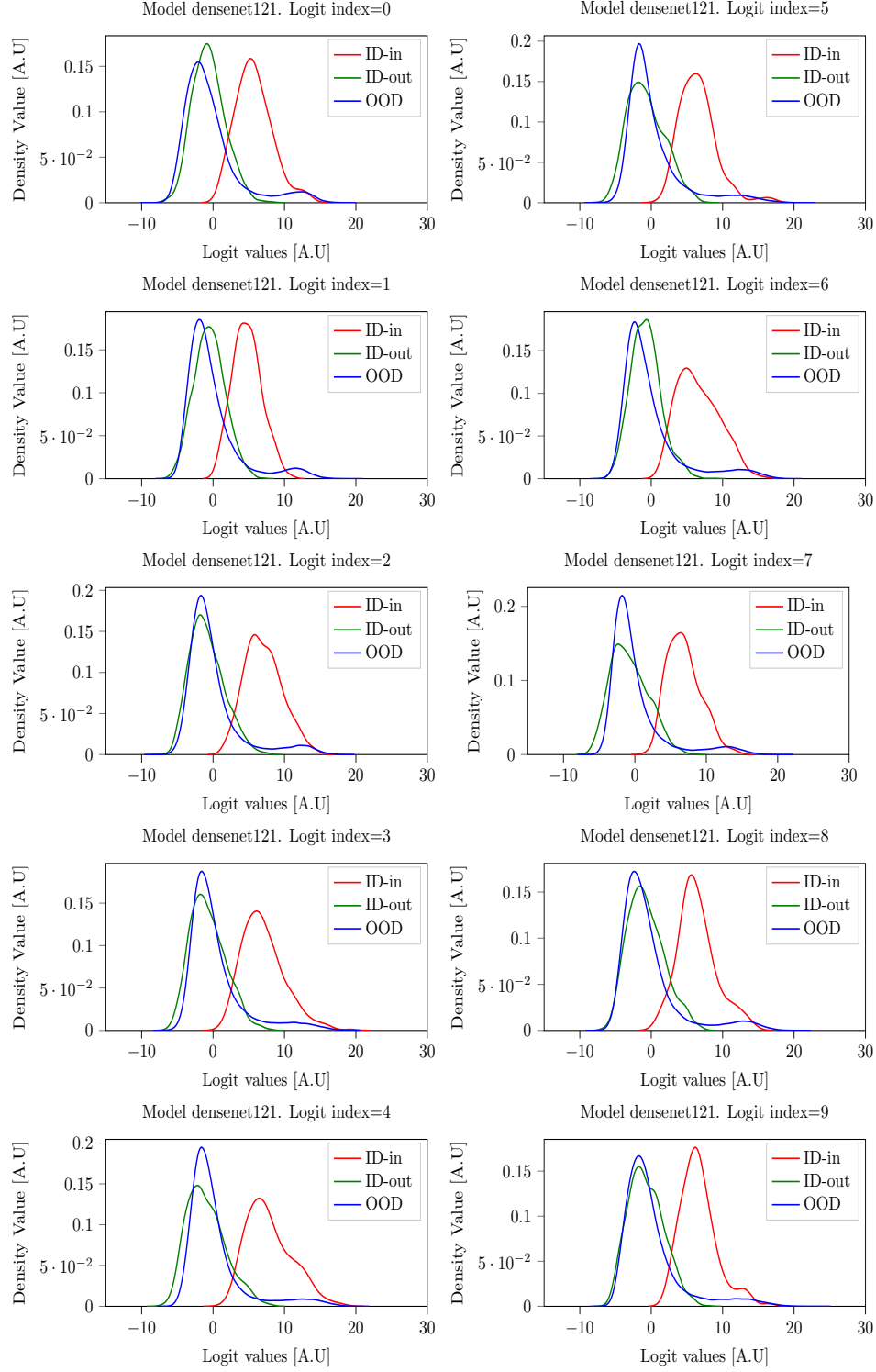


Figure 34: Logit cell densities for SVHN as ID with Densenet121.

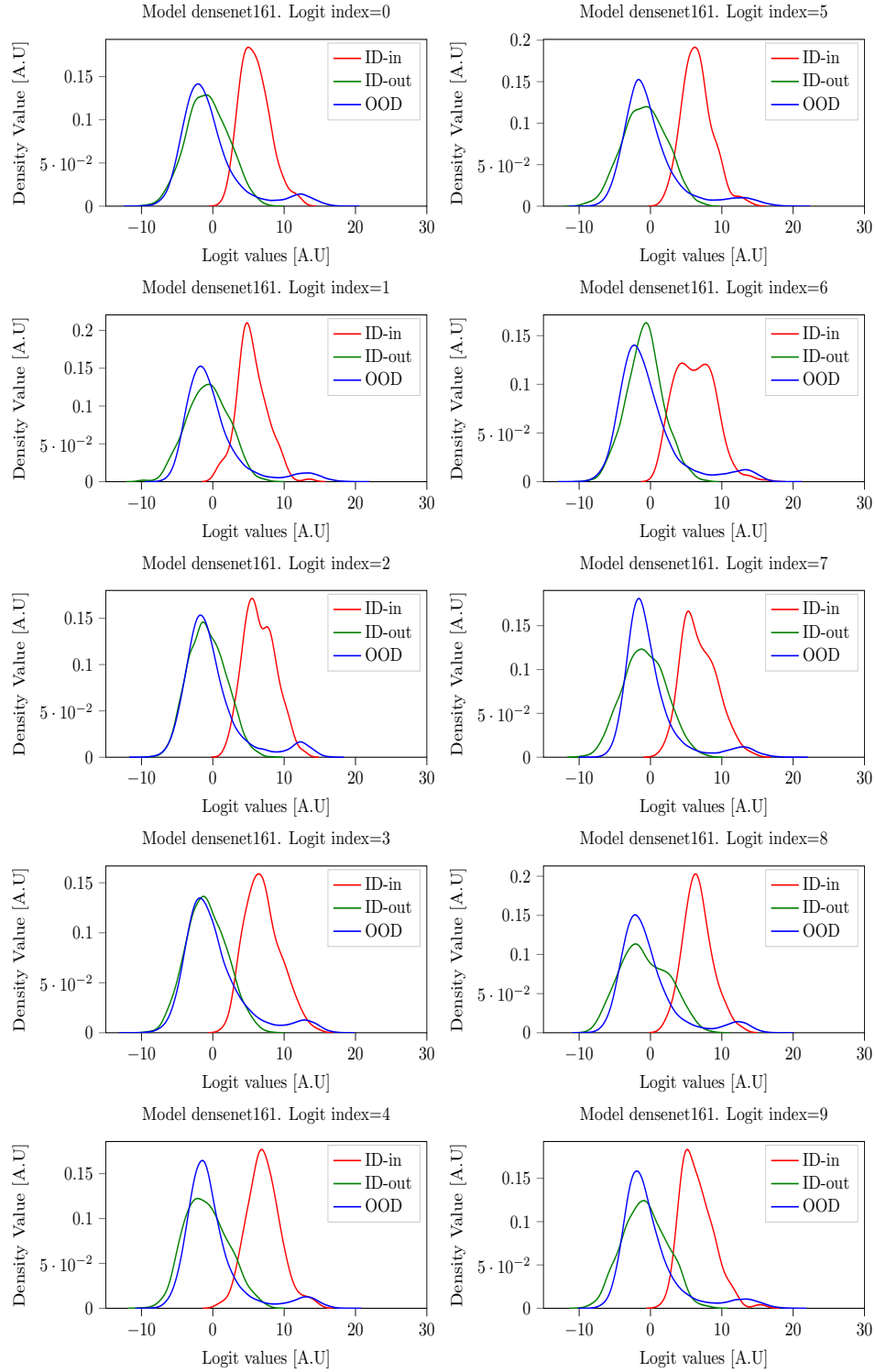


Figure 35: Logit cell densities for SVHN as ID with Densenet161.

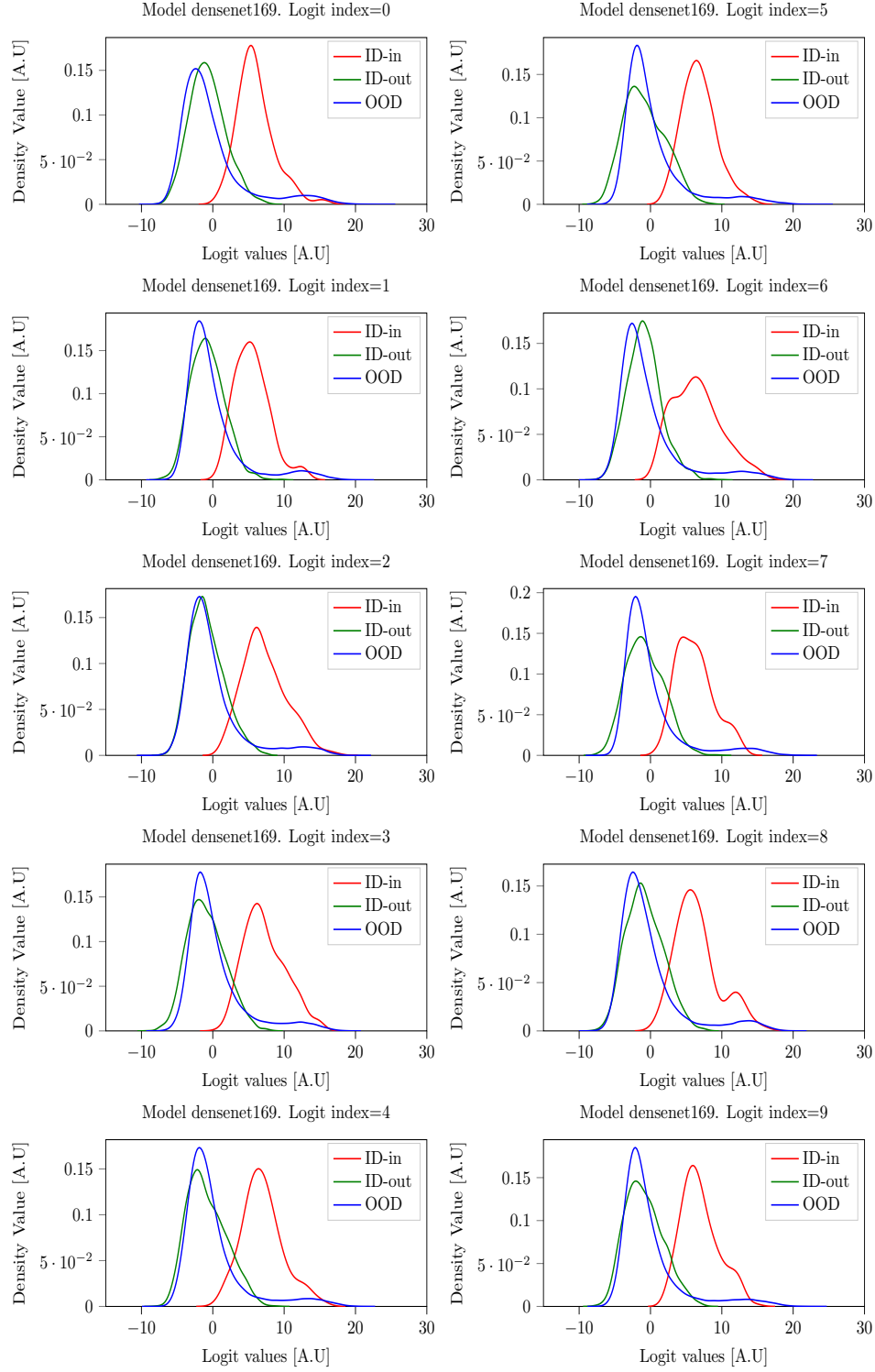


Figure 36: Logit cell densities for SVHN as ID with Densenet169.



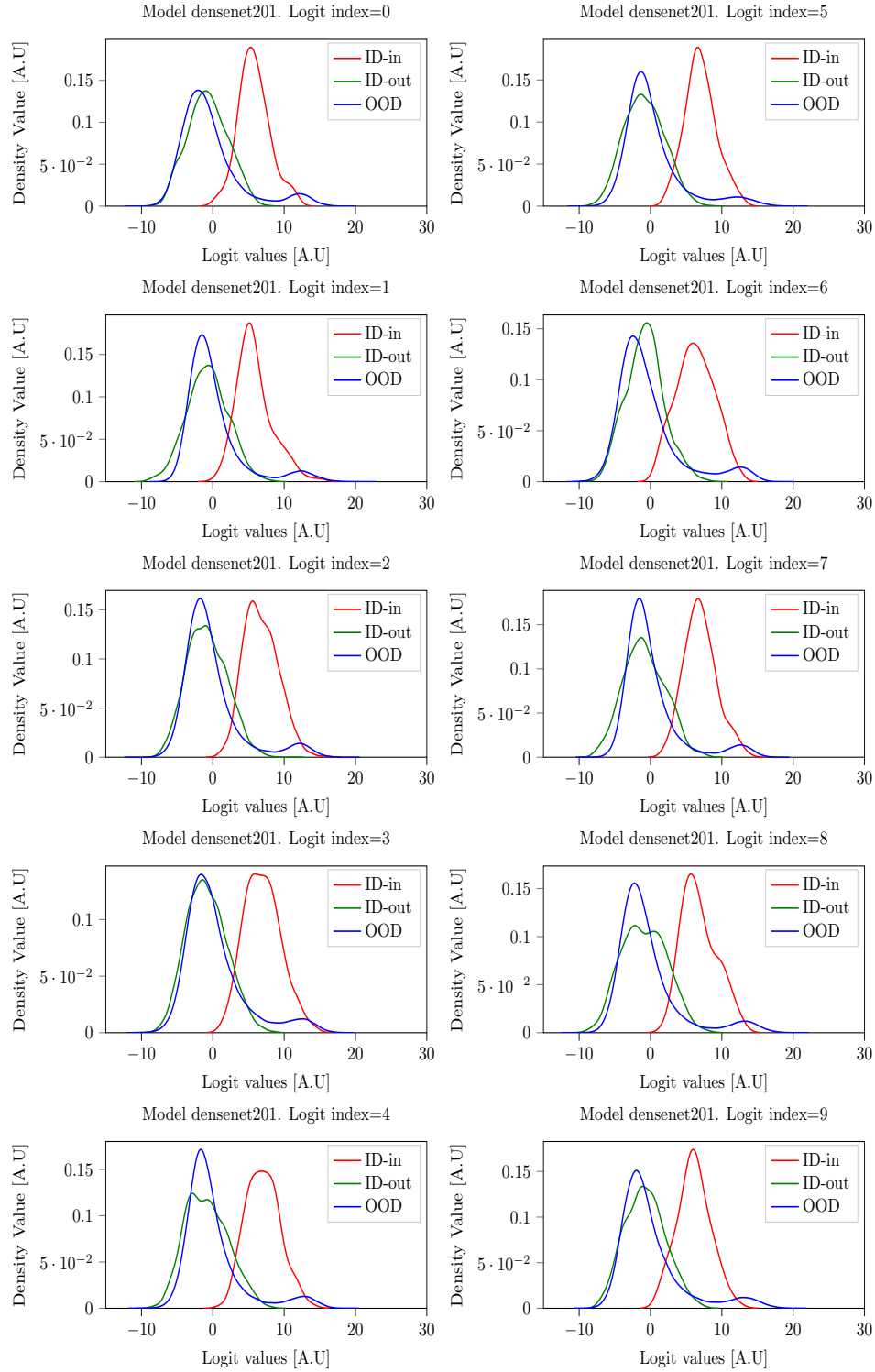


Figure 37: Logit cell densities for SVHN as ID with Densenet201.

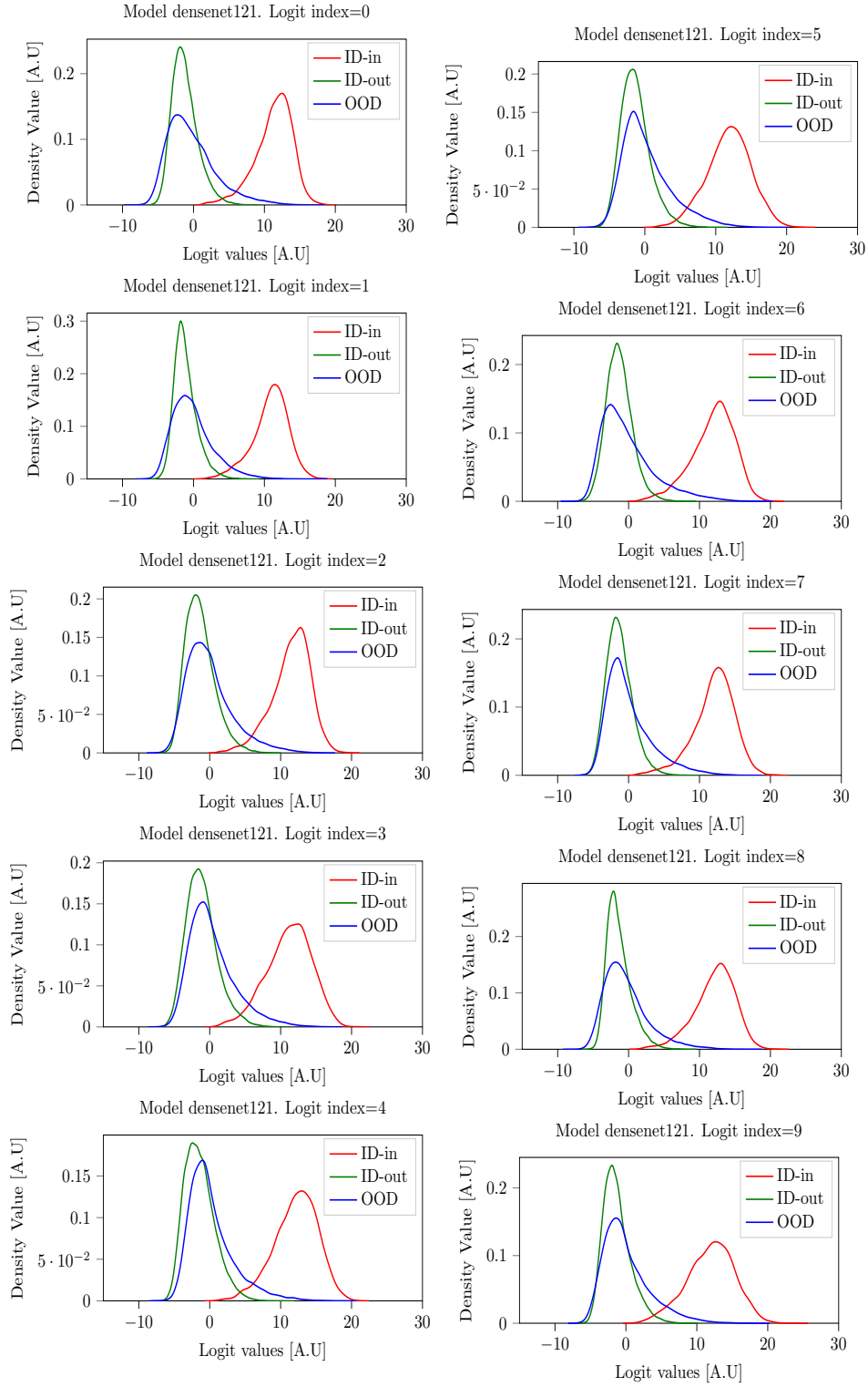


Figure 38: Logit cell densities for CIFAR-10 as ID with Densenet121.

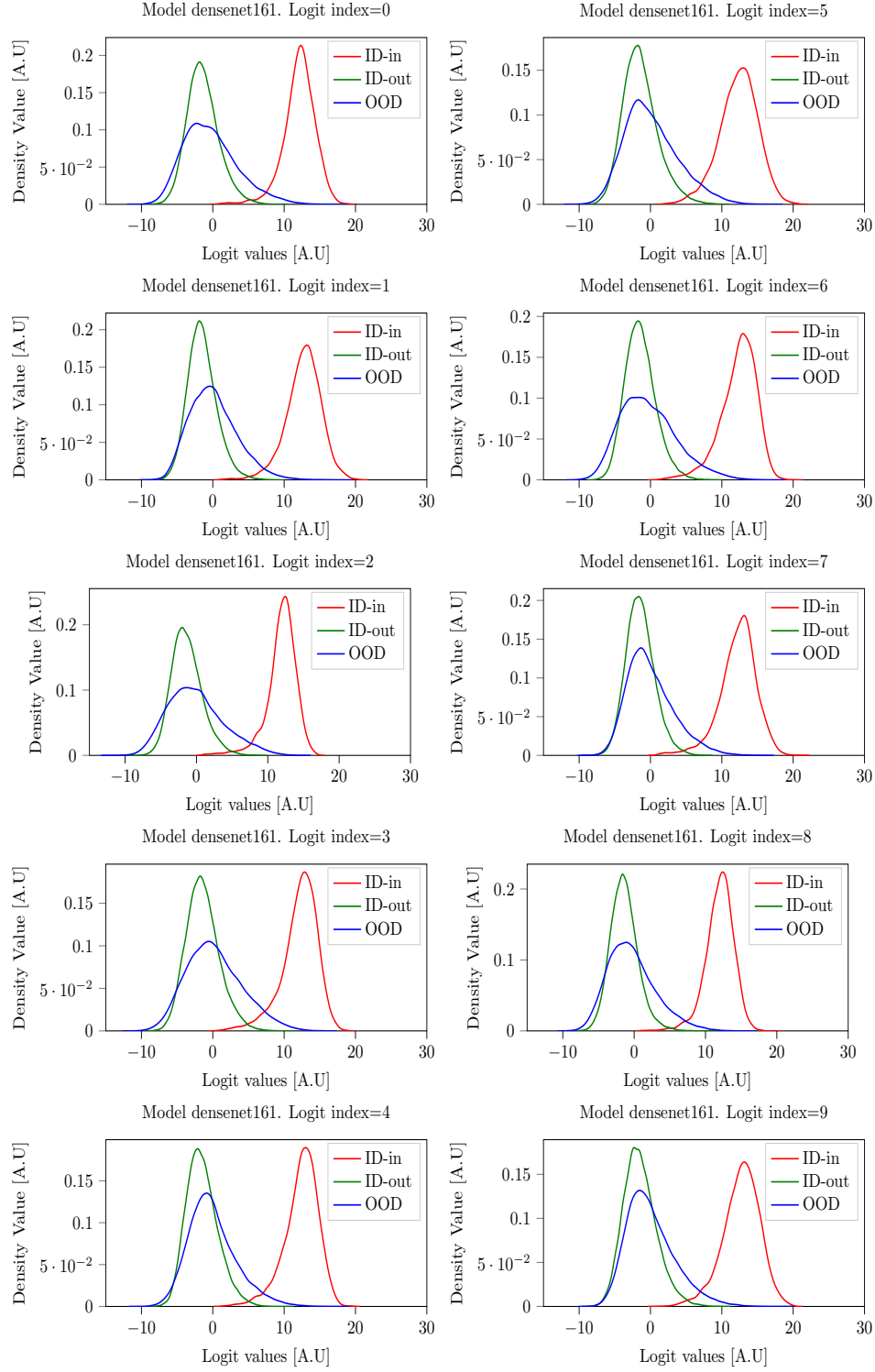


Figure 39: Logit cell densities for CIFAR-10 as ID with Densenet161.

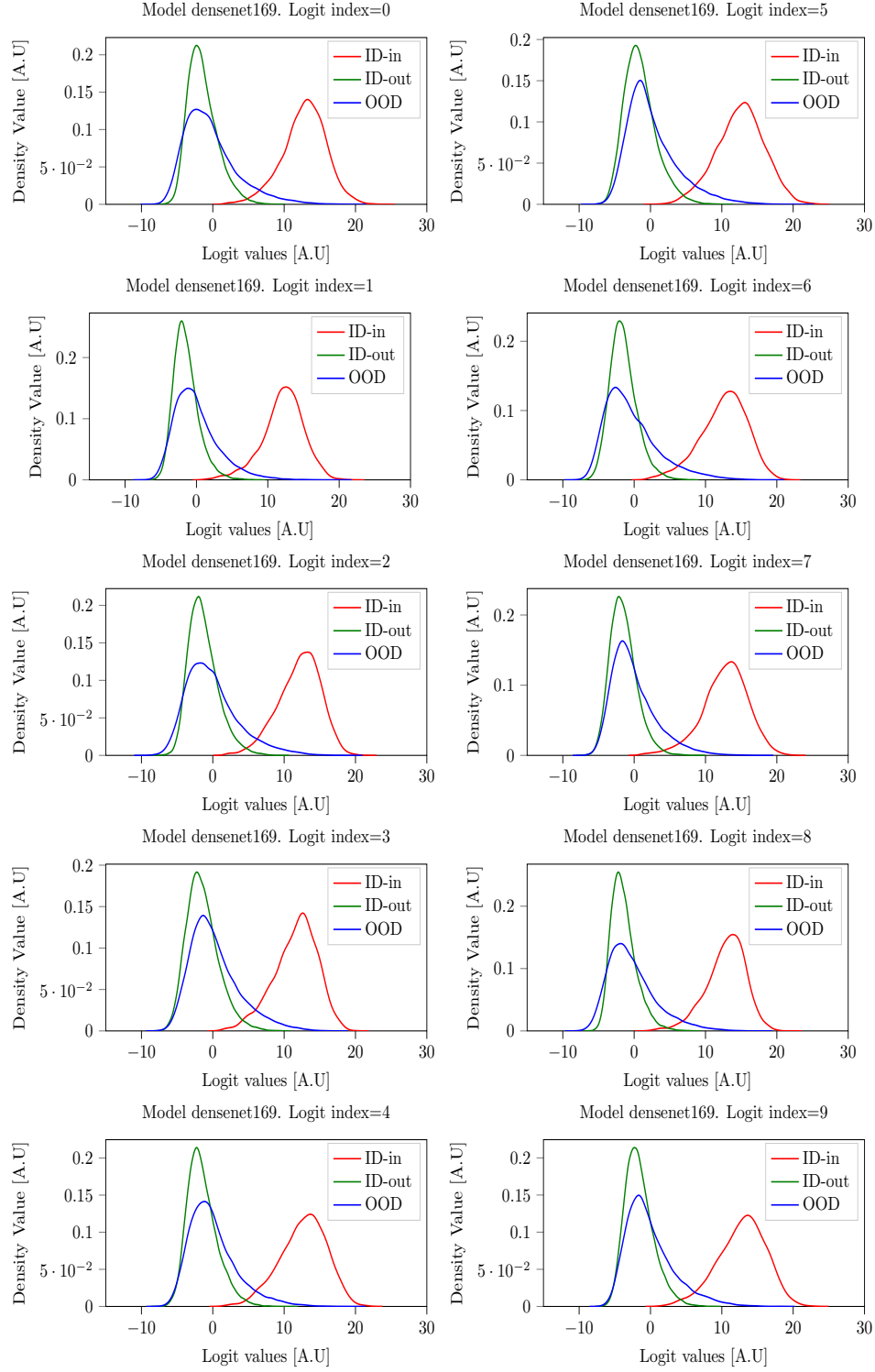


Figure 40: Logit cell densities for CIFAR-10 as ID with Densenet169.

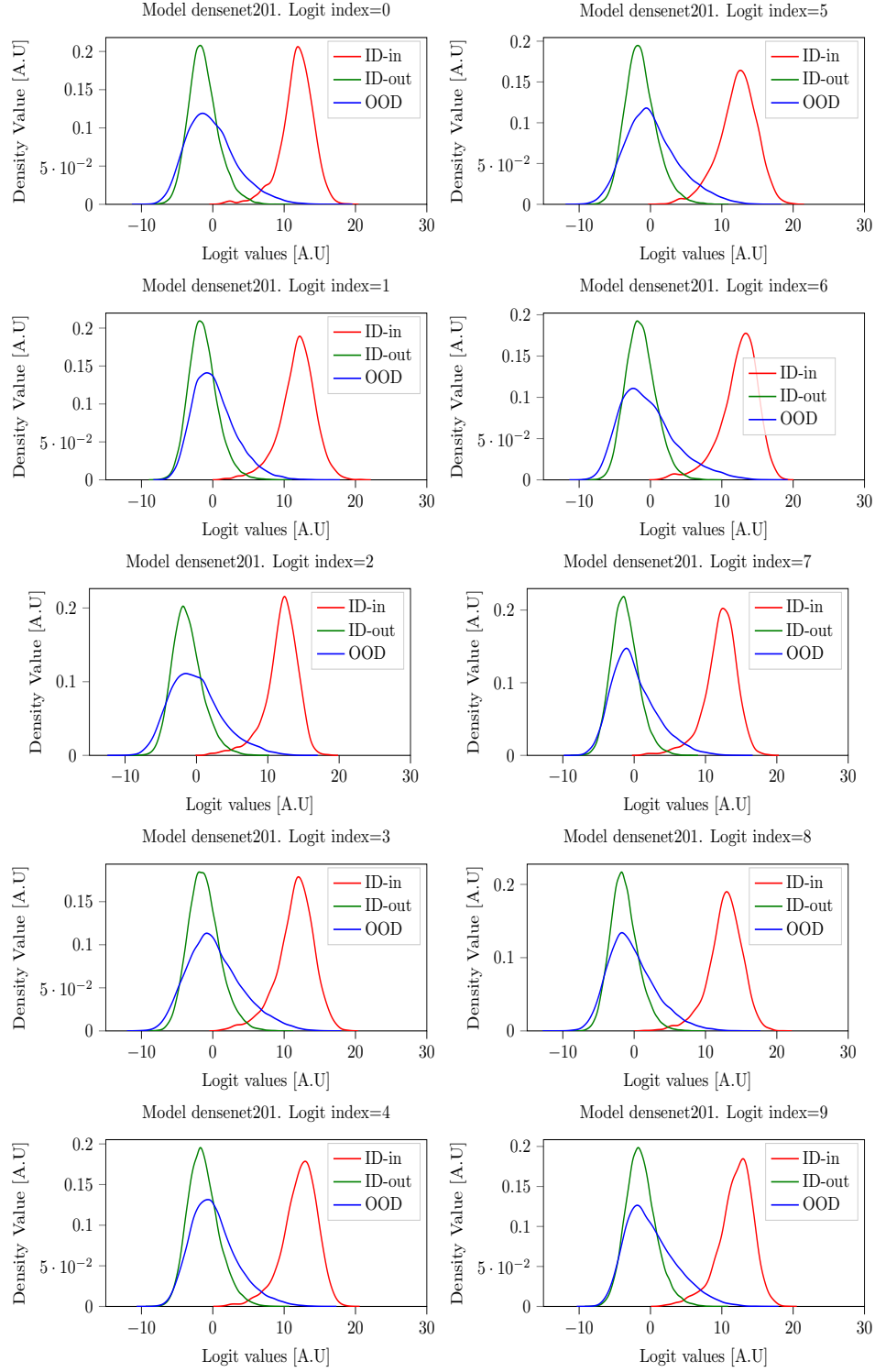


Figure 41: Logit cell densities for CIFAR-10 as ID with Densenet201.

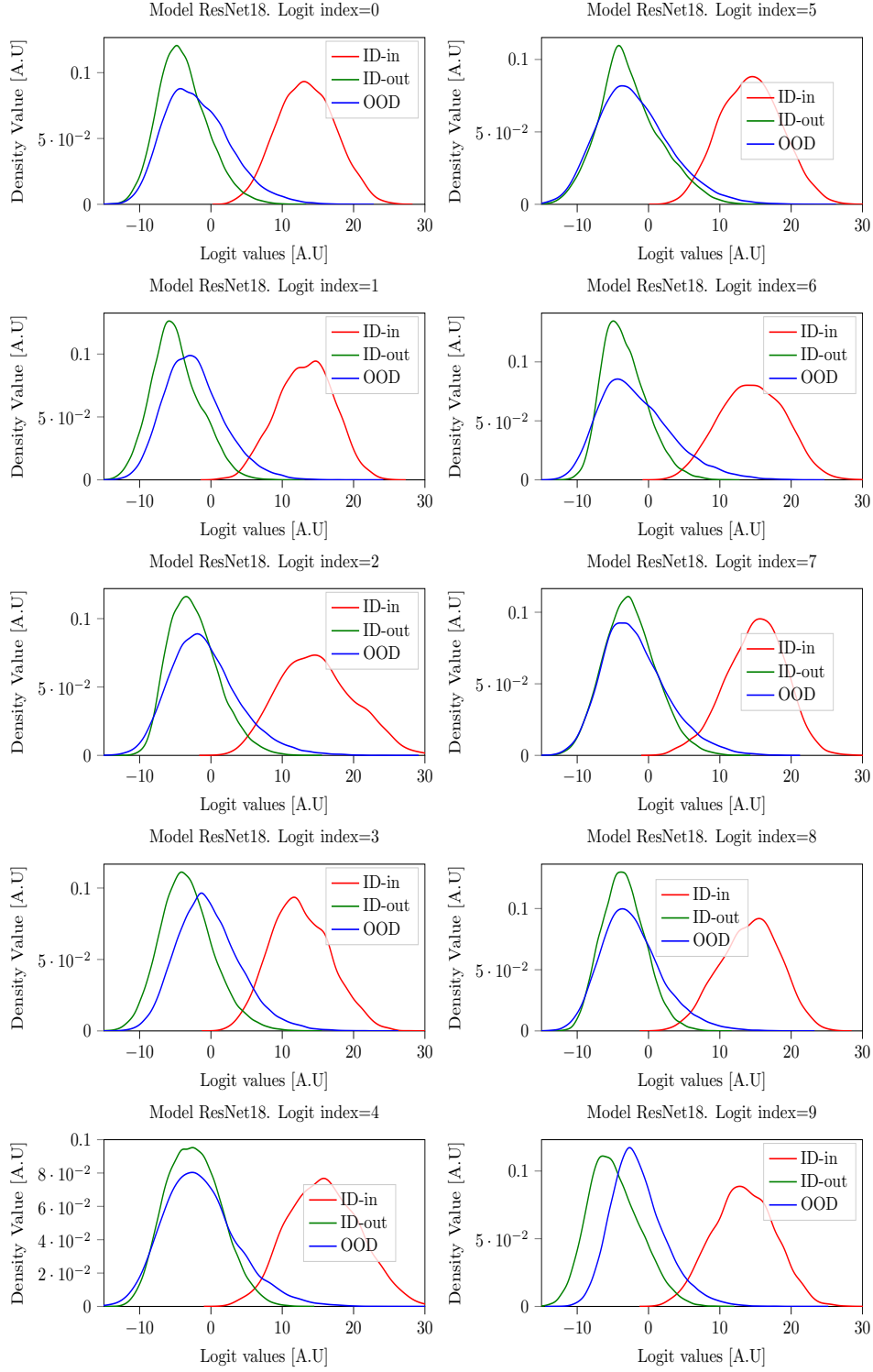


Figure 42: Logit cell densities for CIFAR-10 as ID with ResNet18.

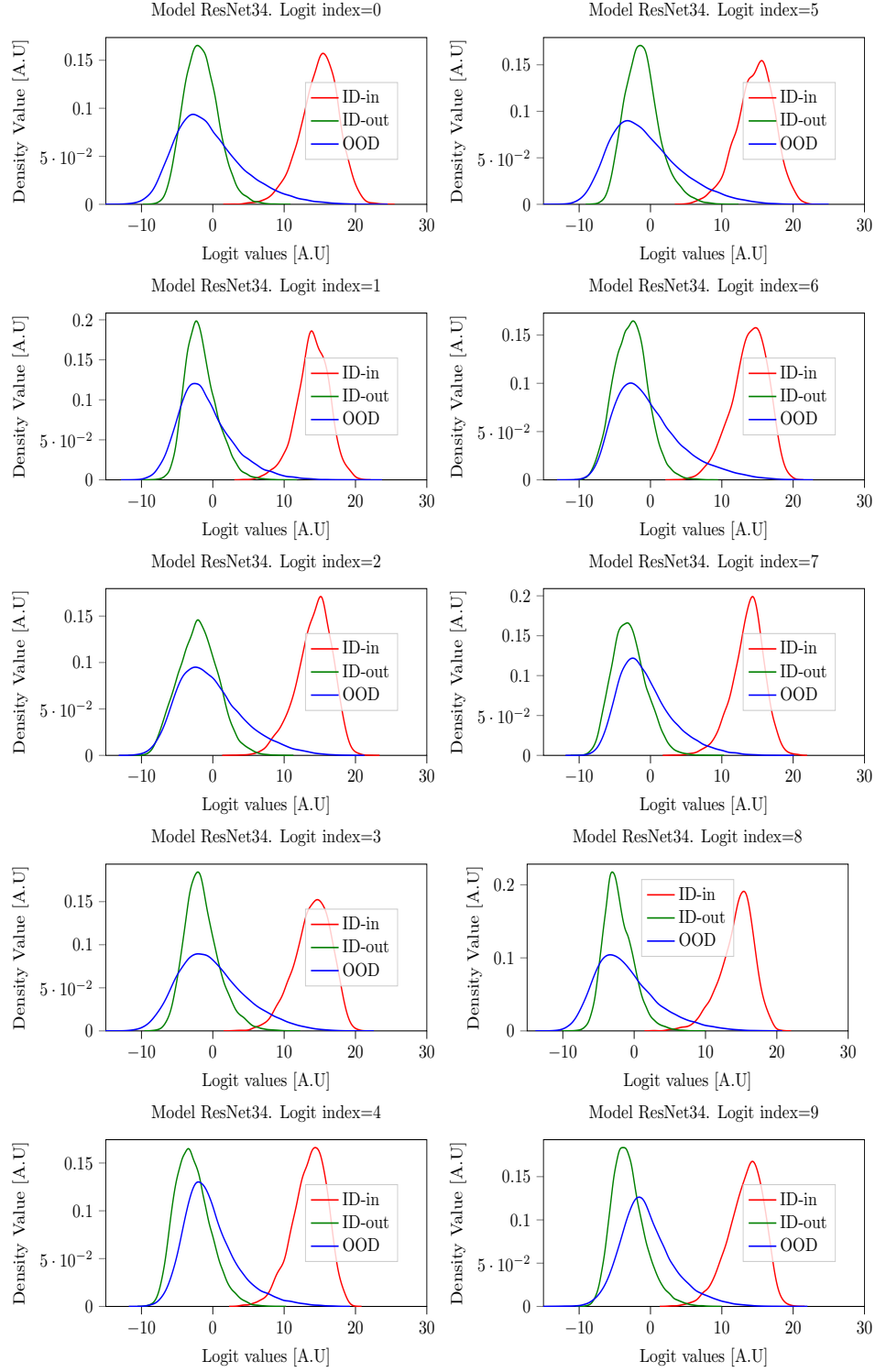


Figure 43: Logit cell densities for CIFAR-10 as ID with ResNet34.

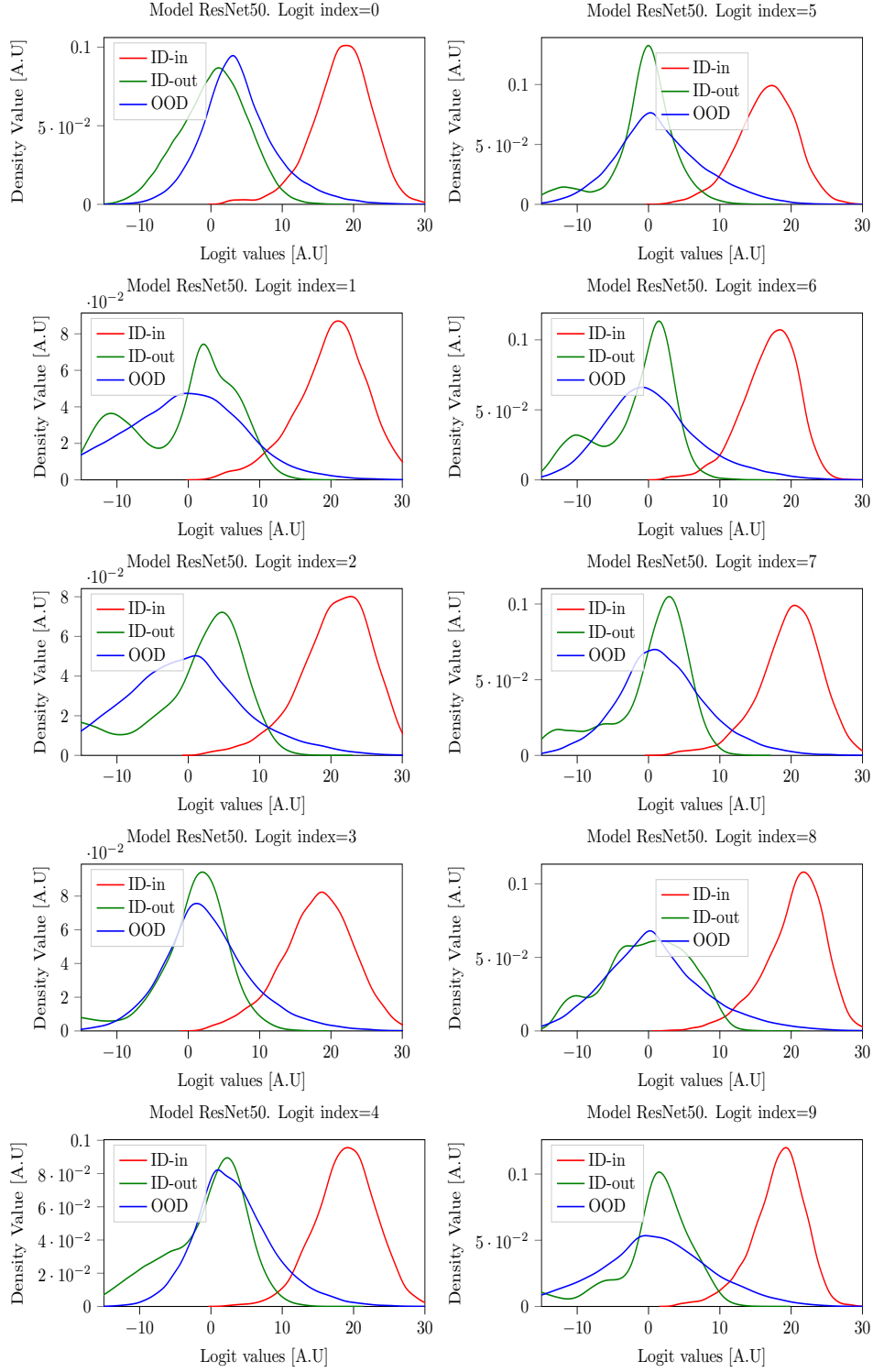


Figure 44: Logit cell densities for CIFAR-10 as ID with ResNet50.



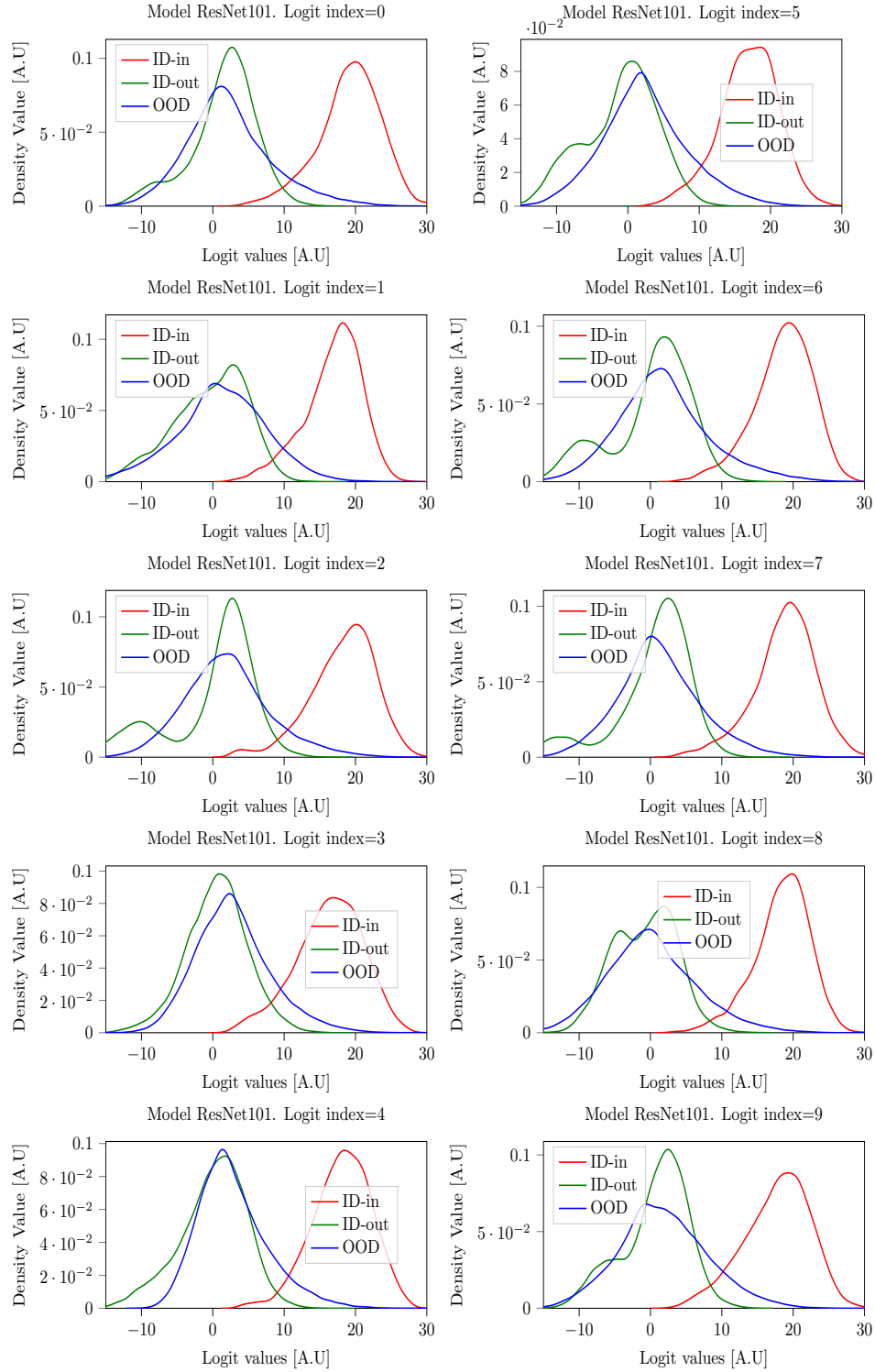


Figure 45: Logit cell densities for CIFAR-10 as ID with ResNet101.

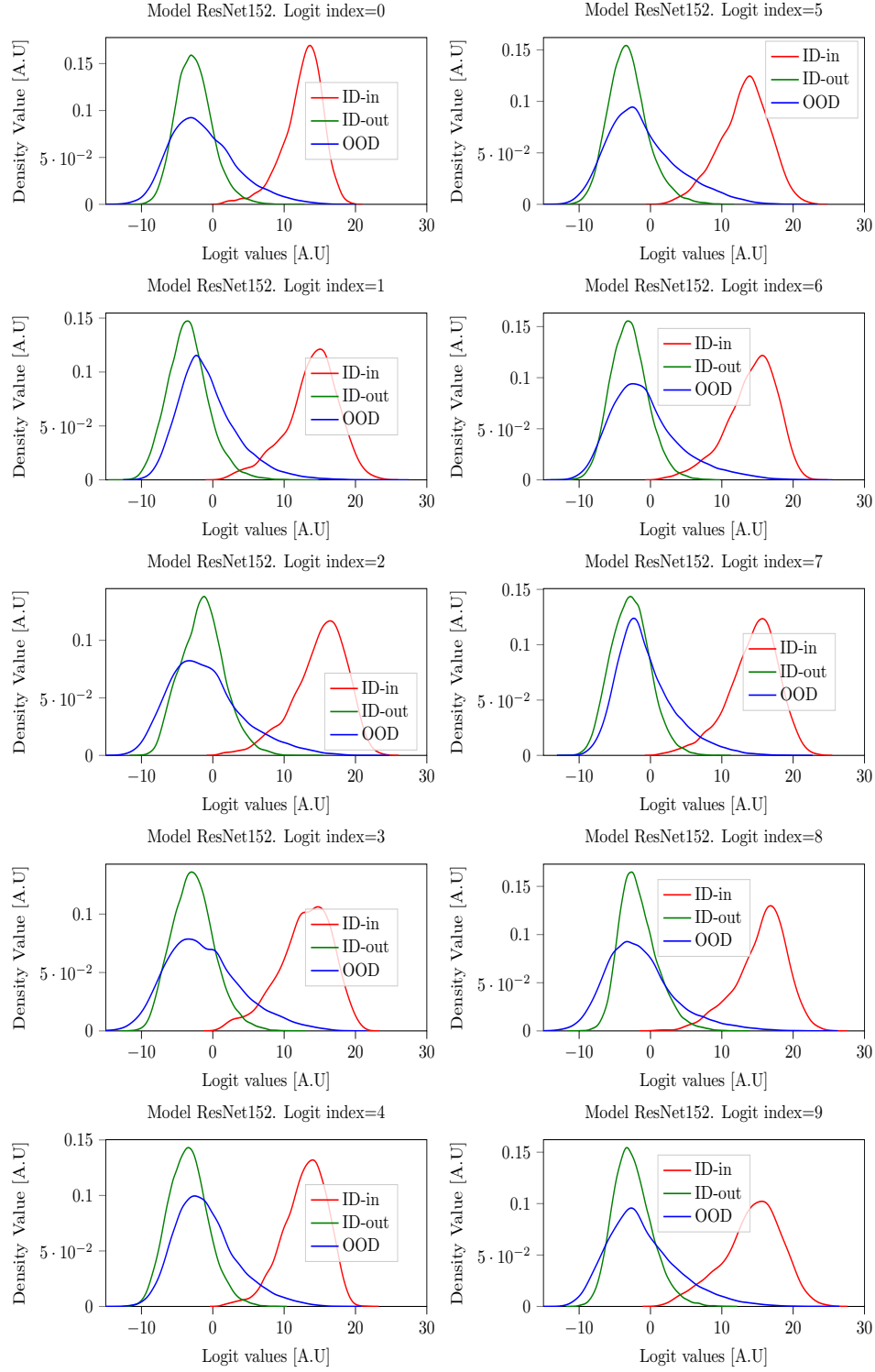


Figure 46: Logit cell densities for CIFAR-10 as ID with ResNet152.

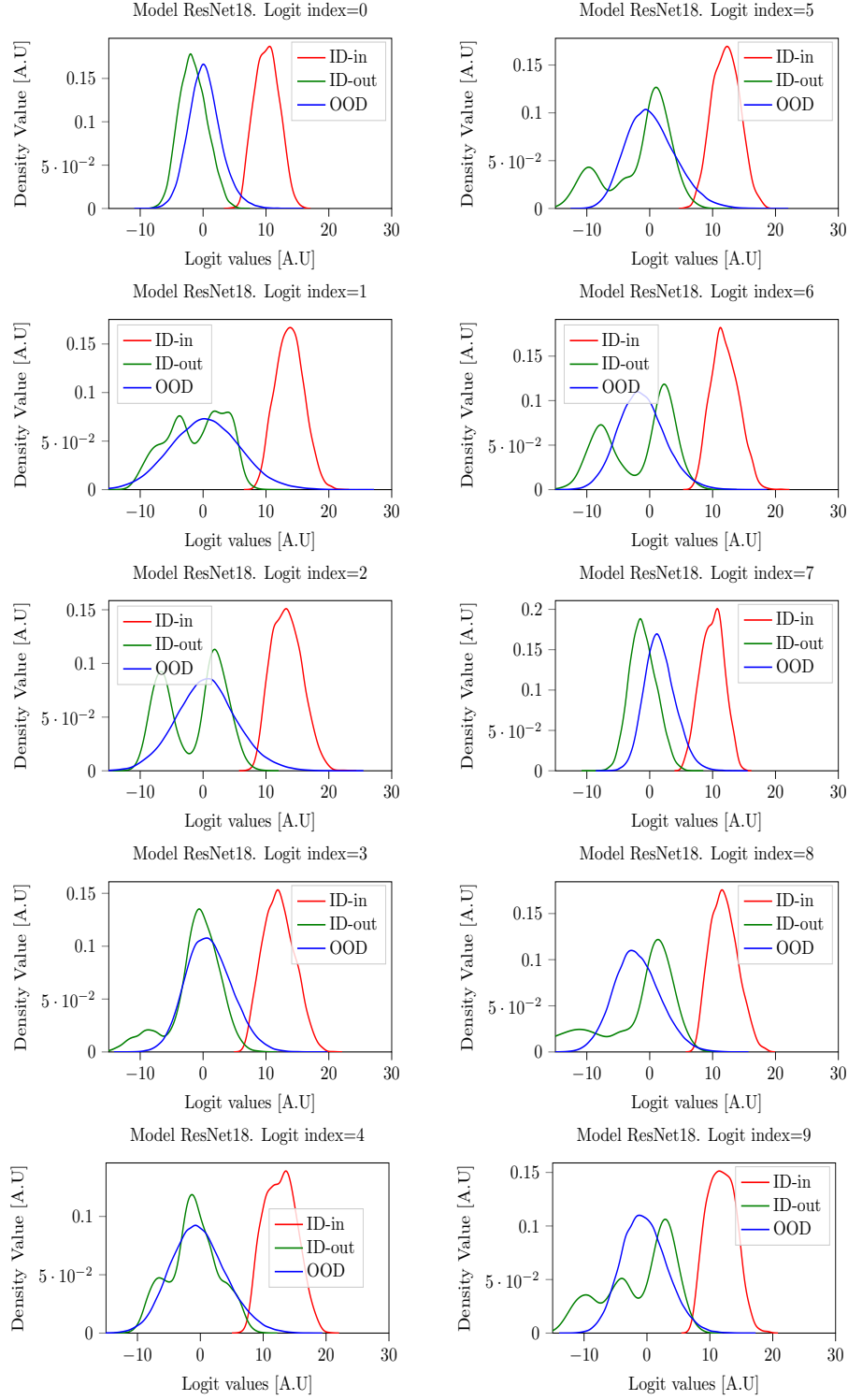


Figure 47: Logit cell densities for SVHN as ID with ResNet18.

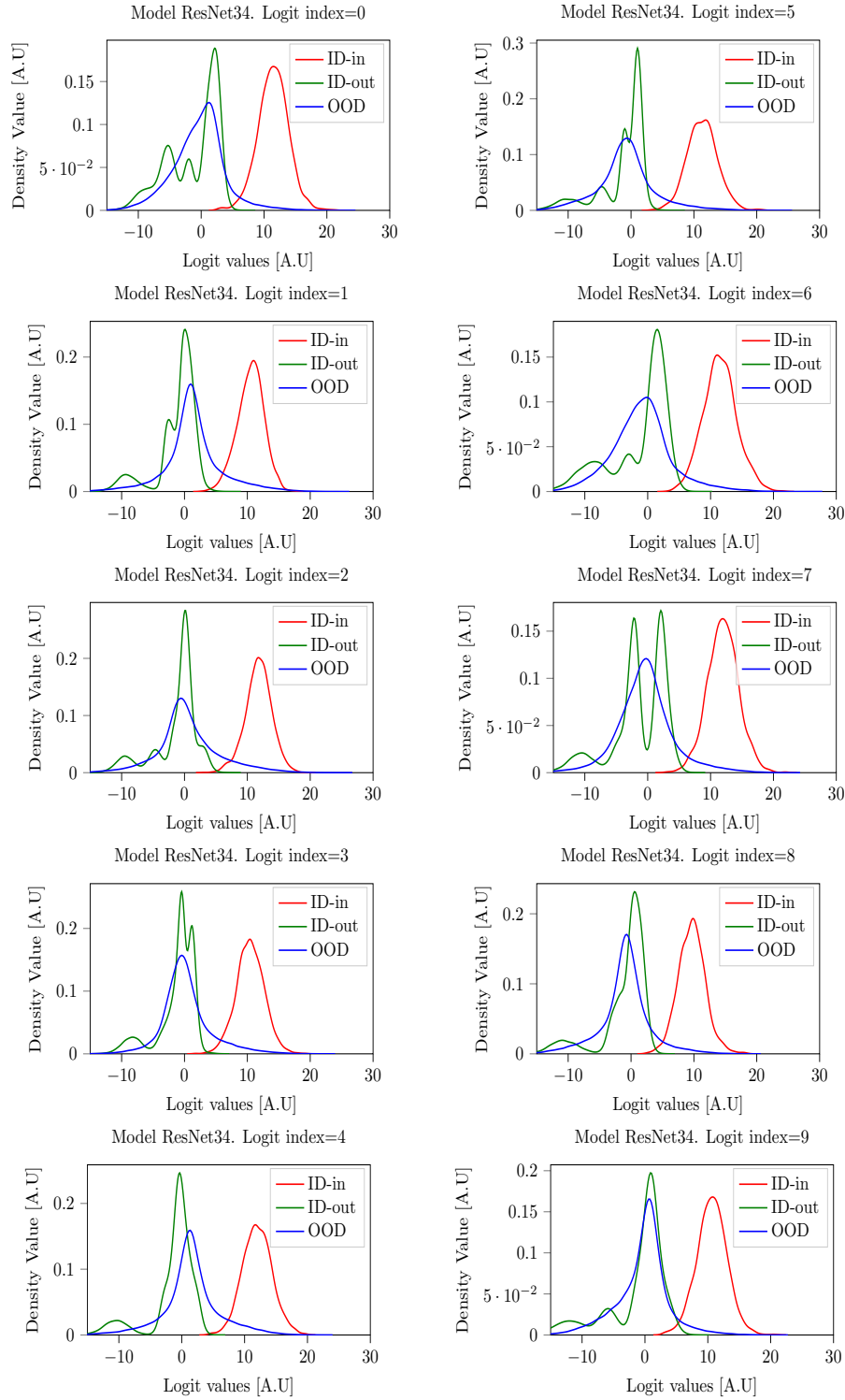


Figure 48: Logit cell densities for SVHN as ID with ResNet34.

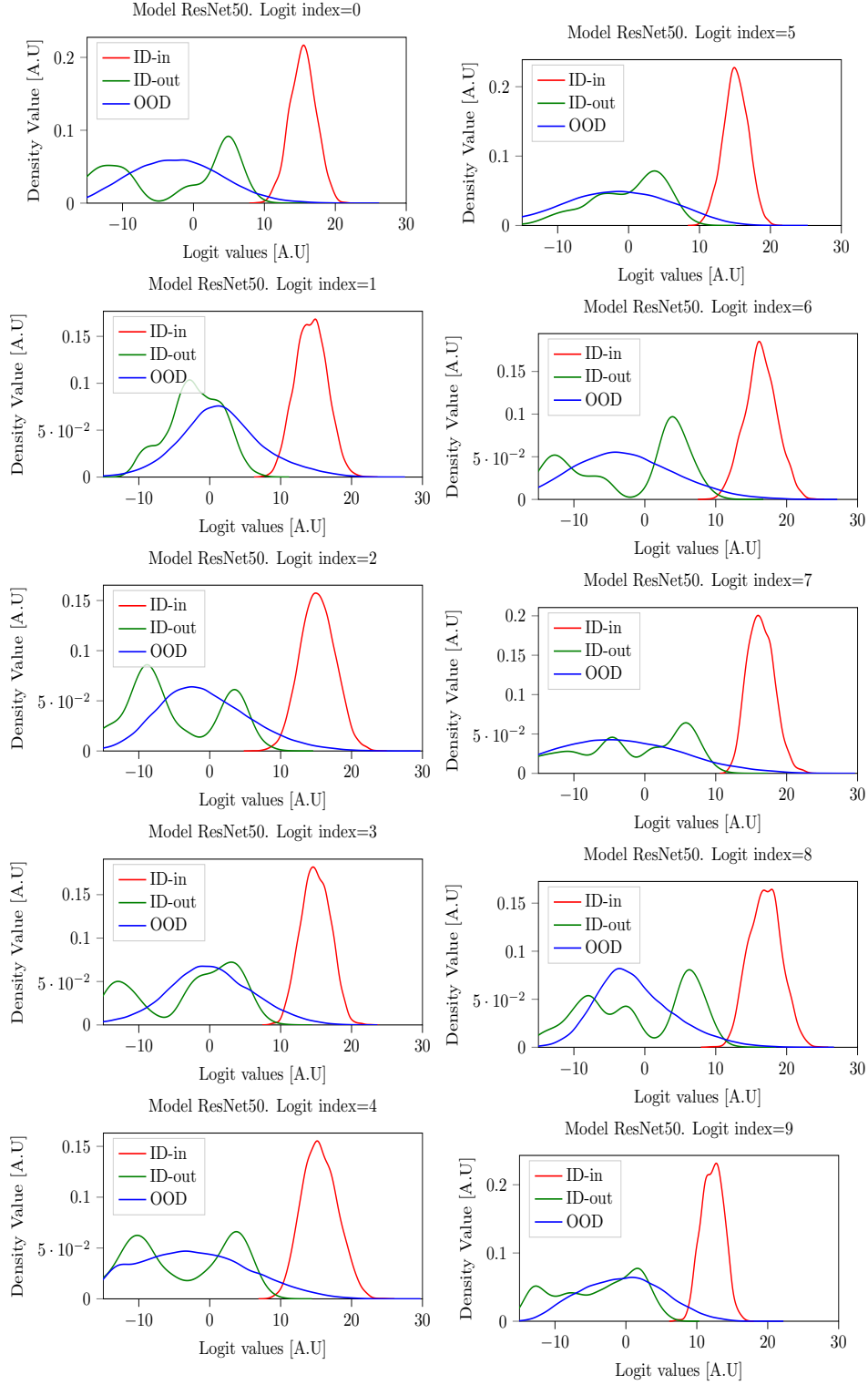


Figure 49: Logit cell densities for SVHN as ID with ResNet50.

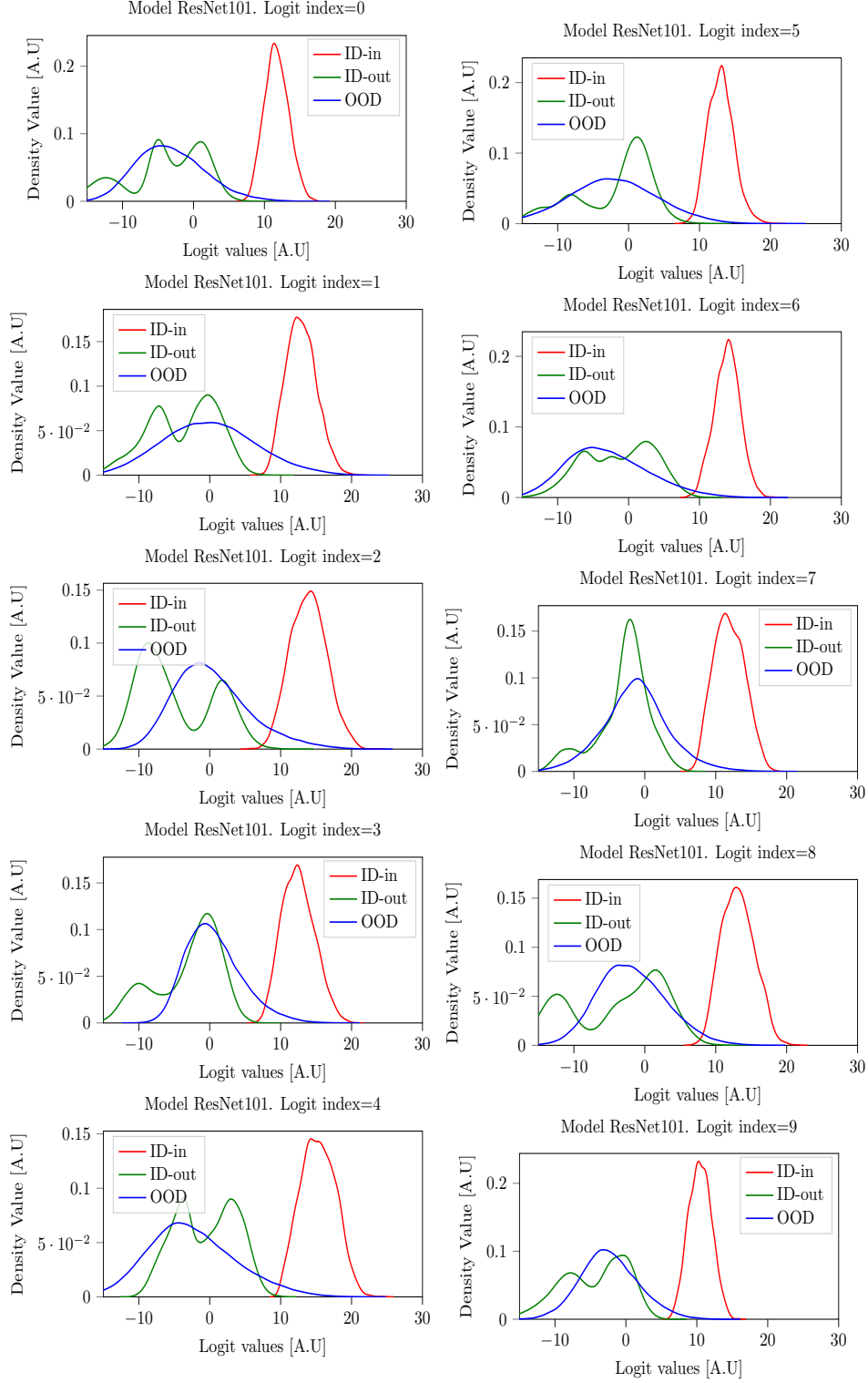


Figure 50: Logit cell densities for SVHN as ID with ResNet101.

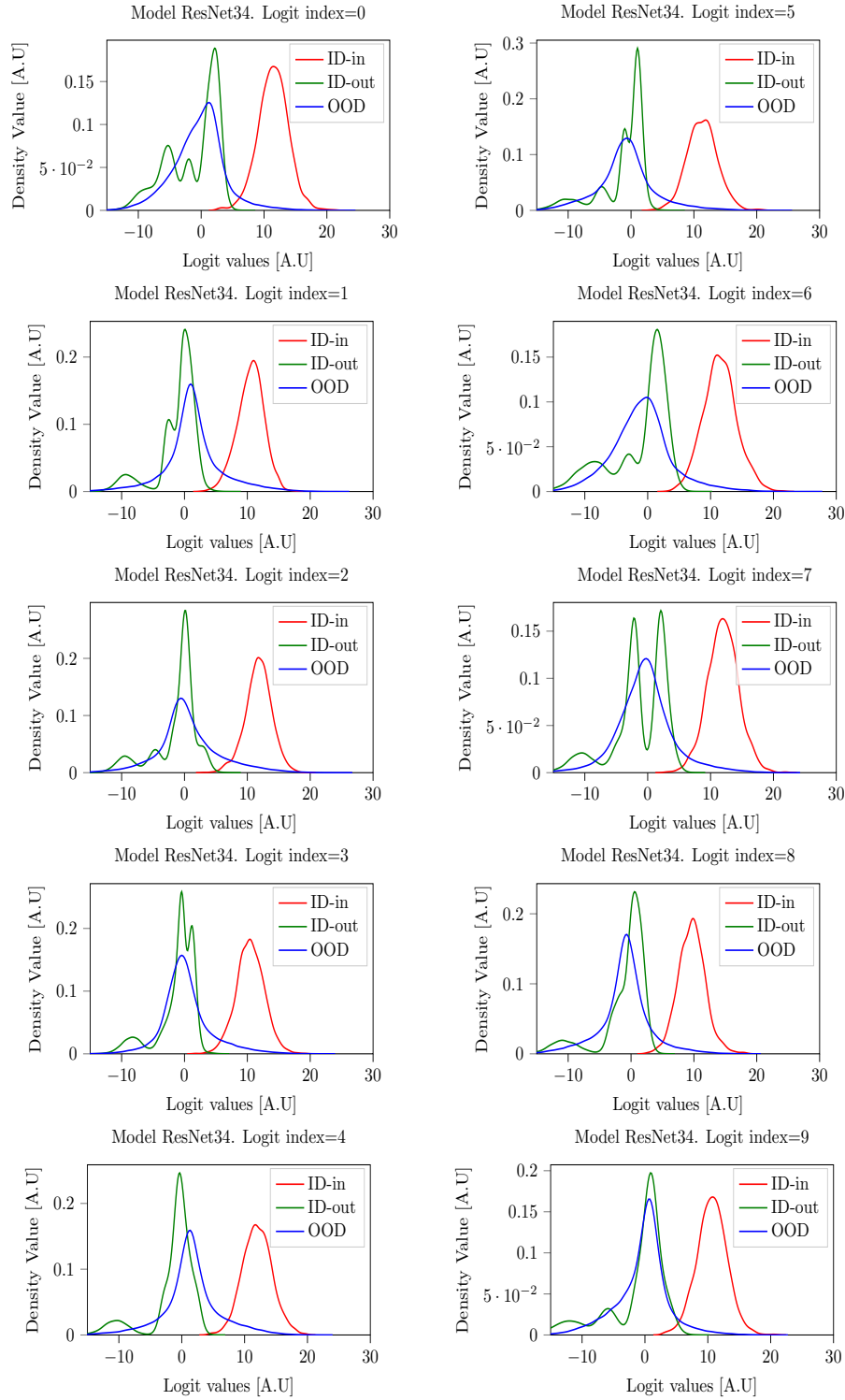


Figure 51: Logit cell densities for SVHN as ID with ResNet152.

## I EXPERIMENTS ON DIFFERENT VISION TRANSFORMERS

Apart from using CIFAR-10 as ID training data, we rerun the same experiments on ViT using SVHN as ID and  $\{\mathcal{D}\} \setminus \text{SVHN}$  as OOD (see fig. 52) Meanwhile figs. 53 to 60 showcase a detailed visualization of the ID and OOD logits for each cell across different variants of ViT for comprehensive comparative analysis.

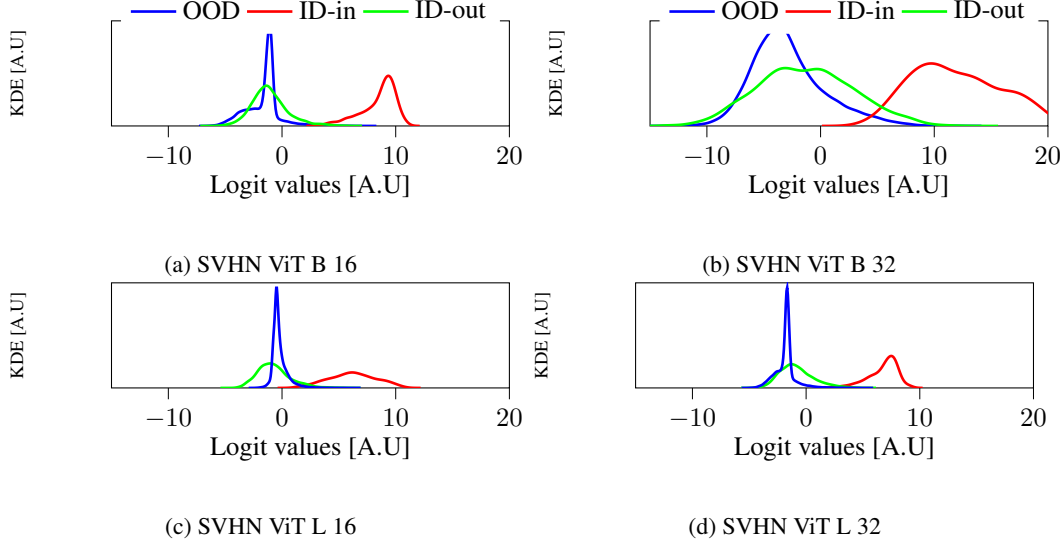


Figure 52: An analysis of the density over aggregated logits across distinct ViT architectures trained on the SVHN as the ID data, while the OOD includes  $\{\mathcal{D}\} \setminus \text{SVHN}$ .



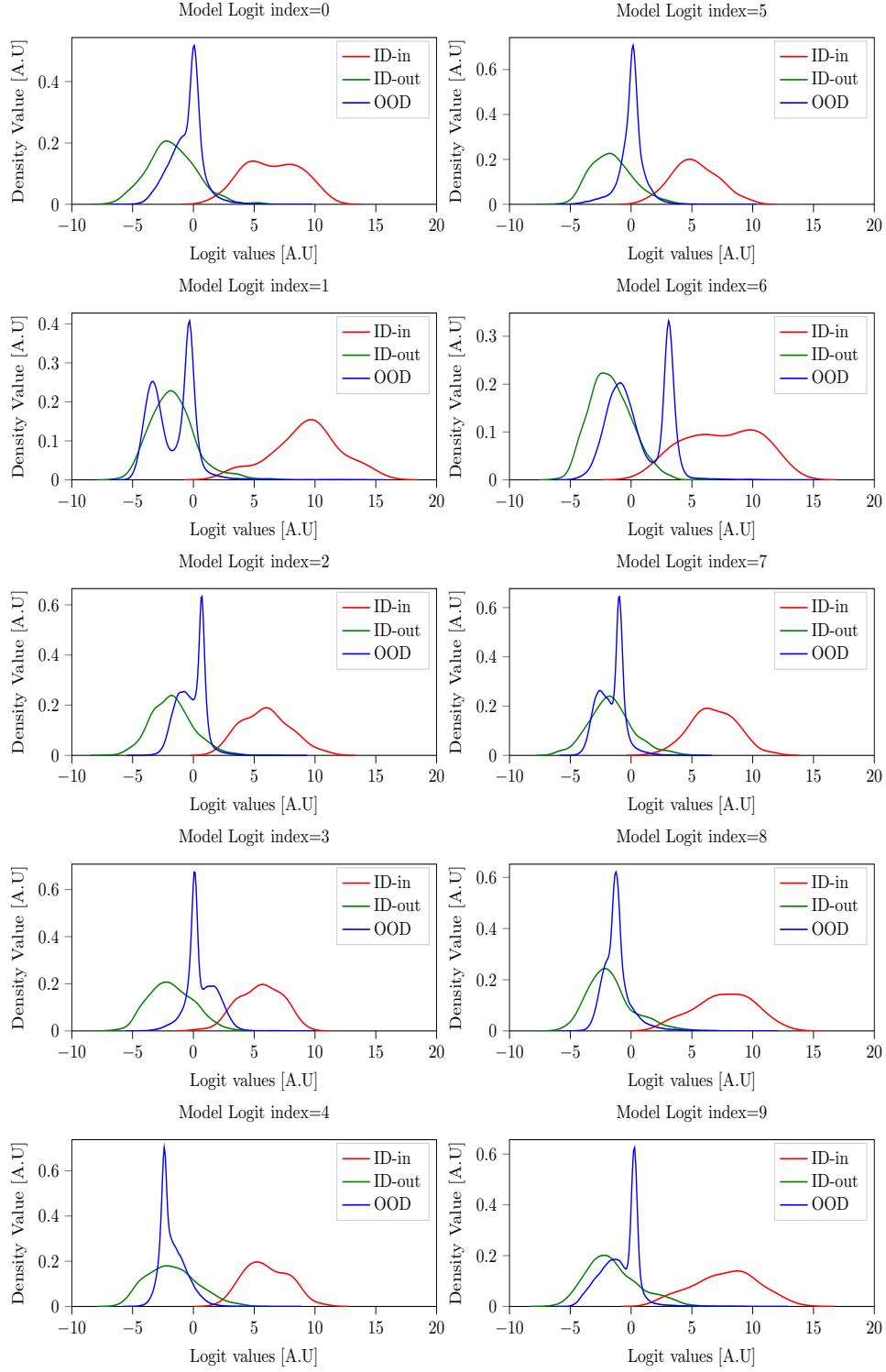


Figure 53: Logit cell densities for CIFAR-10 as ID with ViT-B-16.

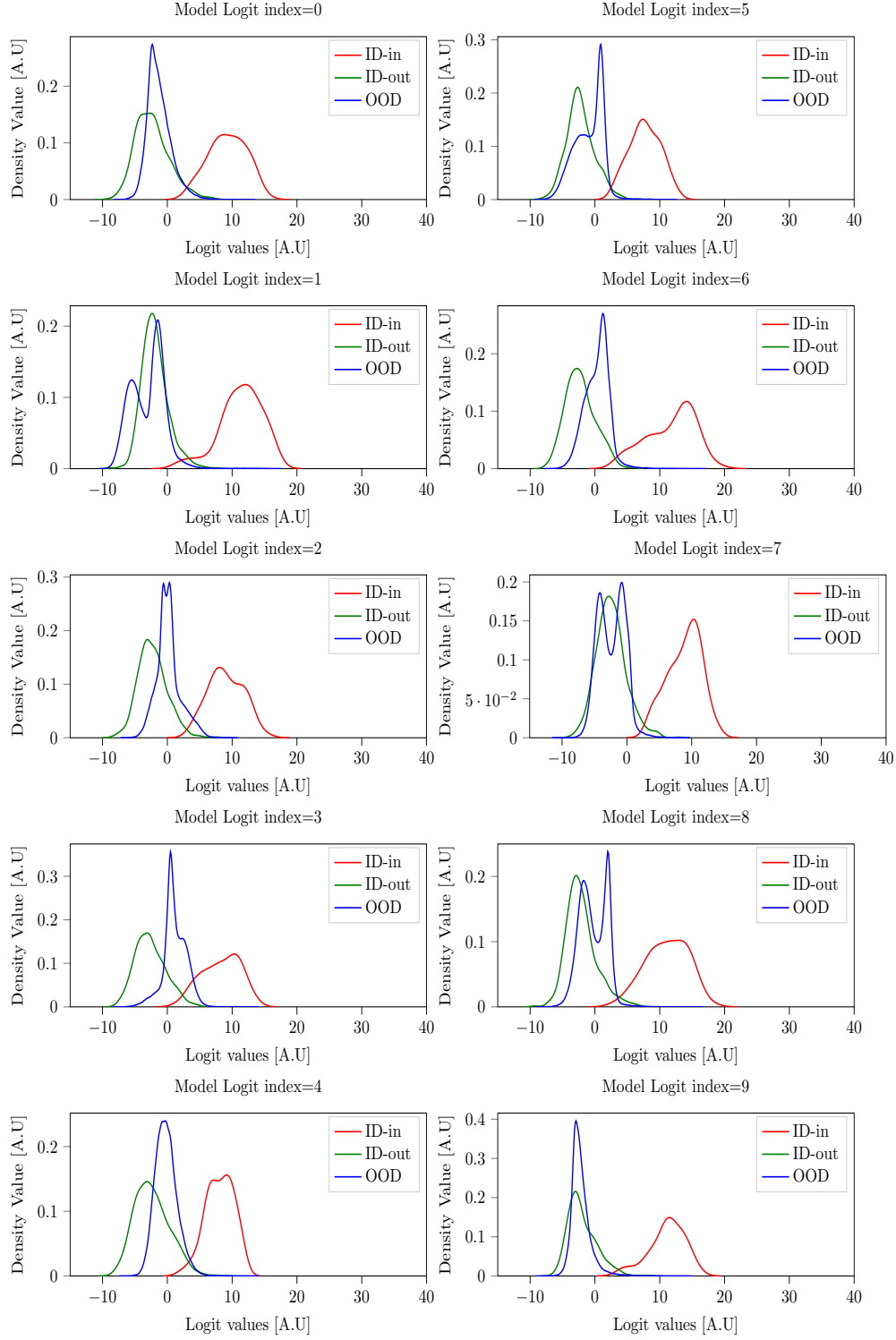


Figure 54: Logit cell densities for CIFAR-10 as ID with ViT-B-32.

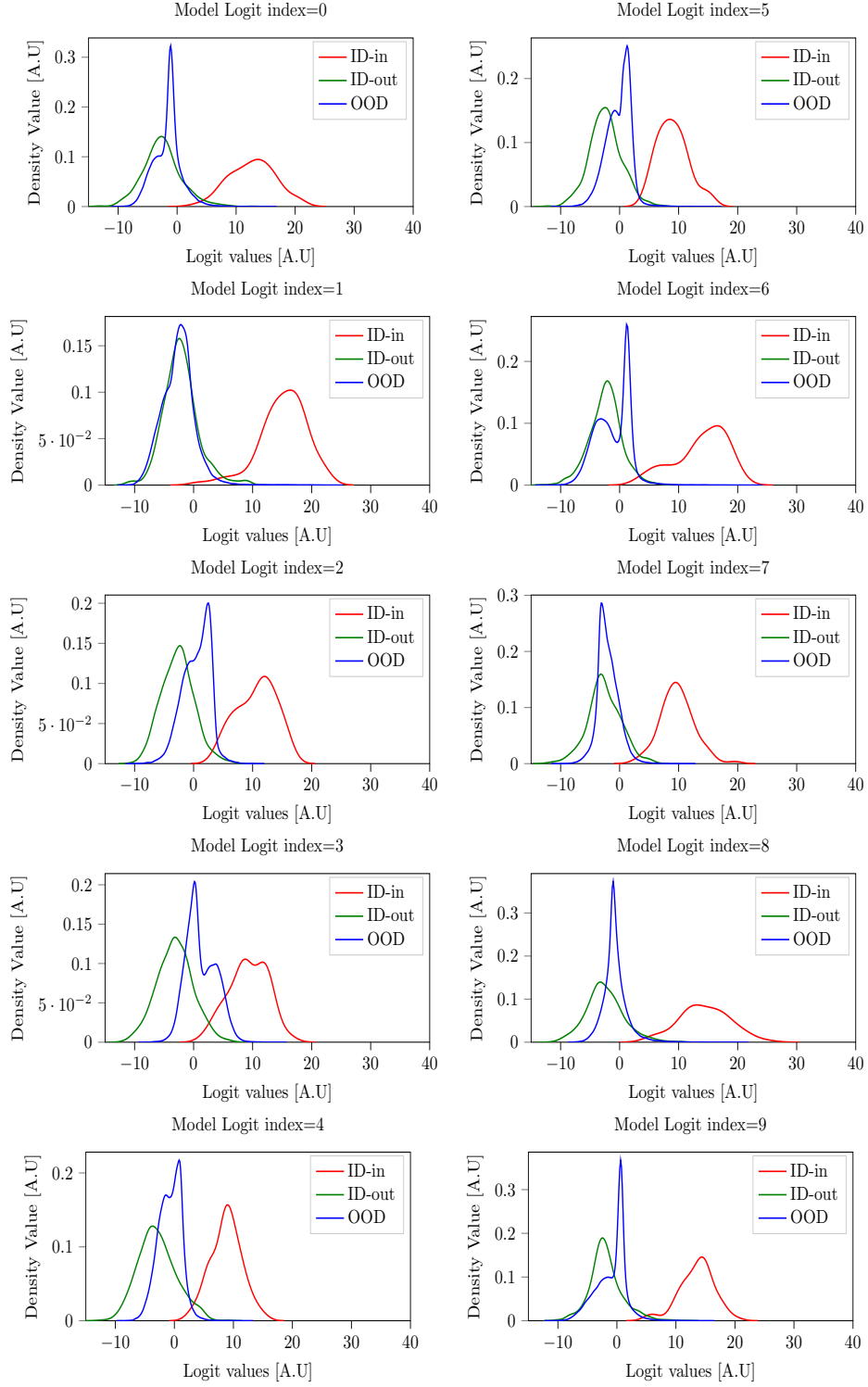


Figure 55: Logit cell densities for CIFAR-10 as ID with ViT-L-16.

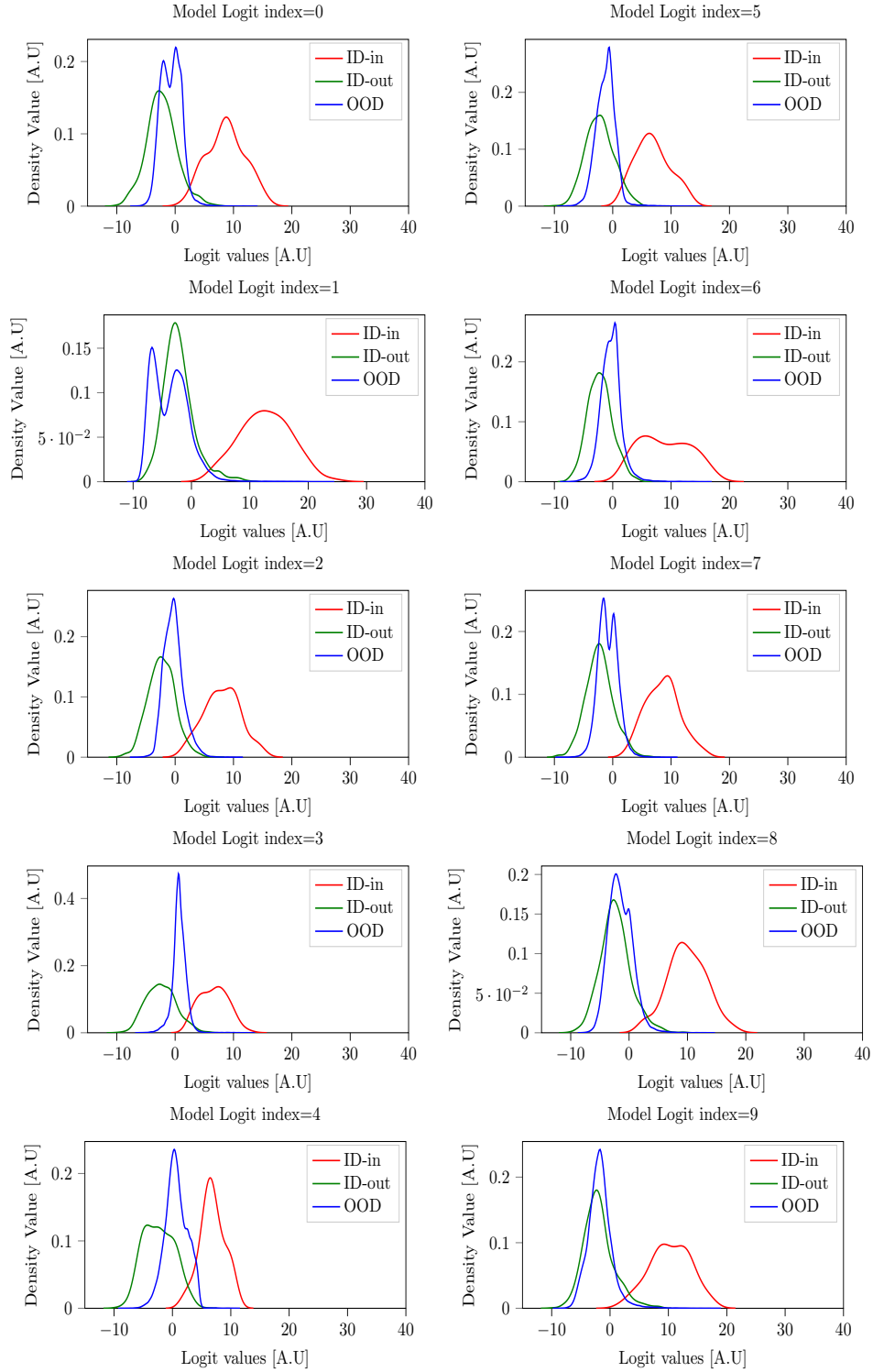


Figure 56: Logit cell densities for CIFAR-10 as ID with ViT-L-32.

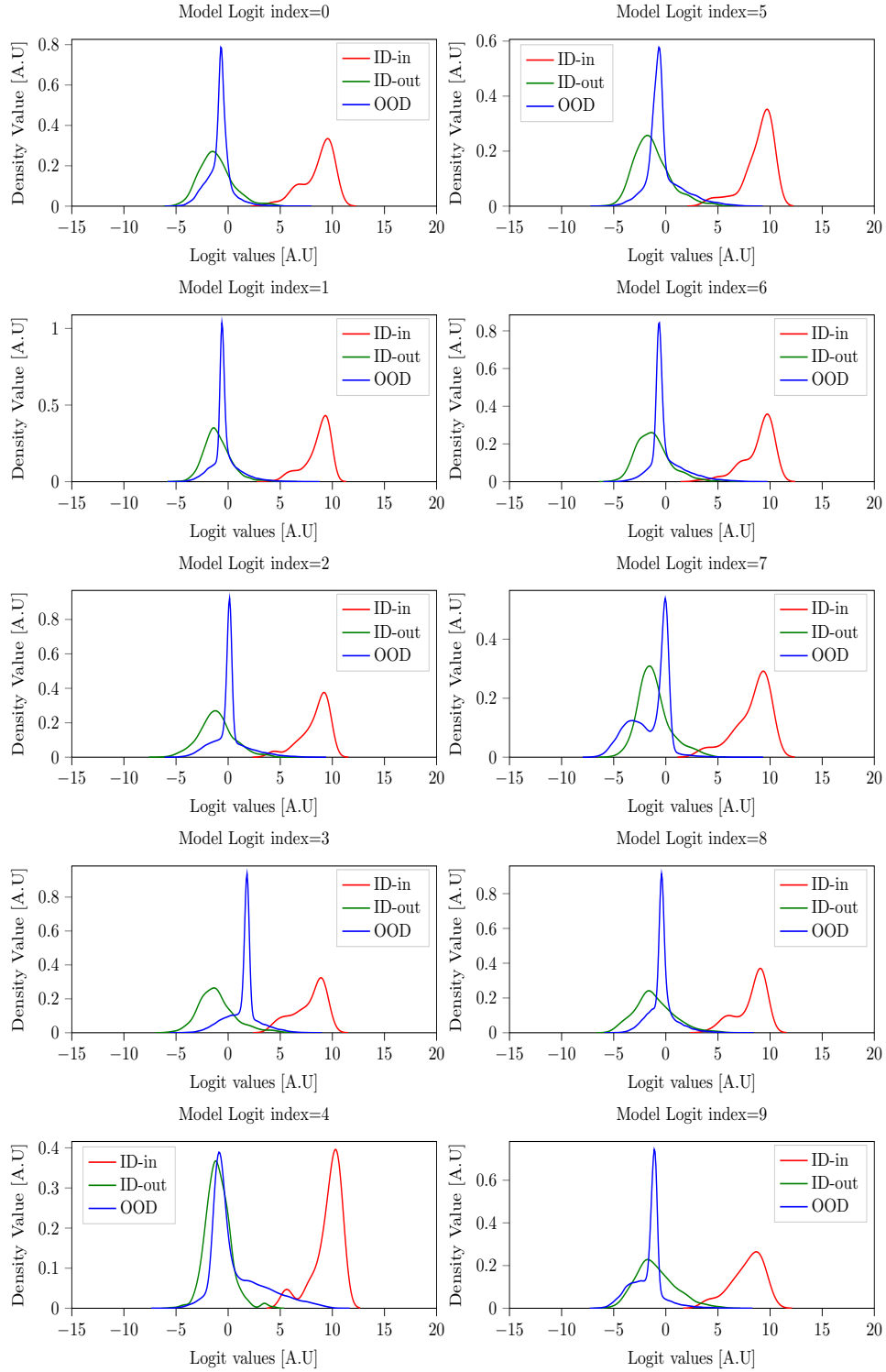


Figure 57: Logit cell densities for SVHN as ID with ViT-B-16.

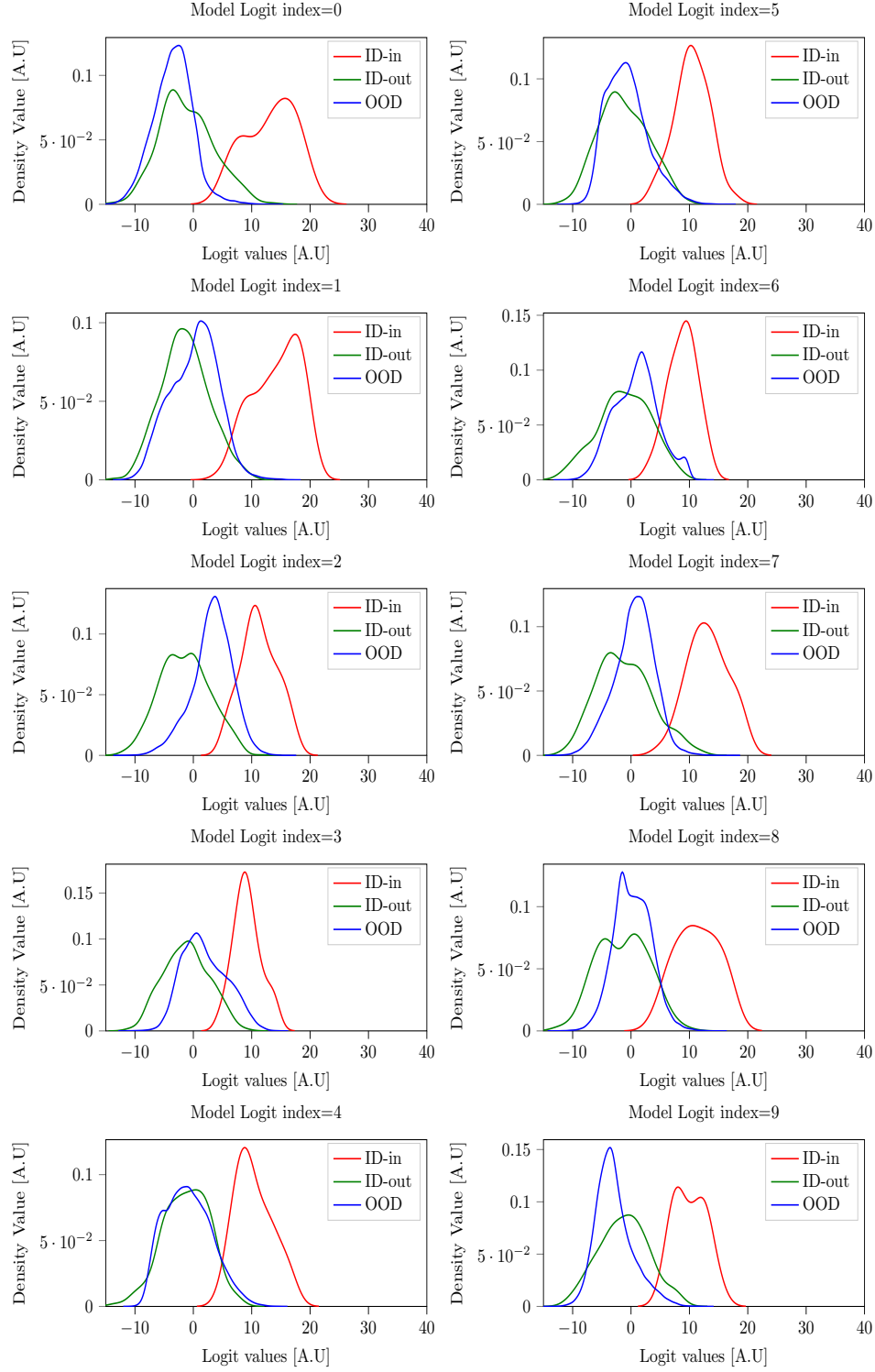


Figure 58: Logit cell densities for SVHN as ID with ViT-B-32.

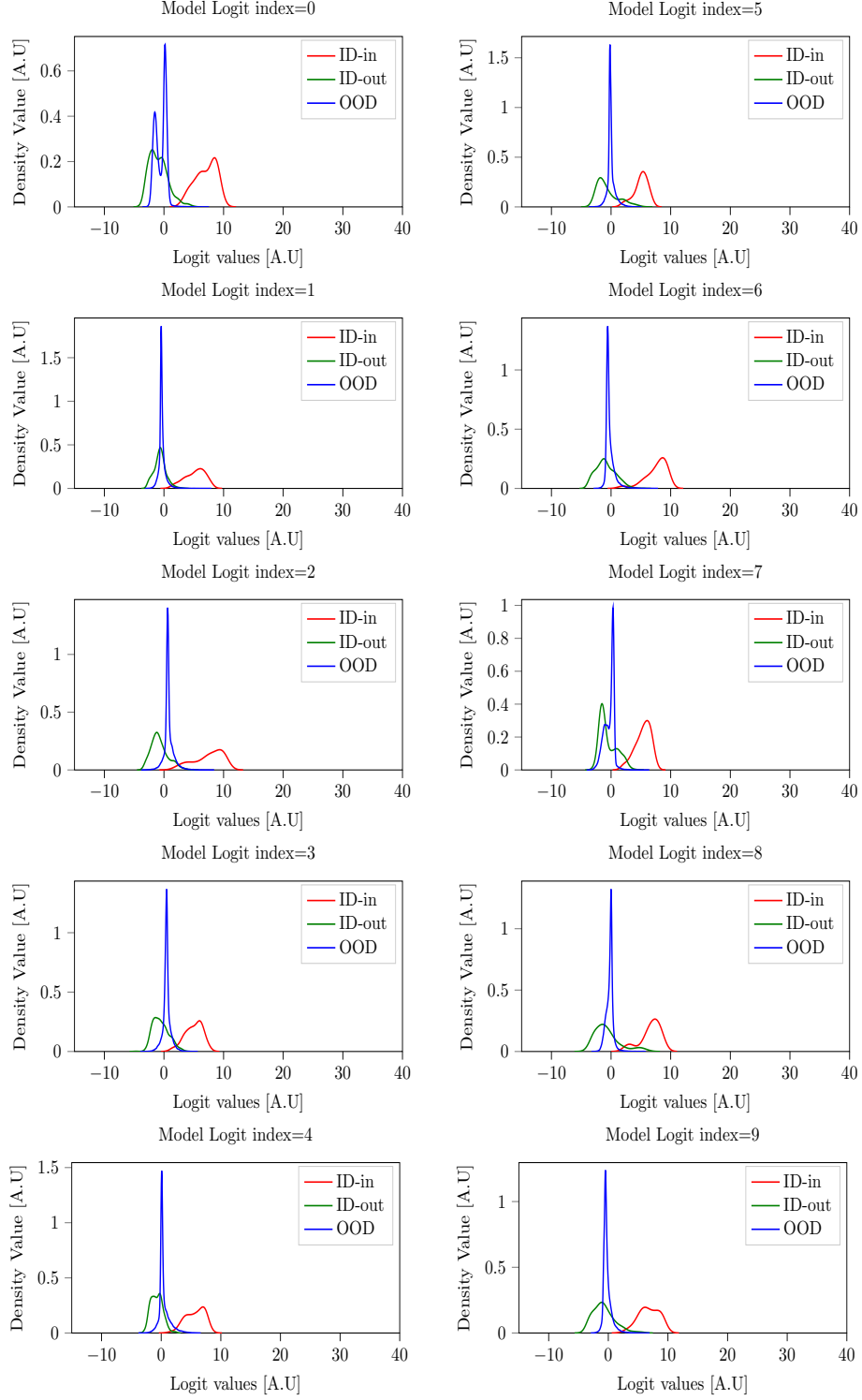


Figure 59: Logit cell densities for SVHN as ID with ViT-L-16.

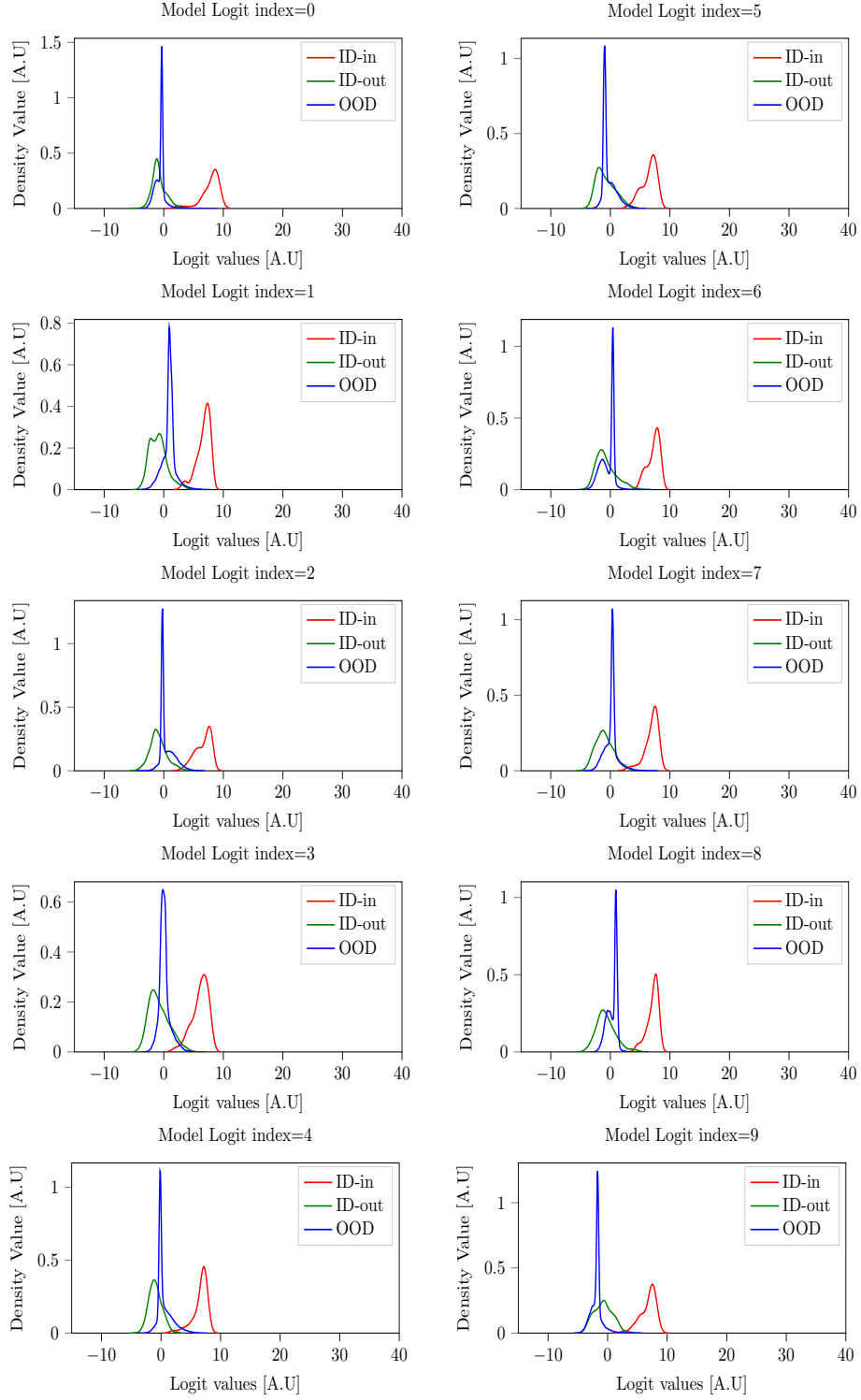


Figure 60: Logit cell densities for SVHN as ID with ViT-L-32.



## J ADDITIONAL EXPERIMENTATION ON GRAYSCALE IMAGE

The classifier model, which is used for this purpose, consists of three convolutional layers followed by two fully connected layers (see table 5). This model is then trained using the Adam optimizer (Kingma & Ba, 2017) via a learning rate of  $lr = 10^{-4}$  with weight decay  $w_{decay} = 10^{-6}$  and with  $\beta_1 = 0.8$  and  $\beta_2 = 0.999$ . A batch size of 256 is applied for both test and train data. No augmentation or regularization is applied to the training process. ReLU activation is utilized at every layer of the network. For a detailed visual analysis of each logit cell, refer to figs. 61 and 62.

Table 5: A convolutional neural network model for the experiment on the fashion-MNIST and MNIST datasets.

Layer (Type)	Matrix	Nr Parameters
Conv2d-1	[64,28,28]	640
BatchNorm2d-2	[64,28,28]	128
ReLU	[64,28,28]	0
Conv2d-3	[128,14,14]	73,856
BatchNorm2d-4	[128,14,14]	256
ReLU	[128,14,14]	0
Conv2d-5	[256,5,5]	295,168
BatchNorm2d-6	[256,5,5]	512
ReLU	[256,5,5]	0
Linear-7	[16]	16,400
ReLU	[16]	0
Linear-8	[10]	170

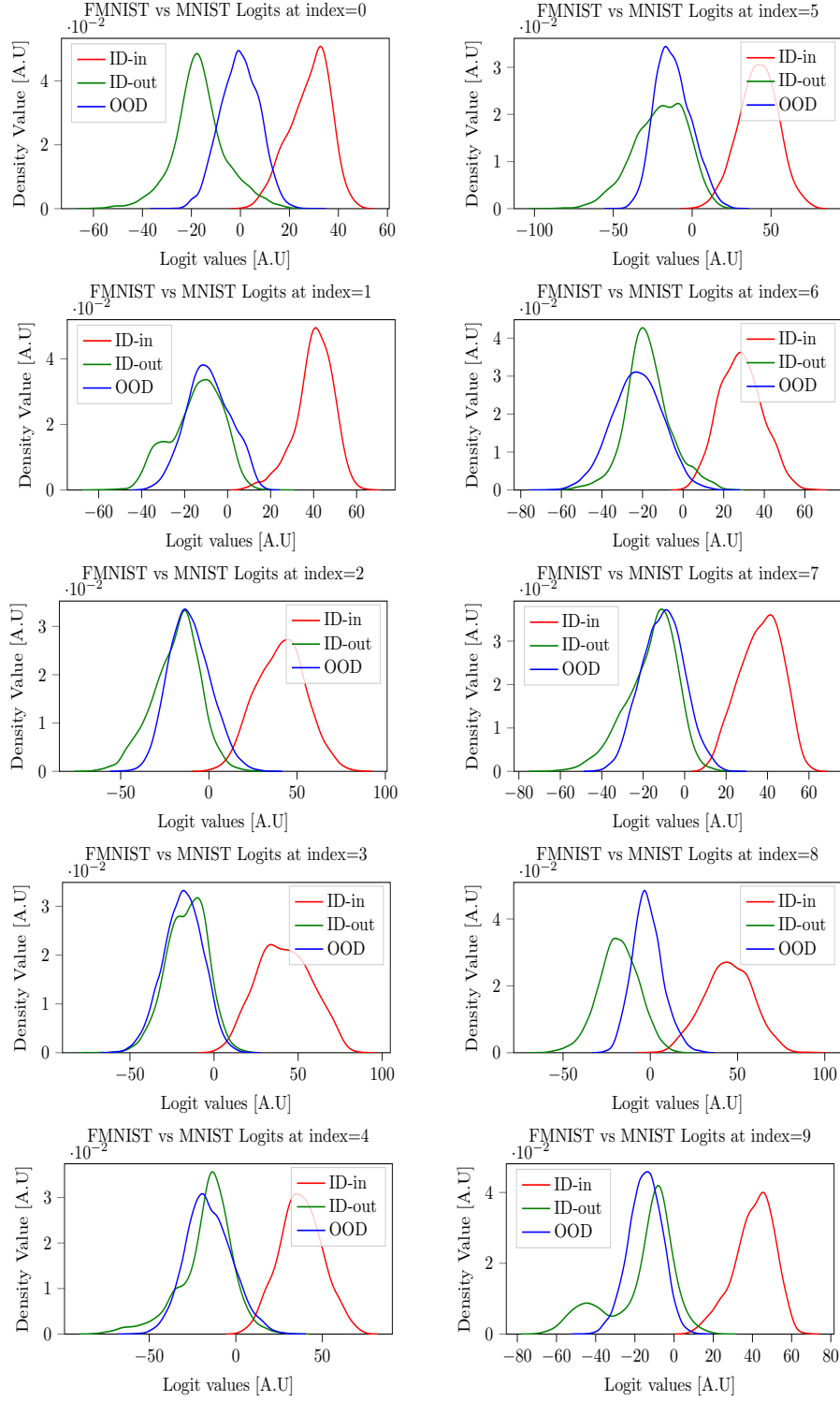


Figure 61: Logit cell densities for FMNIST as ID and MNIST as OOD with the model in table 5.

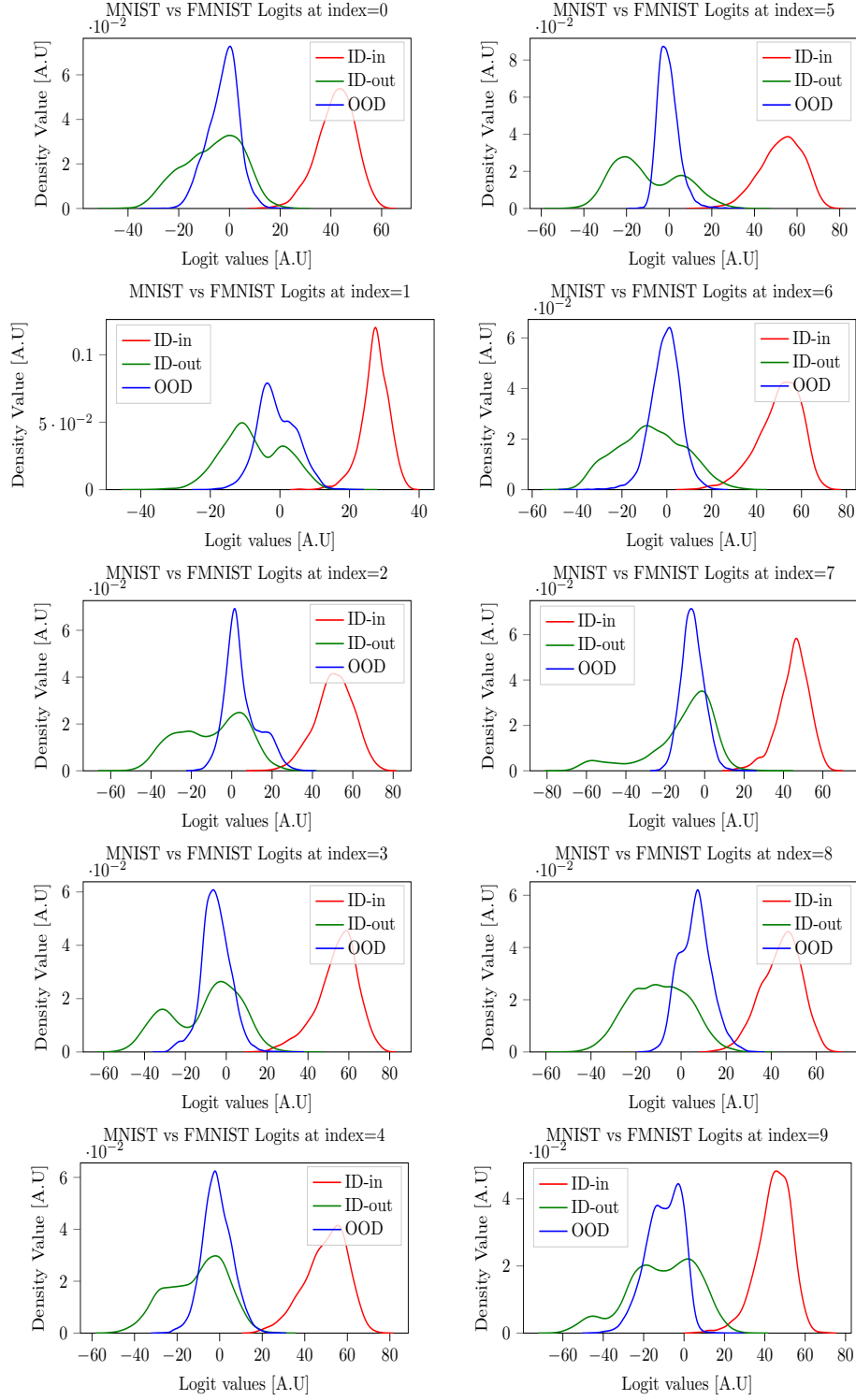


Figure 62: Logit cell densities for MNIST as ID and FMNIST as OOD with the model in table 5.