Hiding in a Plain Sight: Out-of-Distribution Data in the Logit Space Embeddings

Vangjush Komini [1 2], **Sarunas Girdzijauskas** [1 3]
[1] KTH Royal Institute of Technology, Stockholm, Sweden
[2] Qamcom Research and Technology AB, Stockholm, Sweden
[3] RISE Research Institutes of Sweden AB, Stockholm, Sweden
{vangjush,sarunasg}@kth.se

Abstract

Out-of-distribution (OOD) data are detrimental to the performance of deep learning (DL) classifiers, leading to extensive research focused on their detection. Current state-of-the-art OOD detection methods employ a scoring technique designed to assign lower scores to OOD samples compared to in-distribution (ID) ones. Nevertheless, these approaches lack foresight into the configuration of OOD and ID data within the latent space, instead making an implicit assumption regarding their inherent separation. As a result, most OOD detection methods result in complicated and hard-to-validate scoring techniques. This study conducts a thorough analysis of the logit embedding landscape, revealing that both ID and OOD data exhibit a distinct trend. Specifically, we demonstrate that OOD data tends to reside near to the center of the logit space. In contrast, ID data tends to be situated farther from the center, predominantly in the positive regions of the logit space, thus forming class-wise clusters along the orthogonal axes that span the logit space. This study highlights the critical role of the DL classifier in differentiating between ID and OOD logits.

1 Introduction

Deep learning (DL) classification models perform well at generalizing from large datasets, achieving superior classification accuracy compared to many alternatives. They deliver highly accurate predictions when the test data aligns with the training data's distribution. However, their inability to handle out-of-distribution (OOD) data limits their use in critical fields like biomedicine.

For instance, in the classification of bacteria from genome sequences using DL models, it is imperative to consider the presence of novel (i.e., OOD) bacteria. Neglecting these novel entities may result in misclassifying them as known types Ren et al. [2019].

Recent OOD detection methods predominantly operate under the assumption that a classifier, when trained on ID data, intrinsically maps the logits of OOD samples to a distinct spatial location within the logit landscape, divergent from those of ID instances. Thus, differentiating OOD instances from ID data typically involves assigning high likelihood values to the logit (or softmax) location of the ID samples Vyas et al. [2018], Lee et al. [2018], Sun et al. [2022], Gomes et al. [2022], Liu et al. [2020], Komini and Girdzijauskas [2024].

Nevertheless, these strategies do not possess preliminary knowledge regarding the specific locational distribution of OOD samples in the embedding space. Consequently, these techniques attempt complicated density estimations of the ID logits, categorizing those samples that fall beneath a certain likelihood threshold as OOD. Furthermore, the scalability of these methods in relation to the number

Workshop on Scientific Methods for Understanding Deep Learning, (NeurIPS 2024).

of ID classes is problematic, as they require robust performance in density mapping across all ID classes.

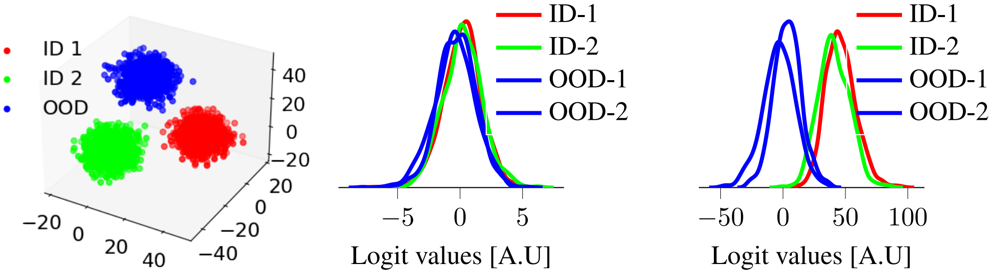
A deficiency in the mapping of even a single ID class may lead to a substantial error rate, where the methods could erroneously classify ID instances as OOD. Instead, our study demonstrates that a well-trained DL classifier, incorporating nonlinearities that suppress negative values (e.g., ReLU), tends to map ID data into distinct class-specific clusters. These ID clusters are situated along orthogonal axes within the positively constrained logit space and are notably separated from the logit space's center. Additionally, we reveal that OOD data are not arbitrarily scattered in the logit space but rather centrally positioned. Consequently, this arrangement ensures minimal overlap between OODs and IDs.

While previous research has explored the separation of OODs and IDs in the logit space Lee et al. [2018], Liu et al. [2020], to our knowledge, this is the first work demonstrating the anticipated configuration of OODs and IDs logits. The noted positioning of OOD and ID logits lays the groundwork for the possible creation of a binary classifier (OOD from ID), which could lead to simpler yet more effective OOD detection models. This study presents the following contributions:
1. An analytical investigation into the spatial allocation of ID data within the logit space.
2. An empirical validation of the observed ID and OOD logit allocation over many models.

2 Method

To illustrate the empirical distributions of ID and OOD logits, both before and after training (see fig. 1a), we employed a binary classification framework utilizing a multilayer perceptron (MLP) model (see Appendix B). To minimize any initial biases, weights of DL models are initialized using a centered Gaussian distribution Glorot and Bengio [2010], He et al. [2015a], while the biases are set to zero. Furthermore, when considering that both ID and OOD data originate from distributions different from the model's initial weight distribution, it is reasonable to assume statistical independence from the model's initialized weights.

This independence leads to minimal covariance between the model's weights and both OOD and ID data. Given that the covariance is defined by $\mathrm{Cov}(\hat{\boldsymbol{x}}, \omega) = \mathbb{E}[\langle \hat{\boldsymbol{x}}, \omega \rangle] - \mathbb{E}[\hat{\boldsymbol{x}}]\mathbb{E}[\omega]$, and recognizing that $\mathbb{E}[\omega] = 0$ initially, we deduce that $\mathrm{Cov}(\hat{\boldsymbol{x}}, \omega) = \mathbb{E}[\langle \hat{\boldsymbol{x}}, \omega \rangle]$. The expected covariance is zero (i.e., $\mathbb{E}[\mathrm{Cov}(\hat{\boldsymbol{x}}, \omega)] = 0$) due to statistical independence, suggesting that the expected dot-product is also zero (i.e., $\mathbb{E}[\langle \hat{\boldsymbol{x}}, \omega \rangle] = 0$). Since DL models primarily operate on dot-product calculations, the ID and OOD data logits are typically centered within the logit space prior to training, as illustrated in fig. 1b.



(a) Raw data

(b) Pre-train

(c) Post-train

Figure 1: Figure 1a shows raw data sampled from a multimodal Gaussian distribution, utilized as training data for a simple MLP binary classifier depicted in Appendix B. In this figure, red and green points denote ID classes for binary classification, and blue points represent OOD data. Figure 1b and Figure 1c demonstrate kernel density estimations (KDE) across logit cells for both OOD and ID data before and after model training, respectively. In both figures, 'OOD-1' and 'OOD-2' refer to KDEs for OOD data within the first and second logits, while 'ID-1' and 'ID-2' represent KDEs for ID class one data in the first logit cell and ID class two data in the second logit cell, respectively.

2.1 Allocation of ID logits towards positive regions

Training a DL classifier involves utilizing the cross-entropy loss, $(i.e., \boldsymbol{H}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}))$, to encourage the prediction $(\hat{\boldsymbol{Y}})$ to closely align with the ground truth $(\boldsymbol{Y})$. When employing one-hot encoding for both $\hat{\boldsymbol{Y}}$ and $\boldsymbol{Y}$, the training objective simplifies to:

$$\boldsymbol{H}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) = -\sum_{i} \boldsymbol{Y}(i)\log(\hat{\boldsymbol{Y}}(i)) = \underbrace{-\boldsymbol{Y}(j)\log(\hat{\boldsymbol{Y}}(j))}_{\boldsymbol{Y}(j)=1} - \sum_{i,i\neq j} \underbrace{\boldsymbol{Y}(i)\log(\hat{\boldsymbol{Y}}(i))}_{\boldsymbol{Y}(i)=0} = -\log(\hat{\boldsymbol{Y}}(j)).$$

Eventually, the minimization cross-entropy loss $\big(i.e., \min[\boldsymbol{H}(\boldsymbol{Y}, \hat{\boldsymbol{Y}})]\big)$ equivalues to maximum likelihood estimation (MLE) $\big(i.e., \min[-\log(\hat{\boldsymbol{Y}}(j))]\big)$.

As training progresses, the softmax layer aims to generate a response close to one for the cell corresponding to the correct class $\big(i.e., \hat{\boldsymbol{Y}}(j) \to 1\big)$. Additionally, owing to the inherent property that the softmax output is confined within a simplex $\big(i.e., \hat{\boldsymbol{Y}}(j)^{\uparrow} + \sum_{i,i\neq j} \hat{\boldsymbol{Y}}(i)^{\downarrow} = 1\big)$, the remaining cells are pushed towards values close to zero $\big(i.e., \hat{\boldsymbol{Y}}(i)_{i\neq j} \to 0\big)$. Hence, optimization can be conceptualized as the maximization of the softmax cell corresponding to the correct class and the simultaneous minimization of cells associated with incorrect classes.

This pattern of maximization-minimization is also observed in other classification losses, such as mean squared error and Kullback-Leibler divergence, which are commonly employed in training DL classification models. This maximization-minimization optimization extends from softmax cells directly to the respective logit cells, as softmax maintains the order of logits. In particular, the logit cell linked to the correct class tries to attain large positive values (see fig. 1c).

However, when suppressing the negative values in an activation layer, the minimization process results in logit values near zero rather than approaching negative values of high magnitudes. Therefore, ID data are projected toward the positive regions of the logit space as shown in Theorem 1 in Appendix C. Given that the logits reach their high positive value for the correct logit cell indicated by the one-hot encoding and approach zero for all other categories, it is evident that the logits for ID samples cluster by class along orthogonal axes within the logit space.

2.2 Central allocation of OOD logits

Since DL classifiers (i.e., ResNet, DenseNet, ViT) are trained using maximum likelihood estimation and their architectures rely on discrete convolutions, which itself relies on dot-products, the training process inherently seeks to maximize the dot-product between the data and the model's parameters as shown in Theorem 1 in Appendix C. Furthermore, maximizing the dot-product between two vectors enhances their linear association. Correspondingly, a pronounced linear relationship between two vectors typically results in a higher co-variability between these two vectors. As a result, maximizing the dot-product enables a high degree of covariance and, by extension, cross-correlation. Hence, one can safely assume that there is a positive relationship between the dot-product of two vectors (i.e., $\langle \hat{\boldsymbol{x}}, \omega \rangle$) and their covariance (i.e., $\mathrm{Cov}(\hat{\boldsymbol{x}}, \omega) = \mathbb{E}[\langle \hat{\boldsymbol{x}}, \omega \rangle] - \mathbb{E}[\hat{\boldsymbol{x}}]\,\mathbb{E}[\omega]$), given that both metrics assess the degree of alignment between the two vectors.

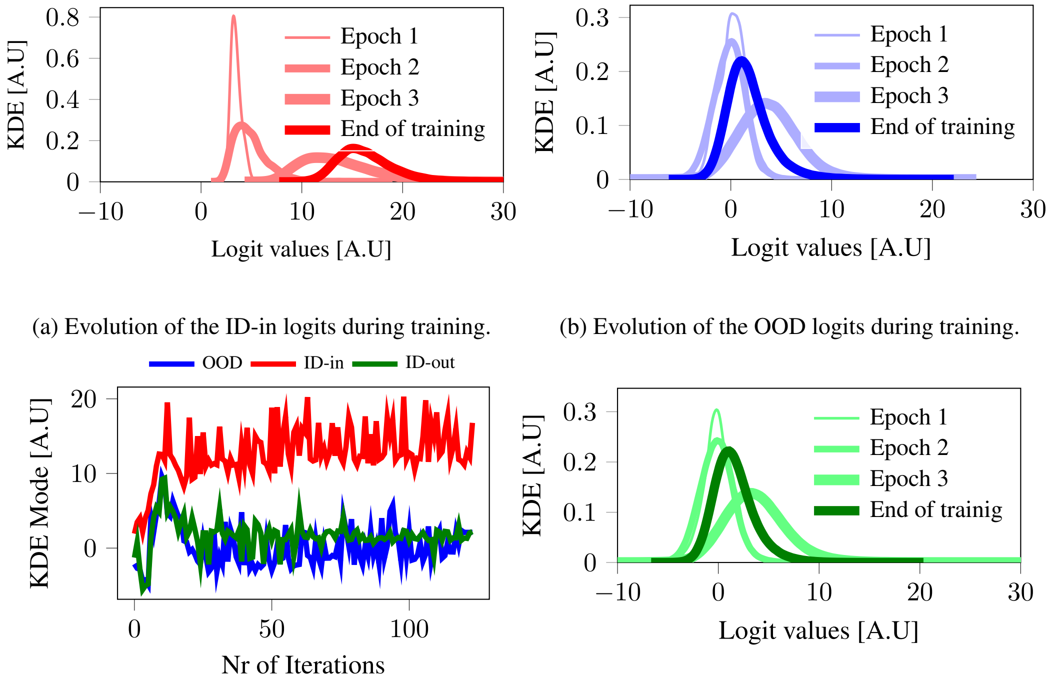
Prior to training, the initial distributions of the model's weights and any given data are disparate. This disparity typically results in both the covariance and the dot-product between the model's weights and any given data being close to zero due to the lack of any established relationship. Thus, an untrained DL model generally steers any input data towards the center of the logit space (see fig. 1b).

After the onset of training, the aim is to align the model's weights with the ID (training) data, maximizing the dot-product and enhancing their covariance, culminating in stronger activation of the logit cell that encodes the correct class (see fig. 1c and Theorem 1 in Appendix C). Considering that OOD and ID data derive from fundamentally different distributions, they inherently exhibit a certain level of statistical independence. This inherent independence implies that the co-variability between OOD and ID data will likely be minimal. Consequently, this also infers that the expected covariance between the model weights and OOD data remains minimal, even post-train.

Given the low covariance between OOD data and the model's weights, their expected dot-product tends to yield smaller magnitudes. Therefore, OOD data tend to remain centered within the logit space even after training (see figs. 1b and 1c).

3 Results

In these experiments, we demonstrate that OOD logits remain near the center of the logit space both before and after training. In contrast, ID logits consistently gravitate towards clusters around class-specific areas in the positive regions of the logit space. Furthermore, we show that these ID clusters align with the orthogonal axis that spans the logit space embeddings.



(a) Evolution of the ID-in logits during training.

(b) Evolution of the OOD logits during training.

(c) OOD and ID mode and validation accuracy.

(d) Evolution of the ID-out logits during training.

Figure 2: Figure 2a presents the density plot across various epochs for the aggregation of ID-in across all logits, while fig. 2b displays the density plot across different epochs for the aggregation of OOD across logits. Similarly, fig. 2d shows the density plot over different epochs for the aggregation of ID-out across all logits. Since the KDE plots are limited to the first three and the final epochs, we included fig. 2c to provide a comprehensive view of the entire trajectory, featuring the peak (i.e, mode) of the density plot for every epoch.

In fig. 2, we empirically illustrate the distribution of ID and OOD logits before and after training. Additionally, we present the evolution of these distributions throughout the training process. To do so, we employed Resnet-9 He et al. [2015b] with CIFAR-100 Krizhevsky et al. [a] as the ID data and CIFAR-10 Krizhevsky et al. [b] as the OOD data. *Additionally, we categorize the ID data as 'ID-in' when the logit values reach their maximum in the cell corresponding to the correct class, as indicated by the one-hot encoding of that class. Conversely, we categorize it as 'ID-out' when this condition is not met.*

We represent the empirical distributions of the logit outputs for both ID and OOD samples via kernel density estimation (KDE) Bishop [2006]. At the beginning of training, one can notice that the densities for both OOD and ID logits are concentrated near zero (see figs. 2a to 2d). While OOD and ID-out logits maintain their central tendency around zero fig. 2c the ID-in logits exhibit a shift towards higher positive values fig. 2a. Analyzing the peak (i.e., mode) of each KDE plot (i.e., ID-in, ID-out, and OOD in fig. 2c), it is evident that ID-in trends towards positive values over time as anticipated by Theorem 1 in Appendix C. Furthermore, the ID-out and OOD logits remain centrally positioned, aligning with our analytical predictions. In addition to the density plots (see figs. 2a, 2b and 2d), which illustrate the aggregation of ID-in, ID-out, and OOD across all logit cells, see Appendix D for detailed visualization of density plots on individual logit cells for a more in-depth analysis.

To delve deeper into the observed phenomenon, we conducted a series of experiments employing various configurations of DenseNet and ResNet (refer to Appendix F) alongside different iterations of Vision Transformers (refer to Appendix G). Additionally, the experiments incorporated the SVHN and CIFAR-10 datasets as in-distribution (ID) samples in distinct trials across all evaluated models, complemented by a broader array of OOD datasets. Beyond variations in architectural models, our investigation extended to the impacts of employing diverse activation functions (see Appendix E). Moreover, our experimental framework was not limited to colored images; we also included experiments with grayscale imagery (see Appendix H).

4 Conclusion

While current research on OOD detection focuses on developing new methods that naturally give higher scores to ID data and, by default, lower scores to OOD samples, this study concentrates on analyzing the differences between OOD and ID logit distributions. Specifically, we demonstrated that ID logits are clustered by class towards the positive region of the logit space, aligning with the orthogonal axis that spans this space. Additionally, OOD logits consistently remain distinct from ID logits, clustering around the center of the logit space.

This behavior of out-of-distribution (OOD) and in-distribution (ID) logits is consistent across various architectures, including different convolutional neural networks and vision transformers. Moreover, the observed pattern remains stable across diverse activation functions.

As a future direction, the observed patterns within OOD, ID-in, and ID-out logits indicate the potential for a novel approach that leverages ID-out logits as proxies for OOD instances. This approach will facilitate the development of a binary classifier neural network designed to differentiate between OOD and ID samples, employing ID-out logits as representative proxies for OOD instances. Consequently, this method addresses OOD detection as a straightforward classification challenge, thereby mitigating the need for threshold-based discrimination methods.

References

Jonathan T. Barron. Continuously differentiable exponential linear units, 2017.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus), 2016.

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*, pages 248–255. Ieee, 2009.

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. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis, 2022.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, 2017.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and D. Mike Titterington, editors, *AISTATS*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org, 2010. URL http://dblp.uni-trier.de/db/journals/jmlr/jmlrp9.html#GlorotB10.

Eduardo Dadalto Camara Gomes, Florence Alberge, Pierre Duhamel, and Pablo Piantanida. Igeood: An information geometry approach to out-of-distribution detection, 2022.

Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015b.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem, 2019.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure, 2019.

Claus Hofmann, Simon Schmid, Bernhard Lehner, Daniel Klotz, and Sepp Hochreiter. Energy-based hopfield boosting for out-of-distribution detection, 2024. URL `https://arxiv.org/abs/2405.08766`.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

Julian Katz-Samuels, Julia Nakhleh, Robert Nowak, and Yixuan Li. Training ood detectors in their natural habitats, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks, 2017.

Vangjush Komini and Sarunas Girdzijauskas. Integrating logit space embeddings for reliable out-of-distribution detection, 2024.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). a.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). b.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks, 2018.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection, 2020.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA, 2013.

Yifei Ming, Ying Fan, and Yixuan Li. Poem: Out-of-distribution detection with posterior sampling, 2022.

Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. How to exploit hyperspherical embeddings for out-of-distribution detection?, 2023.

Diganta Misra. Mish: A self regularized non-monotonic activation function, 2020.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection, 2019.

Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection, 2021.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L. Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers, 2018.

Qizhou Wang, Junjie Ye, Feng Liu, Quanyu Dai, Marcus Kalander, Tongliang Liu, Jianye Hao, and Bo Han. Out-of-distribution detection with implicit outlier transformation, 2023.

Florian Wenzel, Andrea Dittadi, Peter Vincent Gehler, Carl-Johann Simon-Gabriel, Max Horn, Dominik Zietlow, David Kernert, Chris Russell, Thomas Brox, Bernt Schiele, Bernhard Schölkopf, and Francesco Locatello. Assaying out-of-distribution generalization in transfer learning, 2022.

Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R. Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking, 2015.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, 2016.

Table of Content in Appendix

A Related Work

Although the field of OOD detection has been extensively researched, there remains a notable gap: none of the existing works have explicitly investigated the segregation of ID and OOD data within the logit space. Conventional OOD detection methods predominantly classify data by first identifying ID samples, and subsequently labeling all other samples as OOD by default. A recent empirical investigation has not only highlighted the transferability of ID training strategies to OOD detection but also identified a tangible correlation between the robustness of ID training protocols and OOD detection efficacy Wenzel et al. [2022]. This study suggests that refining ID training methods could unlock potential pathways for enhancing OOD detection. Another study examines the influence of pre-trained Vision Transformers (ViT) Vaswani et al. [2023] on ImageNet and reports notable improvements in OOD detection performance Dosovitskiy et al. [2021]. Parallel to these observations, another line of research incorporates outlier data — surrogates for OOD samples — within the training phase. This is achieved through an auxiliary loss term that sharpens the contrast between ID and outlier inputs, potentially strengthening OOD detection Katz-Samuels et al. [2022], Hendrycks et al. [2019], Wang et al. [2023], Du et al. [2022], Ming et al. [2022]. Complementing these approaches, there has been a significant effort to restrict the classification of ID data into a hyperspherical embedding, which intrinsically helps OOD detection Ming et al. [2023].

Another line of research assumes an inherent separation between OOD and ID logits and tries to device scoring techniques using solely ID logits or softmax output. The OOD detection works by telling as OOD anything that is not ID. The earliest work on this front assumes clustering of ID logits into a multimodal Gaussian distribution and then tries to utilize Mahalanobis distance Lee et al. [2018], Ren et al. [2021]. More advanced methods try to upgrade the Mahalonobis distance with geometric information using Fisher Information matrix Gomes et al. [2022] Other works, try to perform a data drive density estimation using energy-based models Liu et al. [2020]. Another promising research demonstrates the utility of enhanced Hopfield networks in amplifying the distinction between ID and OOD data Hofmann et al. [2024].

B Toy example

The training configuration for the model outlined in table 1 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.

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

C In distribution positioning in the logit space during training

Theorem 1. *In the training process of a deep learning classifier utilizing an activation function that suppresses negative values, the logit corresponding to the true class, (i.e., ID-in denoted by $\hat{L}(j)$), attain big positive magnitude ($\hat{L}(j) \rightarrow +\infty$). Simultaneously, the logits representing the incorrect classes, (i.e., ID-out denoted by $\hat{L}(i)$ for $i \neq j$), converge towards minimal magnitude values ($\hat{L}(i)_{i \neq j} \rightarrow 0$).*

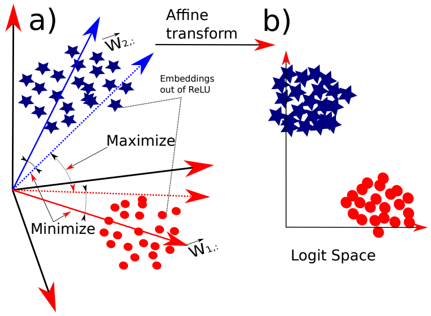


Figure 3: This toy example shows the separation of ID in a binary classification task. Figure a) contains the embeddings (E) rectified with a ReLU. Figure b) shows the linear separation of class-wise clustering of ID data logits $(\hat{L})$. The smaller the angle between $\vec{E}$ and $\vec{W}_{1,:}$, the higher the dot-product $\langle W_{1,i}, E_i \rangle$ Figure a); thus the more distanced from the center the ID logits are (Figure b). The bigger the angle between $(\vec{E})$ and $\vec{W}_{2,:}$, the higher the dot-product $\langle \vec{W}_{2,i}, \vec{E}_i \rangle$ (see, fig. 3 a), the more compact the ID logits are.

Proof. To establish the constraint towards zero for the logit cells not corresponding to the correct class (i.e., ID-out $\hat{L}(i)_{i \neq j} \rightarrow 0$), it is crucial to acknowledge that the predecessor latent space $(\hat{E}(i))$ is confined to positive values as the negative values are suppressed (see, fig. 3.a). The layer preceding the softmax constitutes a linear transformation of the data from high-dimensional embeddings $(\hat{E})$ to the logit space ($\hat{L} = \hat{E} \times W$, where $\times$ denotes matrix multiplication) with dimensions aligning with the number of specified classes (see, fig. 3.b). Since the optimizer seeks maximum response for the logit cell $\hat{L}[i]$ (i.e., ID-in), it aims to maximize the dot-product $\arg\max_{W[i,:]} \langle \hat{E}[:], W[i,:] \rangle^1$, $s.t: \hat{E}[:] \geq 0$.

Considering the embeddings $\hat{E}[:]$ and $W[i,:]$ as vectors in the vector space (see, fig. 3a), maximizing $\langle \vec{E}[:], \vec{W}[i,:] \rangle$ results in the minimization of the angle between $\vec{E}[:]$ and $\vec{W}[i,:]$ $\big(i.e., \min \angle(\vec{W}[i,:], \vec{E}[:])\big)$ while ensuring the former always remains in the positive regions. The optimization aims to maintain the direction of the vector $\vec{W}[i,:]$ akin to the cluster of vectors $\vec{E}[:]$, specifically within the positive regions (see, fig. 3.a).

Moreover, the optimization aims to achieve a minimum response for every other logit cell $\hat{L}[j \neq i]$ that does not correspond to the correct class, expressed as $\arg\min_{W[j \neq i,:]} \langle \hat{E}[:], W[j \neq i,:] \rangle$, subject to the constraint $\hat{E}[:] \geq 0$. In essence, it seeks to maximize the angle between $\vec{W}[j \neq i,:]$ and the cluster of vector data $\vec{E}[:]$ $\big(i.e., \max \angle(\vec{W}[j \neq i,:], \vec{E}[:])\big)$ (see, fig. 3.a).

Thus, the clusters associated with different classes endeavour to attain maximum angular separation from one another, leading the parameter vectors $\vec{W}[i,:]$ to align accordingly. Given that all vectors $\vec{E}[:]$ are angularly separated within the positive region, the maximum angle between these two vectors closely approaches perpendicularity (see, Lemma 1). Consequently, the minimized logit values $\big(\arg\min(\vec{W}[j \neq i,:], \vec{E}[:]) \approx 0\big)$ would asymptotically approach zero during training.

Consequently, the asymptotic behavior of the data configuration in the logit space compels the data points to form compact clusters far from the center of the space, corresponding to their respective

[1] $\langle , \rangle$ indicates the dot-product

classes. This process leads to the minimization of interclass distances and the maximization of intraclass distances.

□

Lemma 1. *In the positive region of a high-dimensional space, the maximum angle that two vectors can attain is perpendicular.*

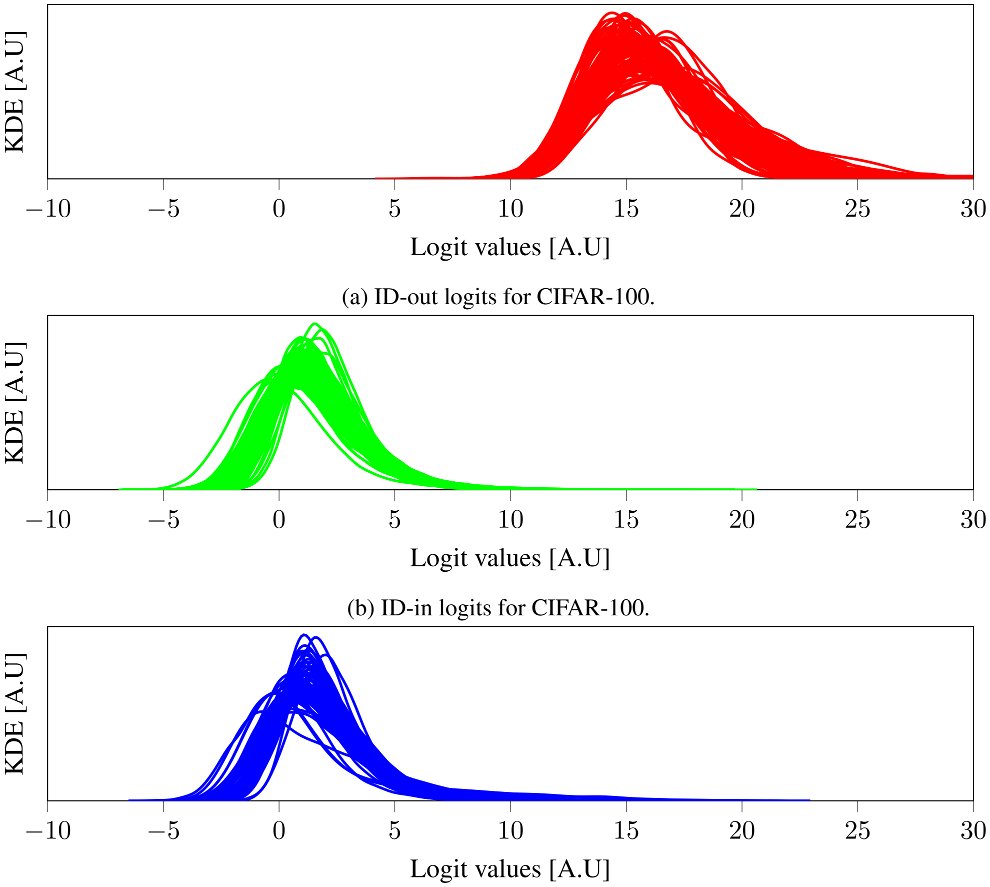
Proof. One way to establish this lemma involves employing the concept of cosine similarity. Let us consider two arbitrary vectors in an N-dimensional space, denoted as X and Y, where $X, Y \in \mathbb{R}^N$. The cosine similarity between these vectors is defined as:

$$\cos_{\text{sim}}(X, Y) = \frac{\sum_{i=1}^{N} X_i Y_i}{\|X\| \|Y\|} \tag{1}$$

Given that both vectors reside in the positive region of the vector space, meaning that each component of the vectors satisfies $X_i \geq 0, Y_i \geq 0 \forall i \in [1, .., N]$, it is evident that any two vectors in the positive region cannot yield a negative value for the cosine similarity. This is because the numerator, representing the dot-product of the vectors, comprises products of non-negative components. Consequently, the numerator cannot be negative. Therefore, the minimum value that $\cos_{\text{sim}}(X, Y)$ can attain is zero, corresponding to perpendicular vectors. □

D 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 decimated every quarter of epochs. Within each quarter, the learning rate is scheduled using a 1cycle 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. 4a and 4b 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. 4c.



(a) ID-out logits for CIFAR-100.

(b) ID-in logits for CIFAR-100.

(c) ODD logits for CIFAR-10.

Figure 4: KDE response CIFAR-100 (ID) vs CIFAR-10 (OOD) while using Resnet-9 with ReLU activation function.

E 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 demonstrated this impact by utilizing a selection of activation functions known for their inherent suppression of negative values, including Celu Barron [2017], Elu Clevert et al. [2016], Gelu Hendrycks and Gimpel [2023], Selu Klambauer et al. [2017], Silu Elfwing et al. [2017], Relu Hein et al. [2019], Leaky-Relu Maas et al. [2013], and Mish Misra [2020].

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

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

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

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

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

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

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

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

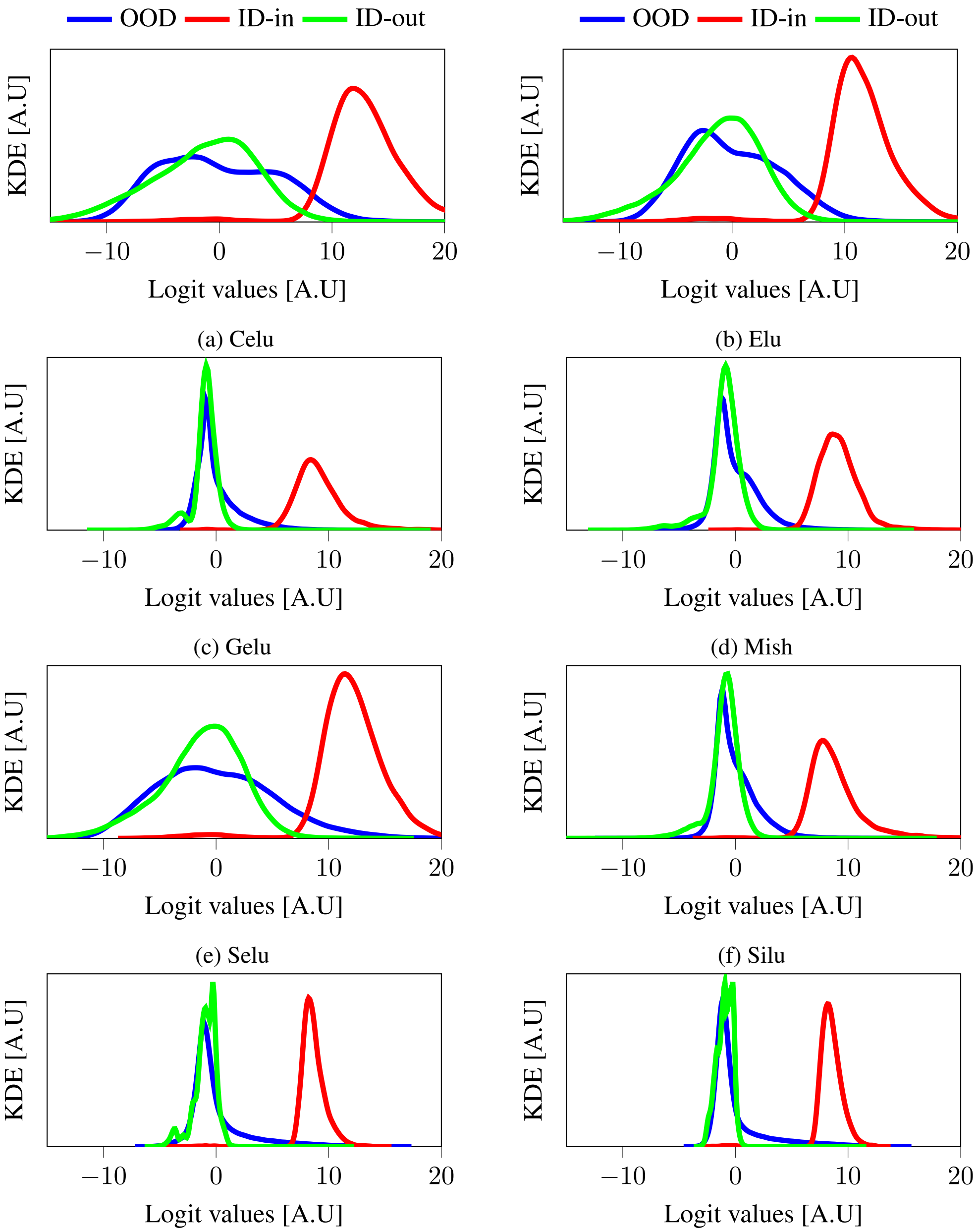
$$\text{Leaky-Relu:} f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}$$

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

A ResNet-34 model was trained on the SVHN dataset Goodfellow et al. [2014], (i.e., ID data), utilizing each activation function. 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 $5 * 10^{-4}$. 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. 5). On the other hand, ID-out and OOD logits are predominantly centralized around zero (see fig. 5). Consequently, despite the application of varying non-linearities, the relative configuration of ID and OOD logits remains similar. In addition to the density plots depicted in fig. 5, which show the distribution of aggregated ID-in, ID-out, and OOD across all logit cells, figs. 6 to 13 provides a detailed visualization of density plots on individual logit cells using various activation functions.



(a) Celu

(b) Elu

(c) Gelu

(d) Mish

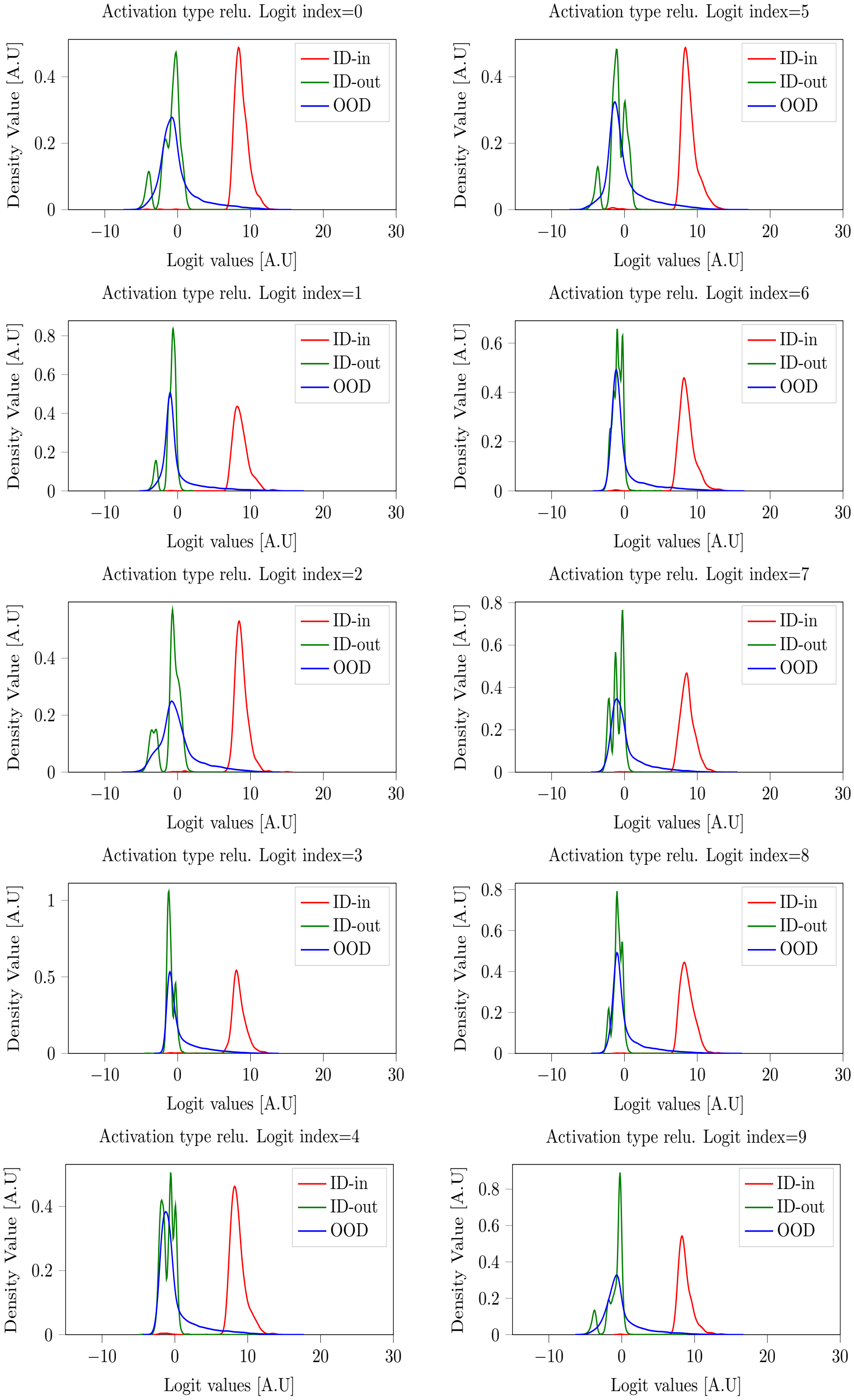
(e) Selu

(f) Silu

(g) Relu

(h) Leaky Relu

Figure 5: 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.


Activation type relu. Logit index=0

Activation type relu. Logit index=5

Activation type relu. Logit index=1

Activation type relu. Logit index=6

Activation type relu. Logit index=2

Activation type relu. Logit index=7

Activation type relu. Logit index=3

Activation type relu. Logit index=8

Activation type relu. Logit index=4

Activation type relu. Logit index=9

Figure 6: Densities over each logit cell from a Resnet-34 classifier with Relu activation.

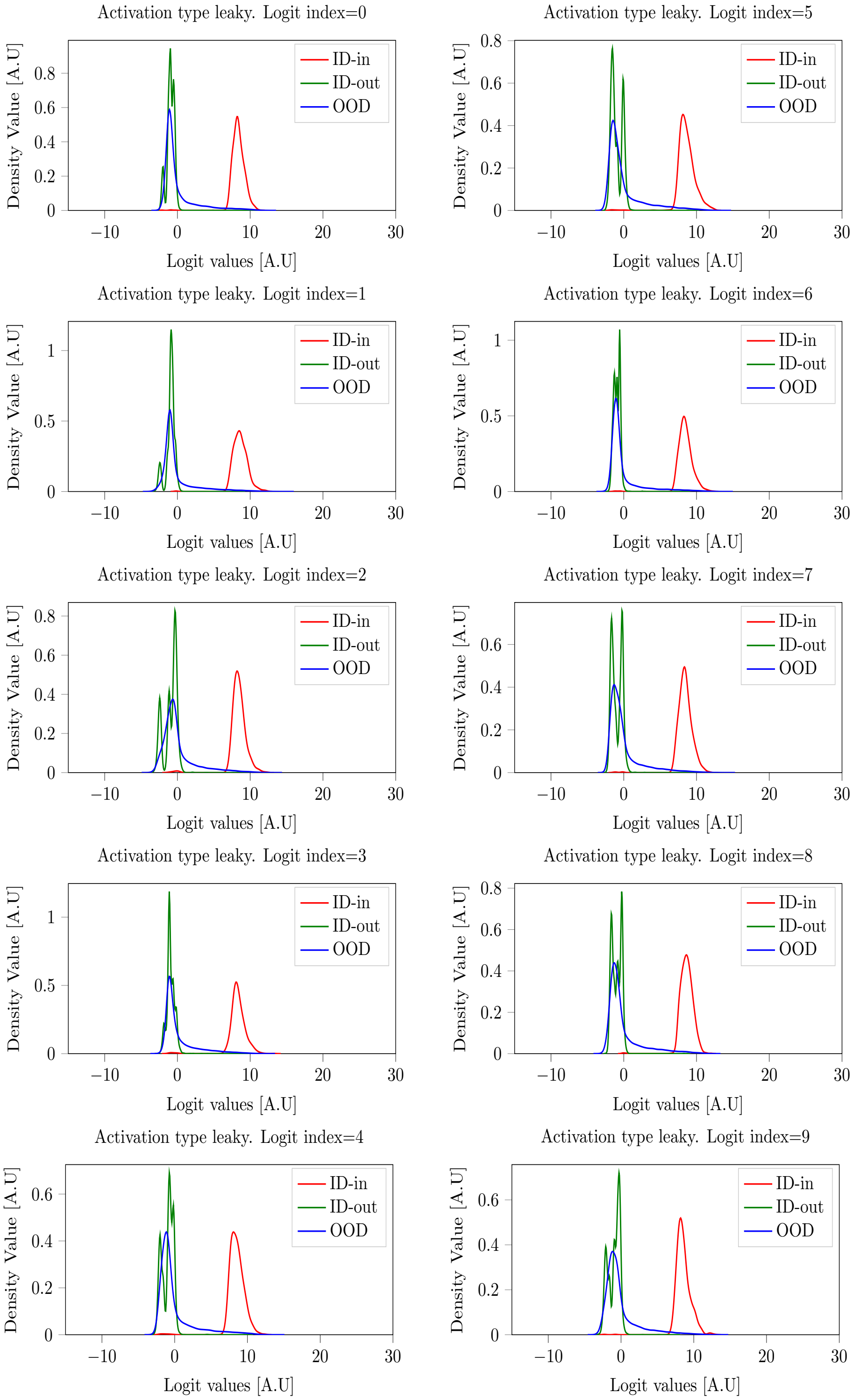


Figure 7: Densities over each logit cell from a Resnet-34 classifier with Leaky Relu activation.

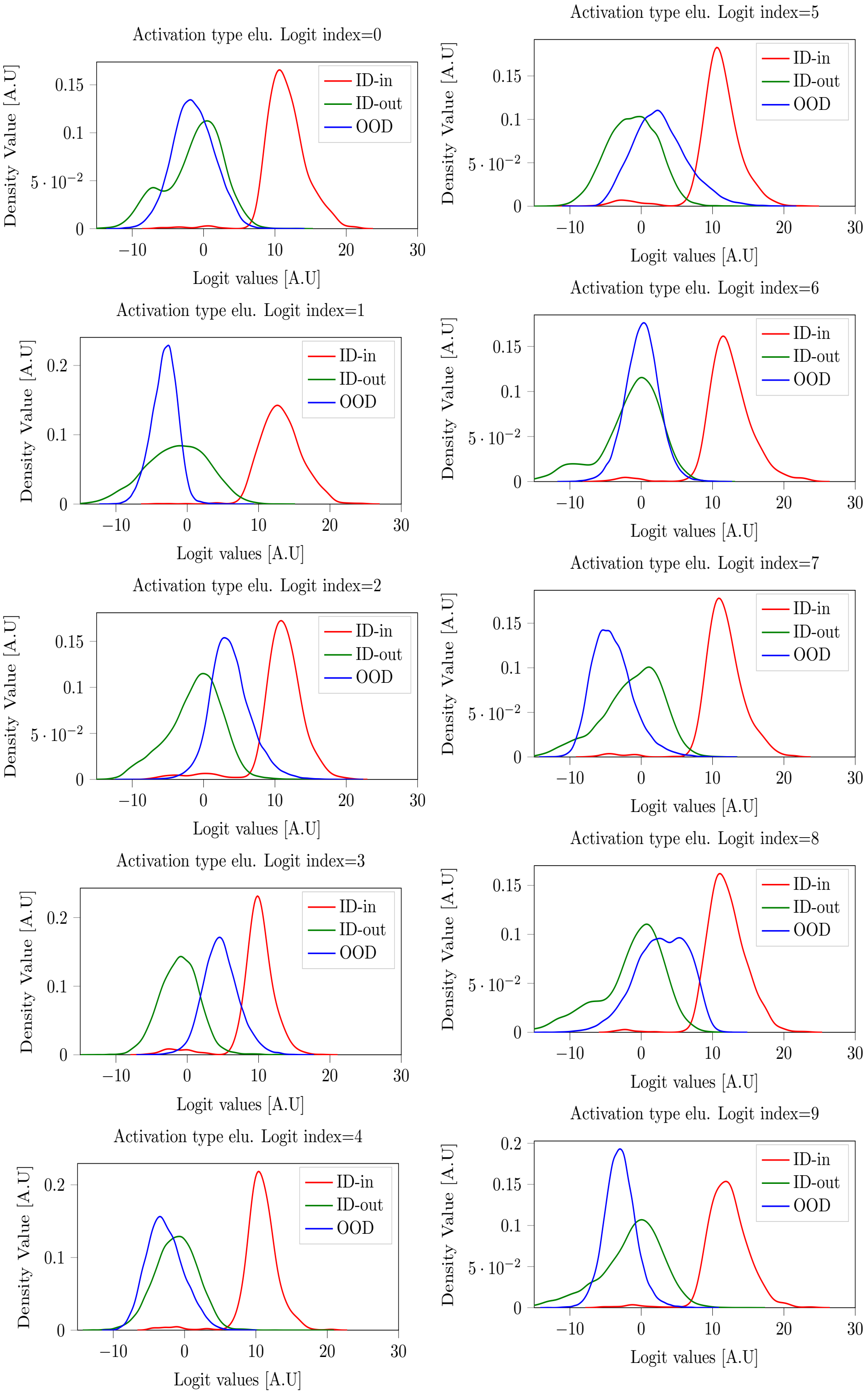


Figure 8: Densities over each logit cell from a Resnet-34 classifier with Elu activation.

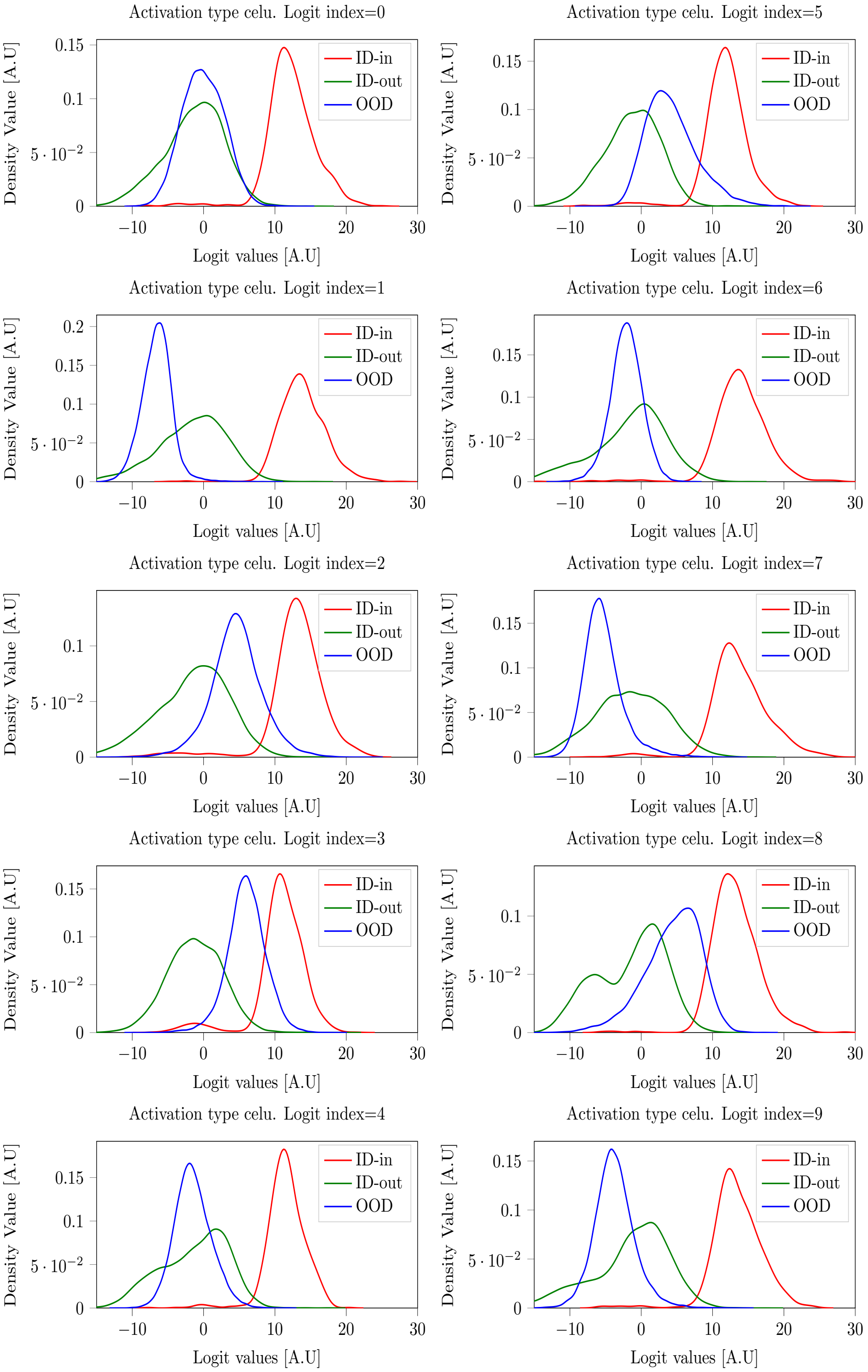
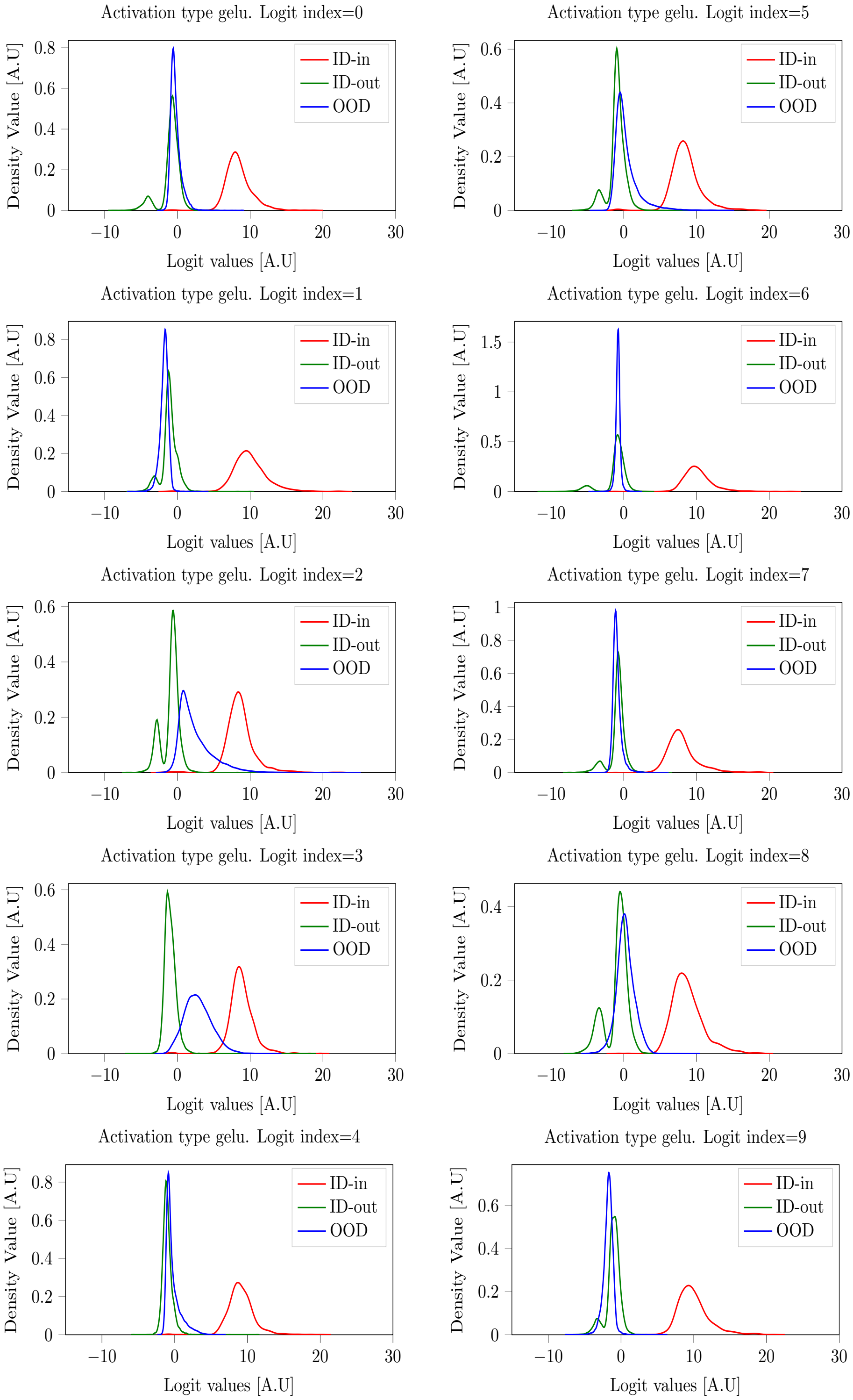


Figure 9: Densities over each logit cell from a Resnet-34 classifier with Celu activation.



Activation type gelu. Logit index=0

Activation type gelu. Logit index=5

Activation type gelu. Logit index=1

Activation type gelu. Logit index=6

Activation type gelu. Logit index=2

Activation type gelu. Logit index=7

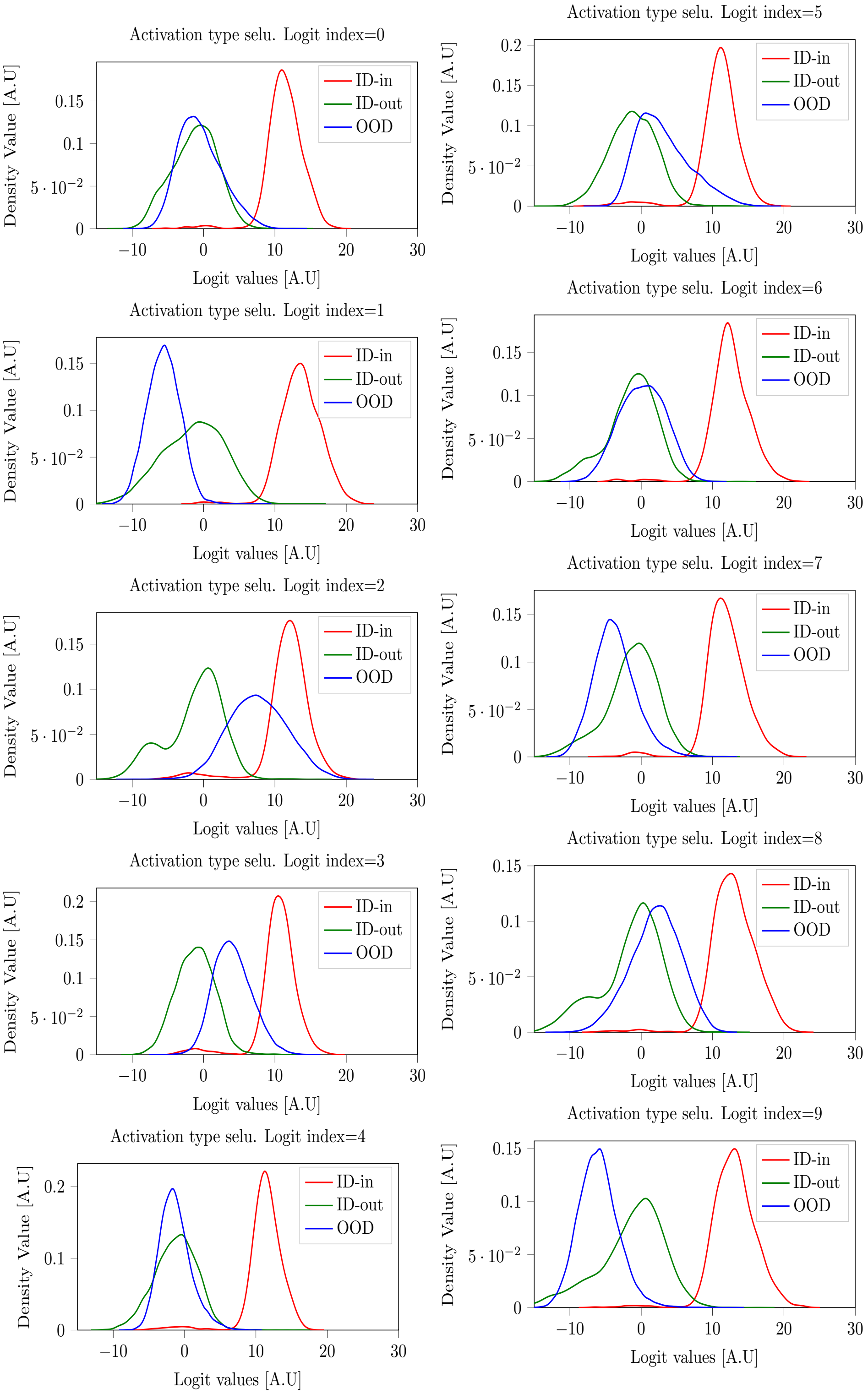
Activation type gelu. Logit index=3

Activation type gelu. Logit index=8

Activation type gelu. Logit index=4

Activation type gelu. Logit index=9

Figure 10: Densities over each logit cell from a Resnet-34 classifier with Gelu activation.



Activation type selu. Logit index=0

Activation type selu. Logit index=1

Activation type selu. Logit index=2

Activation type selu. Logit index=3

Activation type selu. Logit index=4

Activation type selu. Logit index=5

Activation type selu. Logit index=6

Activation type selu. Logit index=7

Activation type selu. Logit index=8

Activation type selu. Logit index=9

Figure 11: Densities over each logit cell from a Resnet-34 classifier with Selu activation.

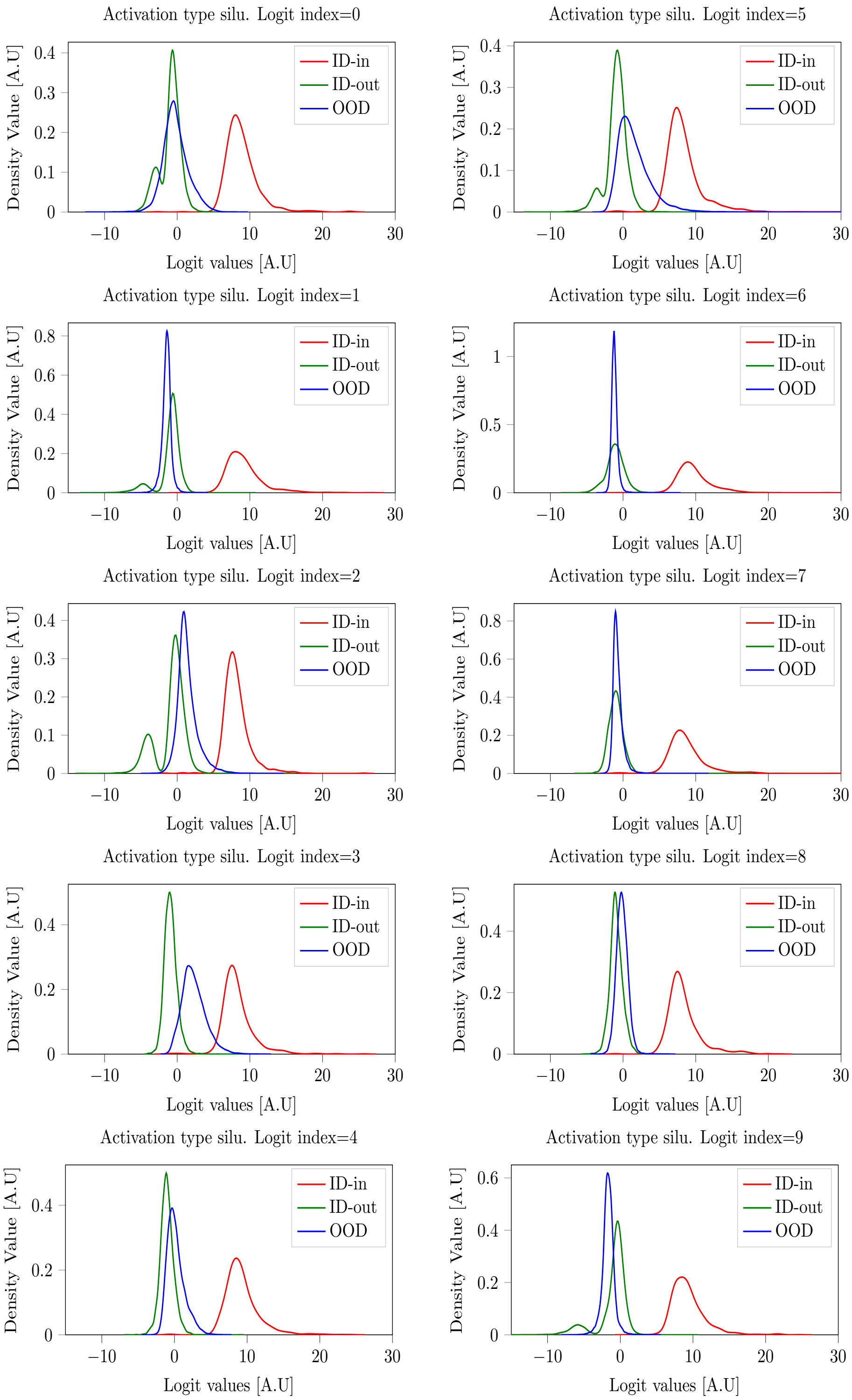


Figure 12: Densities over each logit cell from a Resnet-34 classifier with Silu activation.

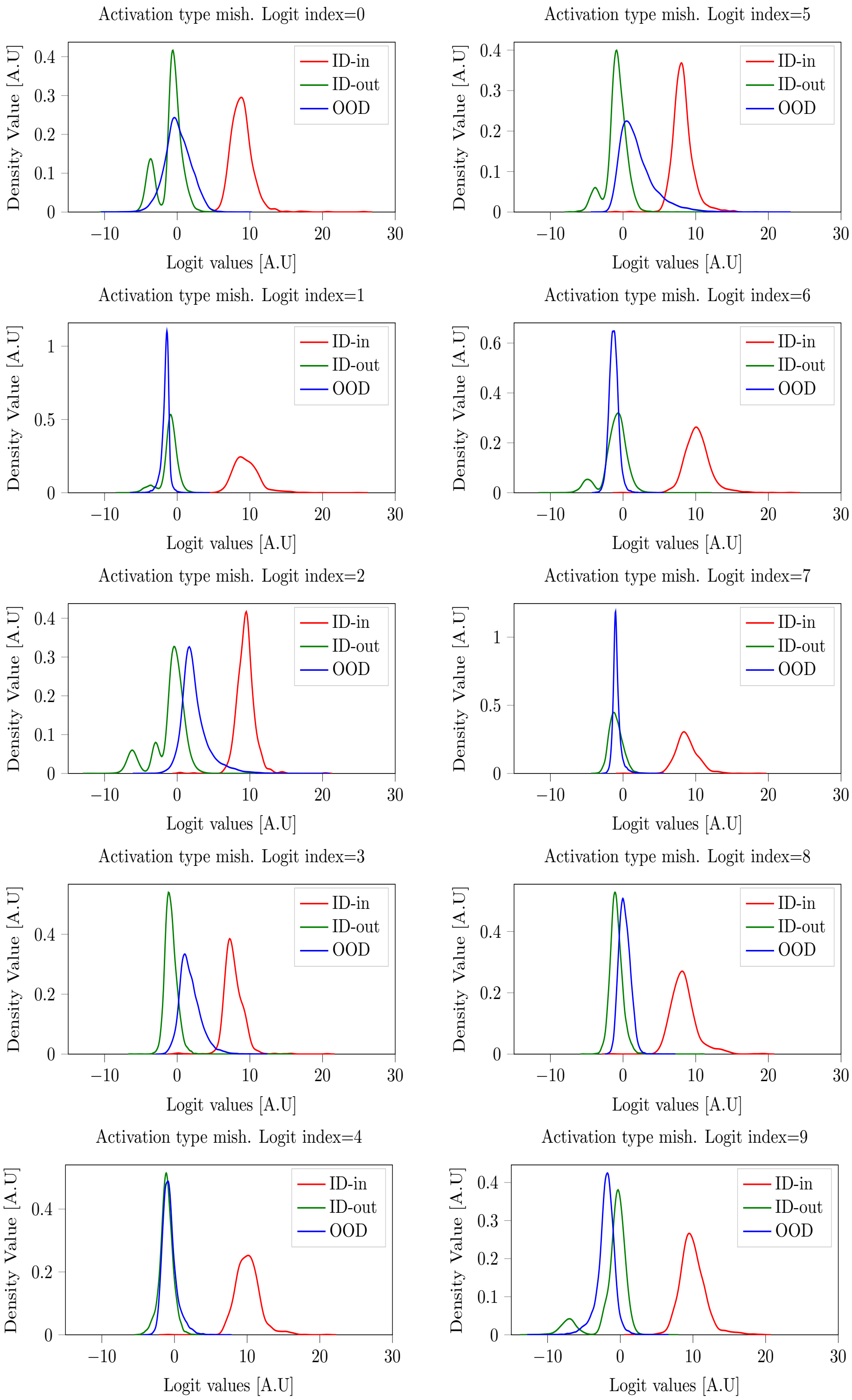


Figure 13: Densities over each logit cell from a Resnet-34 classifier with Mish activation.

F Experiments on different classifiers

The analysis of ID and OOD logits has been expanded across various DL classifier models. Our study examines various iterations of DenseNet Huang et al. [2018], specifically versions 121, 161, 169, and 201, as well as ResNet He et al. [2015b], encompassing versions 18, 34, 50, 101, and 152. Furthermore, the utilized experimental dataset comprises $\{D\} = \{$SVHN, CIFAR-100, CIFAR-10, Tiny ImageNet Deng et al. [2009], iSUN Xu et al. [2015], LSUN Yu et al. [2016]$\}$.

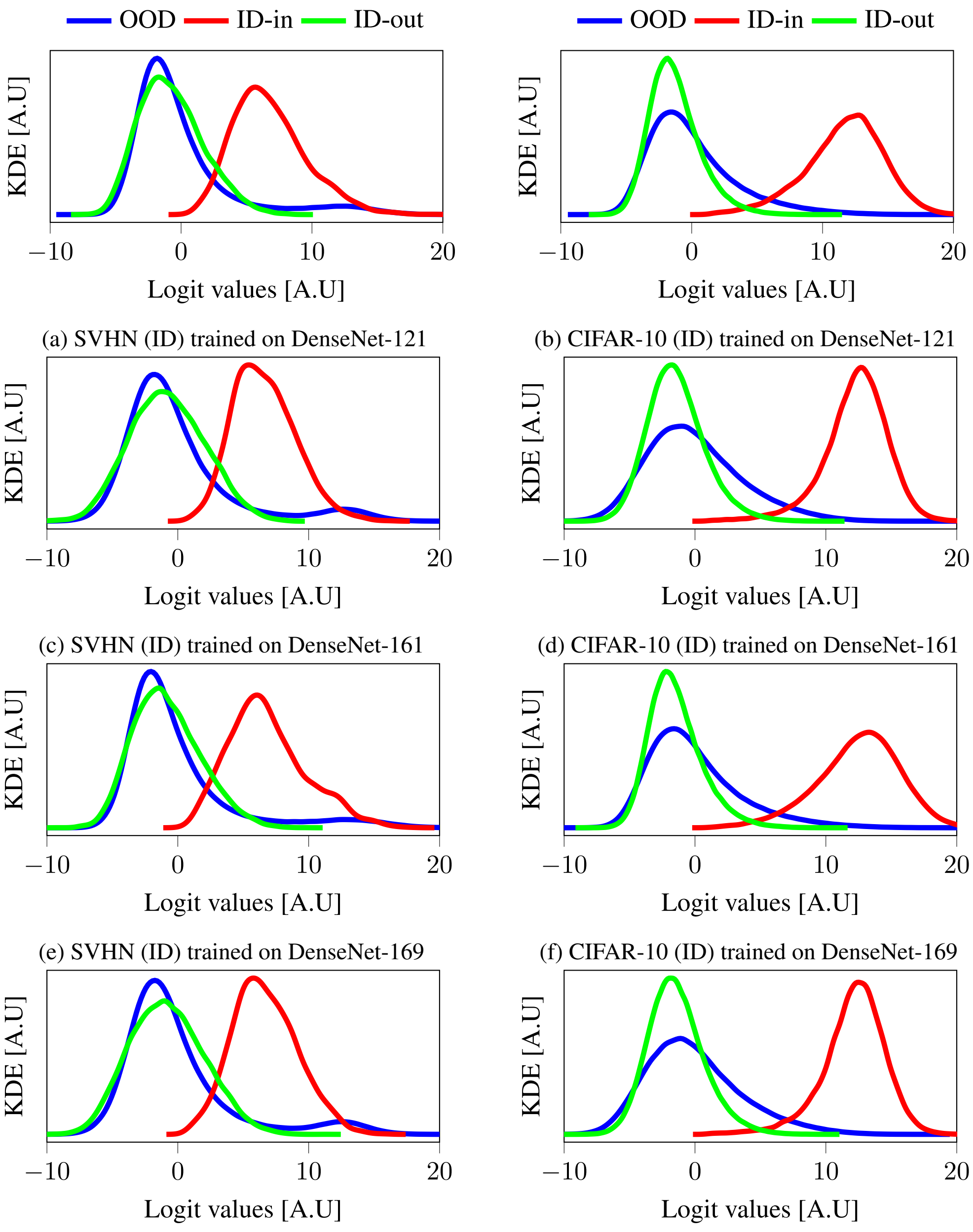
Each model undergoes separate training on CIFAR-10 and SVHN as ID datasets.

Densenet and ResNet models are trained using 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 $5 * 10^{-4}$. A batch size of 256 is applied for both test and train data while the number of epochs is 200. ReLU is utlized as activation function for every layer. No regularization is applied to the training process, while the training data are augmented with random flipping and cropping.

When CIFAR-10 is utilized as ID the remaining datasets are employed as OOD data, specifically$\{D\}$ without CIFAR-10 (i.e, $\{D\}/$CIFAR-10) is utilzed as OOD. Similarly, when SVHN is utilized as ID the remaining datasets are employed as OOD data, specifically$\{D\}$ without SVHN (i.e, $\{D\}/$SVHN) is utilzed as OOD.

Observations indicate that the ID-in logits consistently tend toward higher positive values across various versions of DenseNet (see fig. 14) and ResNet (see fig. 15). Contrarily, ID-out and OOD logits tend to be concentrated around zero (as shown in fig. 5).

Notice that the density plots in figs. 14 and 15 demonstrate the distribution of the ID-in, ID-out, and OOD for all logit cells aggregated together. For thorough visual representations on a per-logit-cell basis, across all different versions of DensetNet and ResNet see Figures 16 to 33.



(a) SVHN (ID) trained on DenseNet-121

(b) CIFAR-10 (ID) trained on DenseNet-121

(c) SVHN (ID) trained on DenseNet-161

(d) CIFAR-10 (ID) trained on DenseNet-161

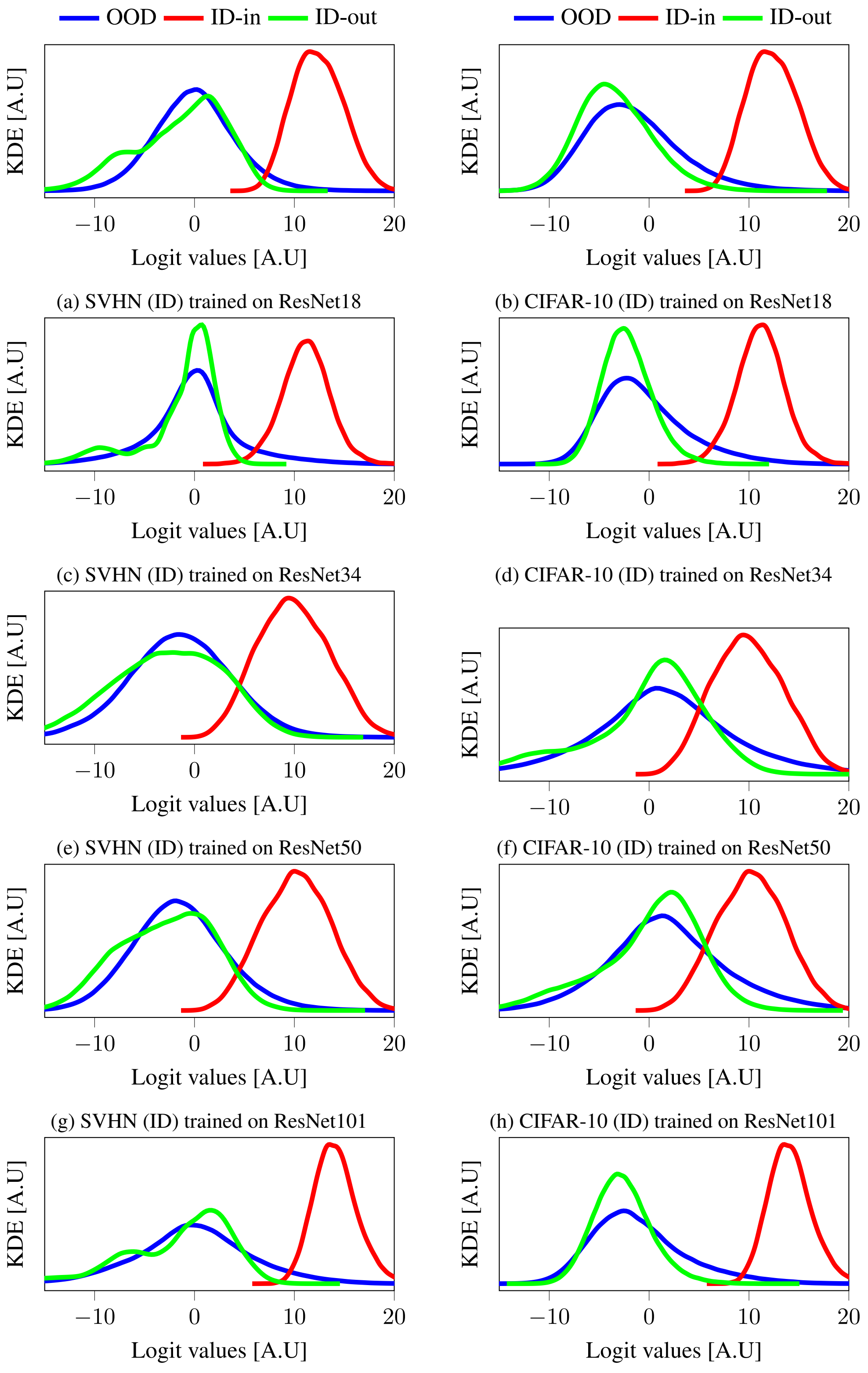
(e) SVHN (ID) trained on DenseNet-169

(f) CIFAR-10 (ID) trained on DenseNet-169

(g) SVHN (ID) trained on DenseNet-201

(h) CIFAR-10 (ID) trained on DenseNet-201

Figure 14: Logit densities across various DenseNet architectures trained on SVHN and CIFAR-10.



(a) SVHN (ID) trained on ResNet18

(b) CIFAR-10 (ID) trained on ResNet18

(c) SVHN (ID) trained on ResNet34

(d) CIFAR-10 (ID) trained on ResNet34

(e) SVHN (ID) trained on ResNet50

(f) CIFAR-10 (ID) trained on ResNet50

(g) SVHN (ID) trained on ResNet101

(h) CIFAR-10 (ID) trained on ResNet101

(i) SVHN (ID) trained on ResNet152

(j) CIFAR-10 (ID) trained on ResNet152

Figure 15: Logit densities across various ResNet architectures trained on SVHN and CIFAR-10.

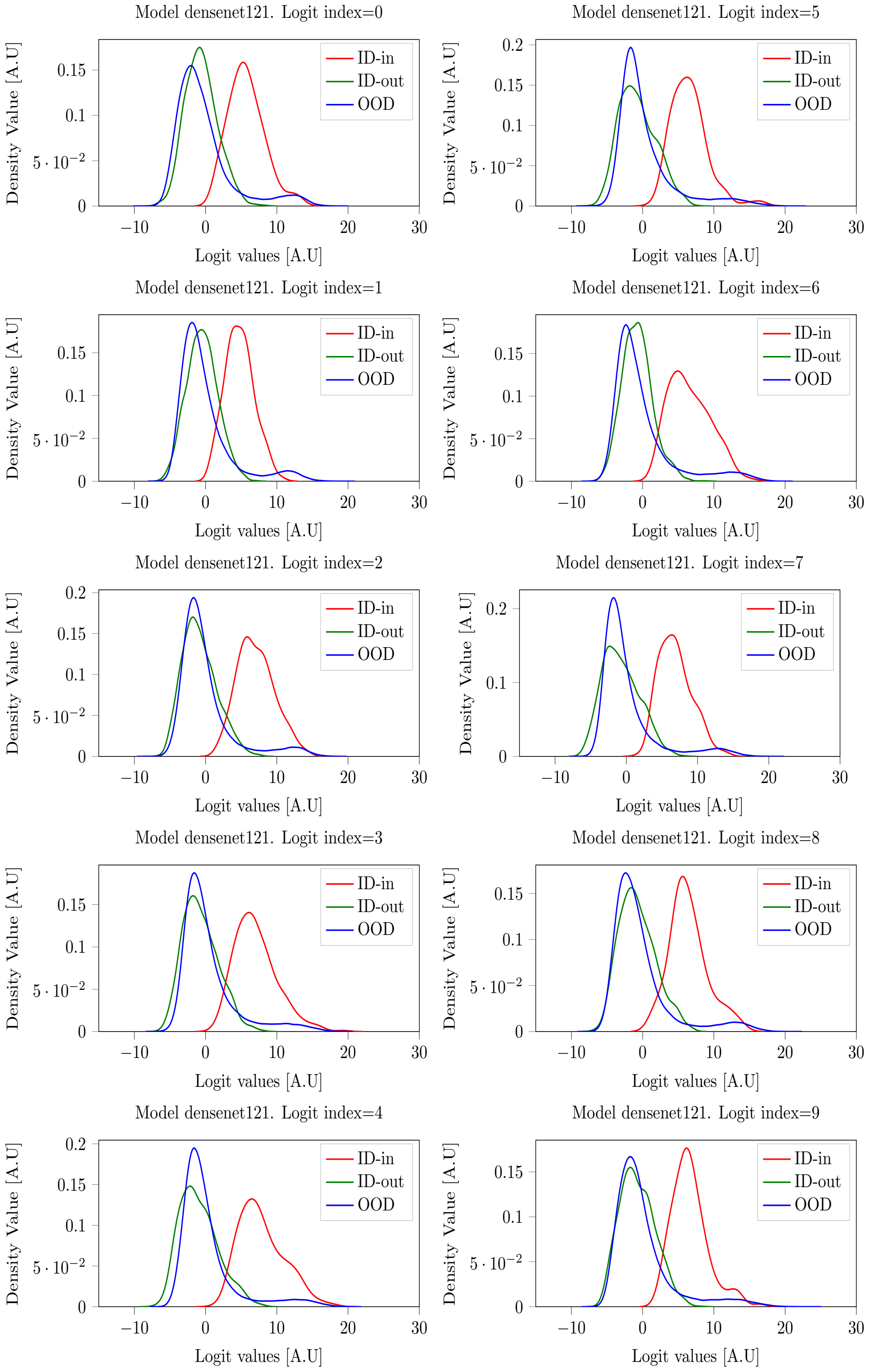


Figure 16: Logit cell densities for SVHN as ID with Densenet121.

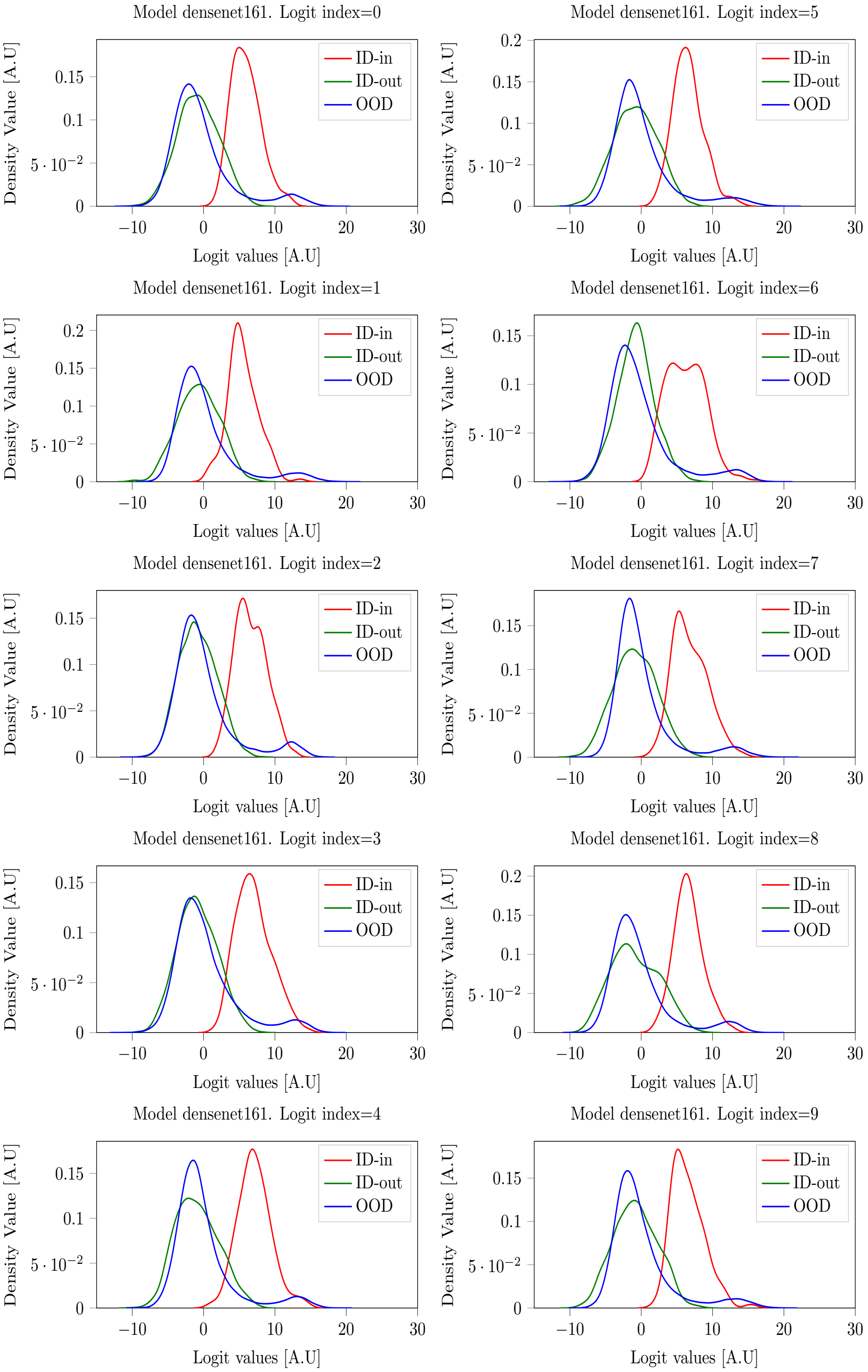


Figure 17: Logit cell densities for SVHN as ID with Densenet161.

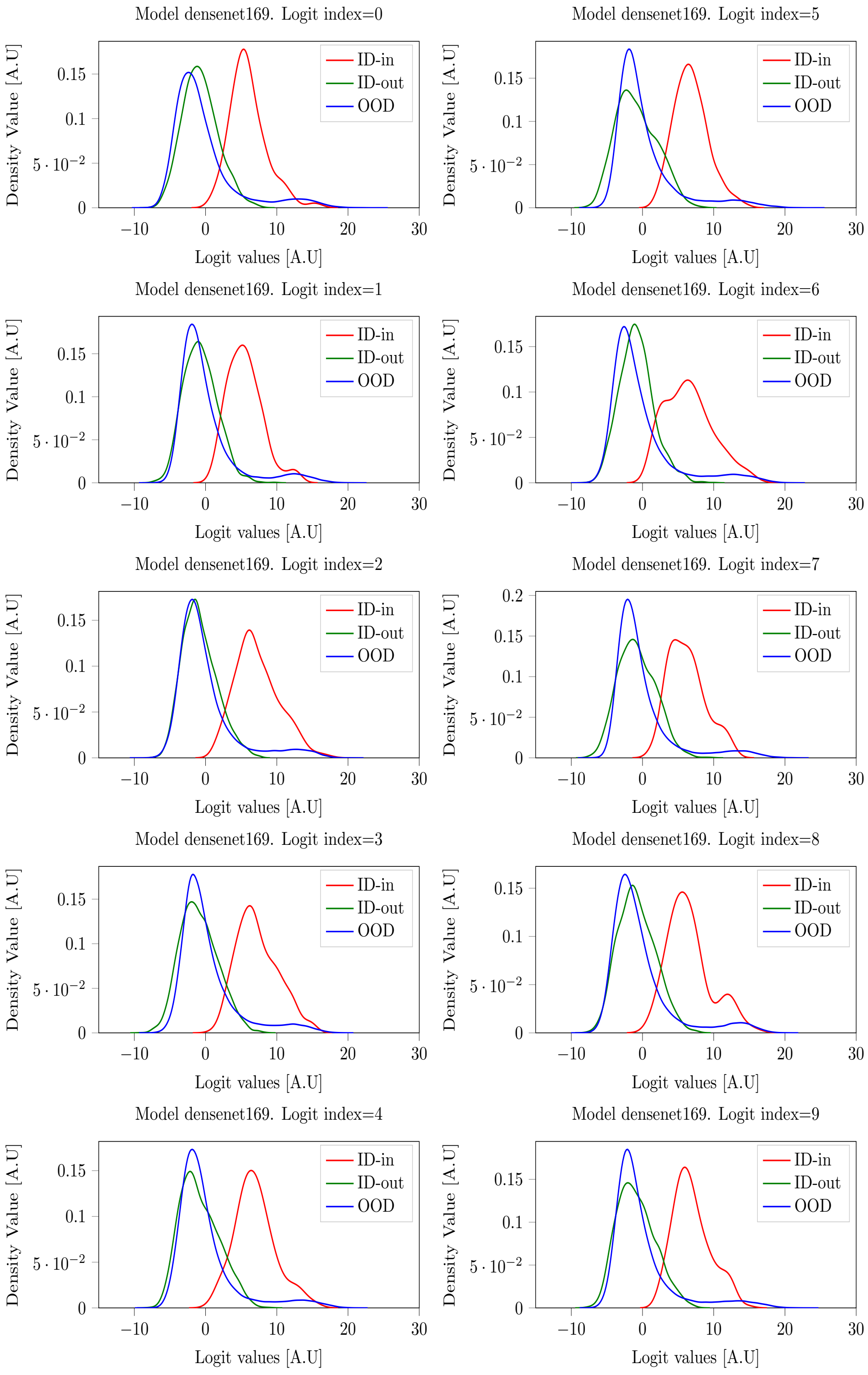


Figure 18: Logit cell densities for SVHN as ID with Densenet169.

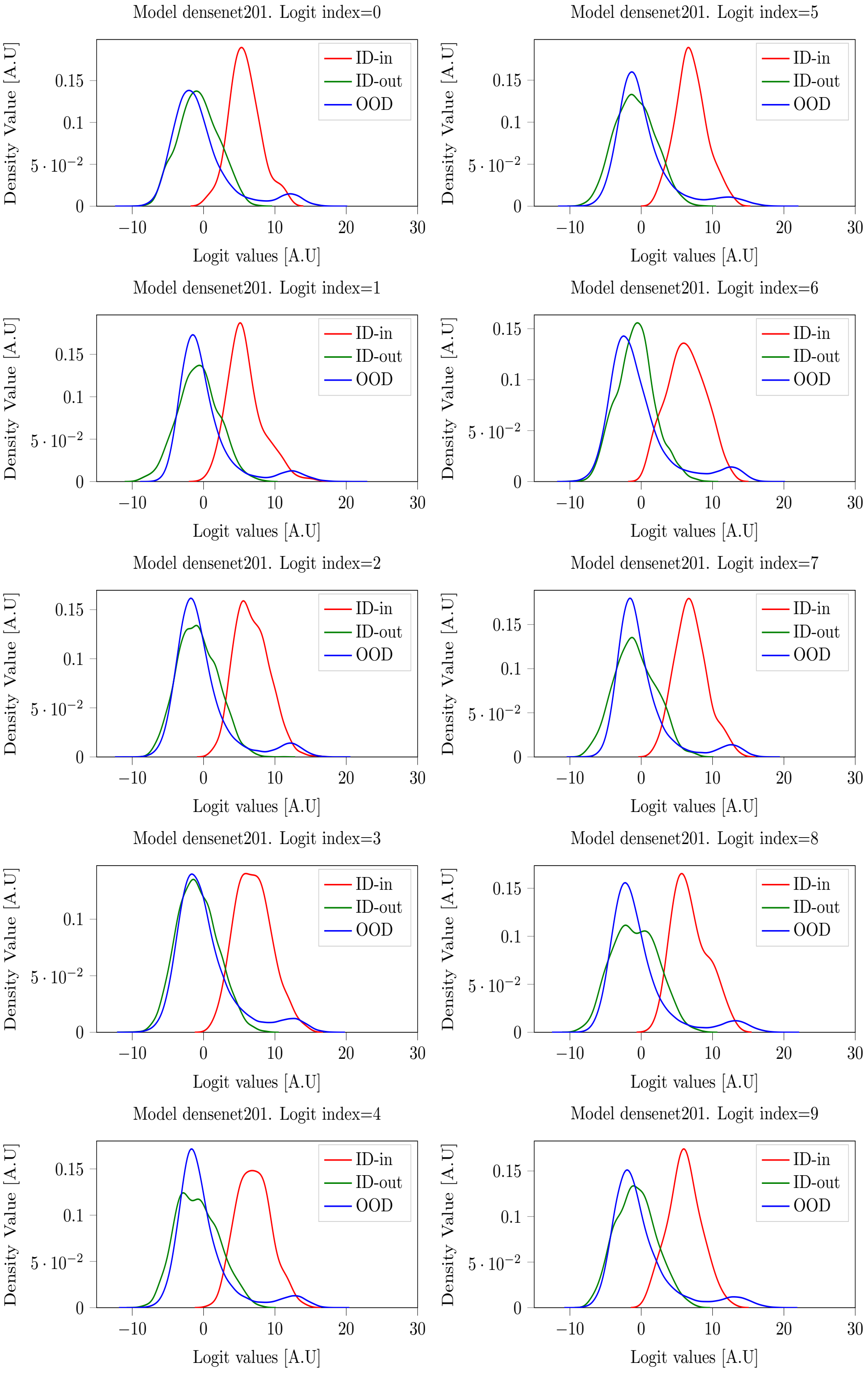


Figure 19: Logit cell densities for SVHN as ID with Densenet201.

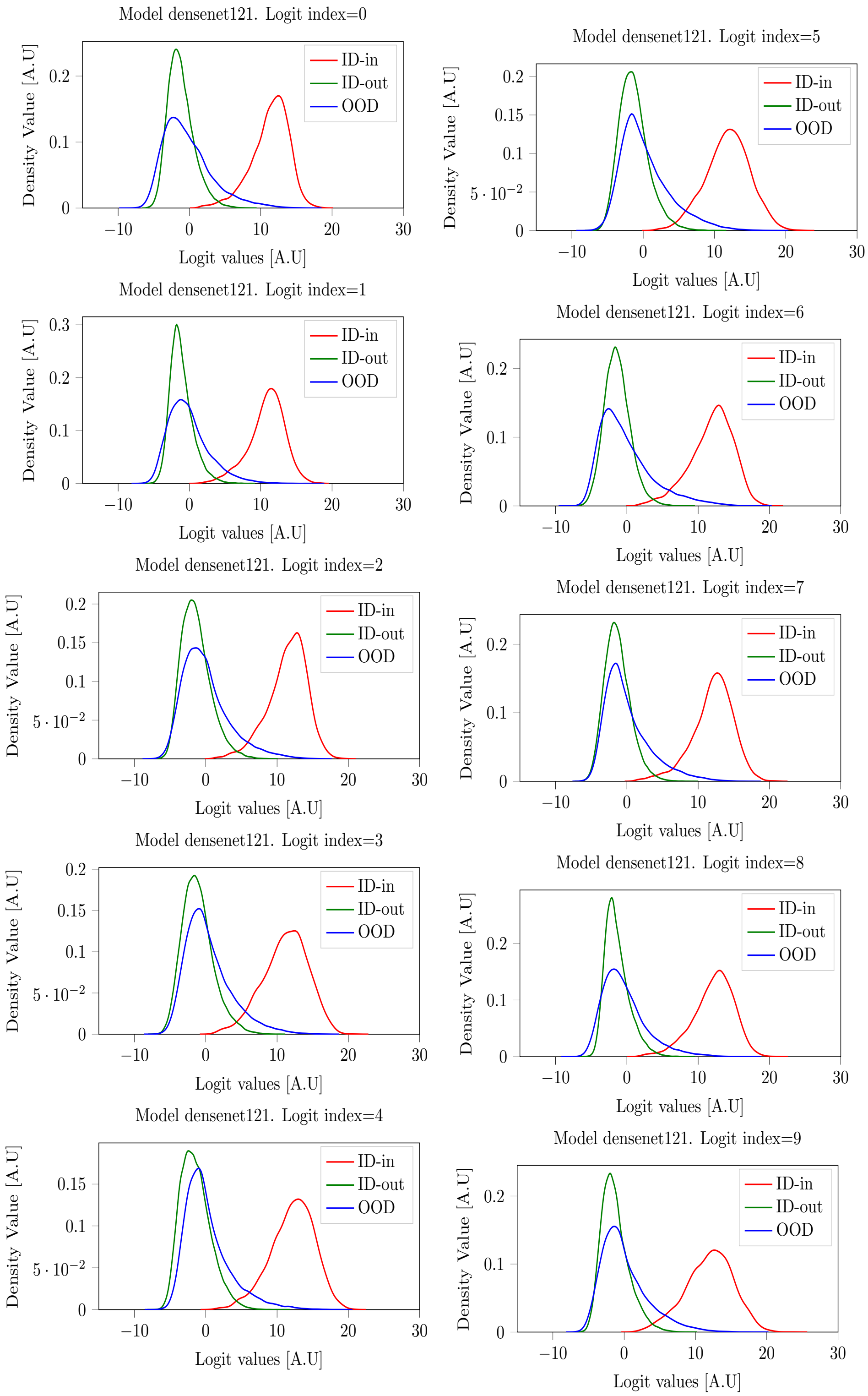


Figure 20: Logit cell densities for CIFAR-10 as ID with Densenet121.

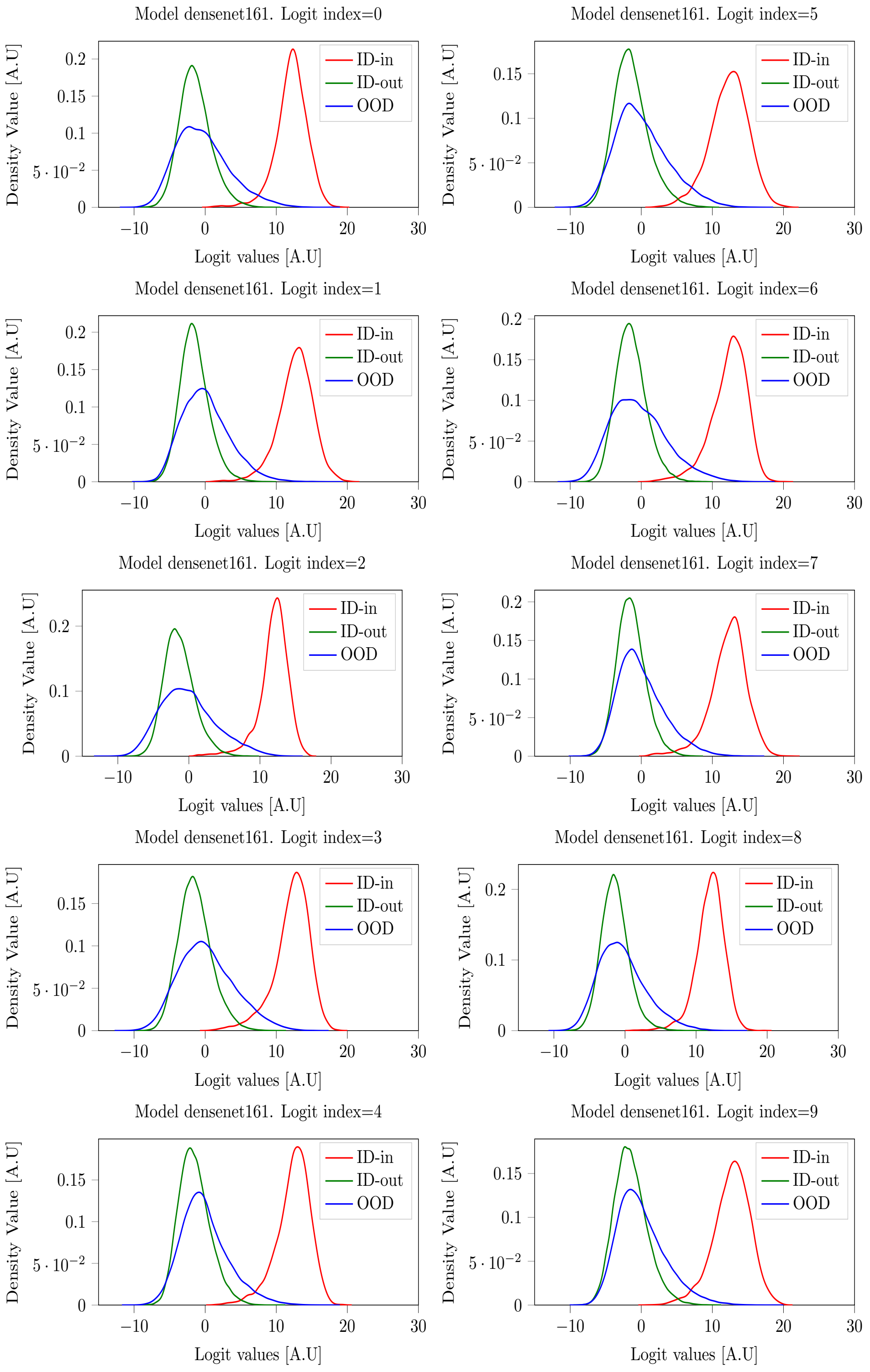


Figure 21: Logit cell densities for CIFAR-10 as ID with Densenet161.

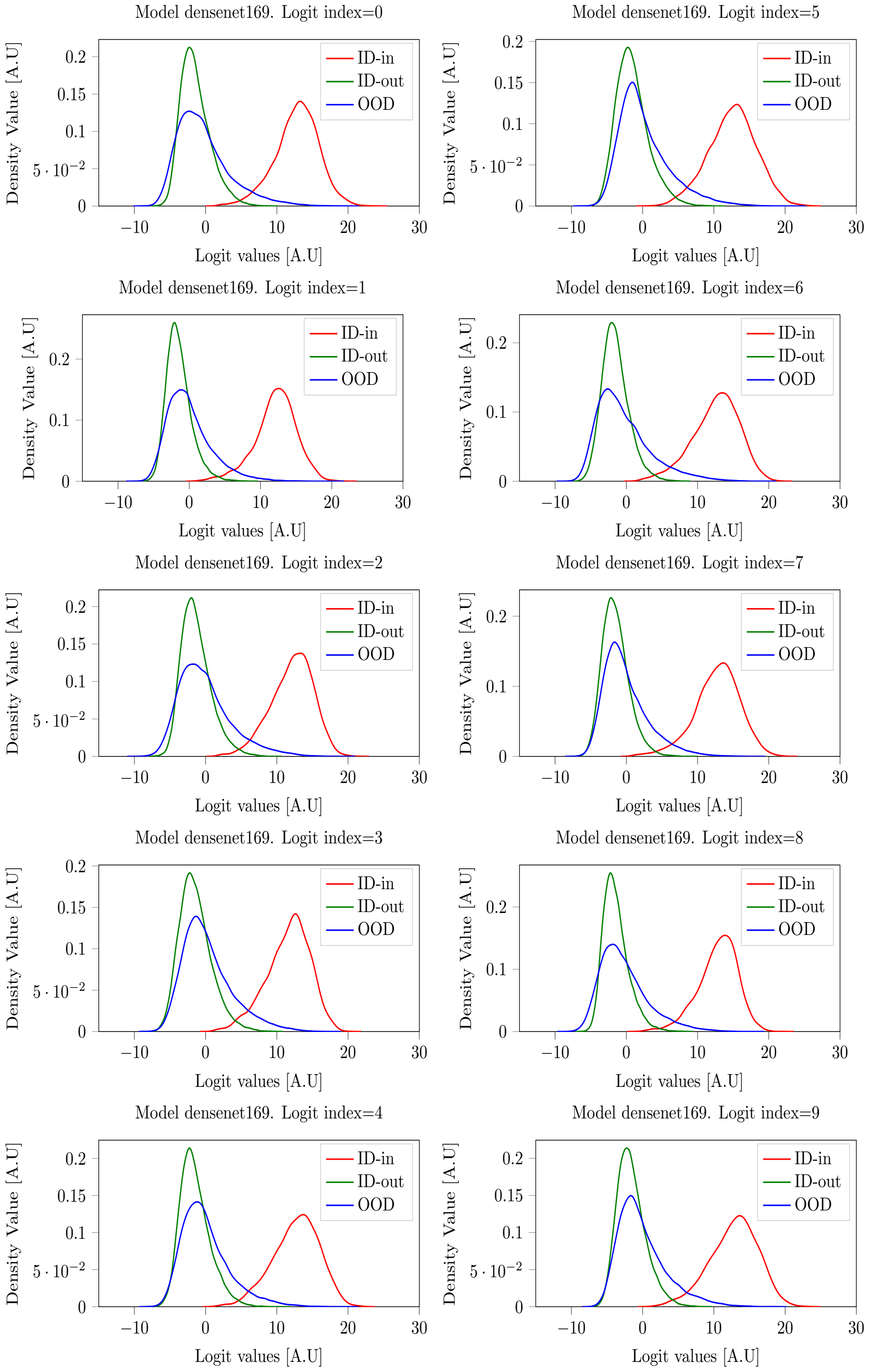
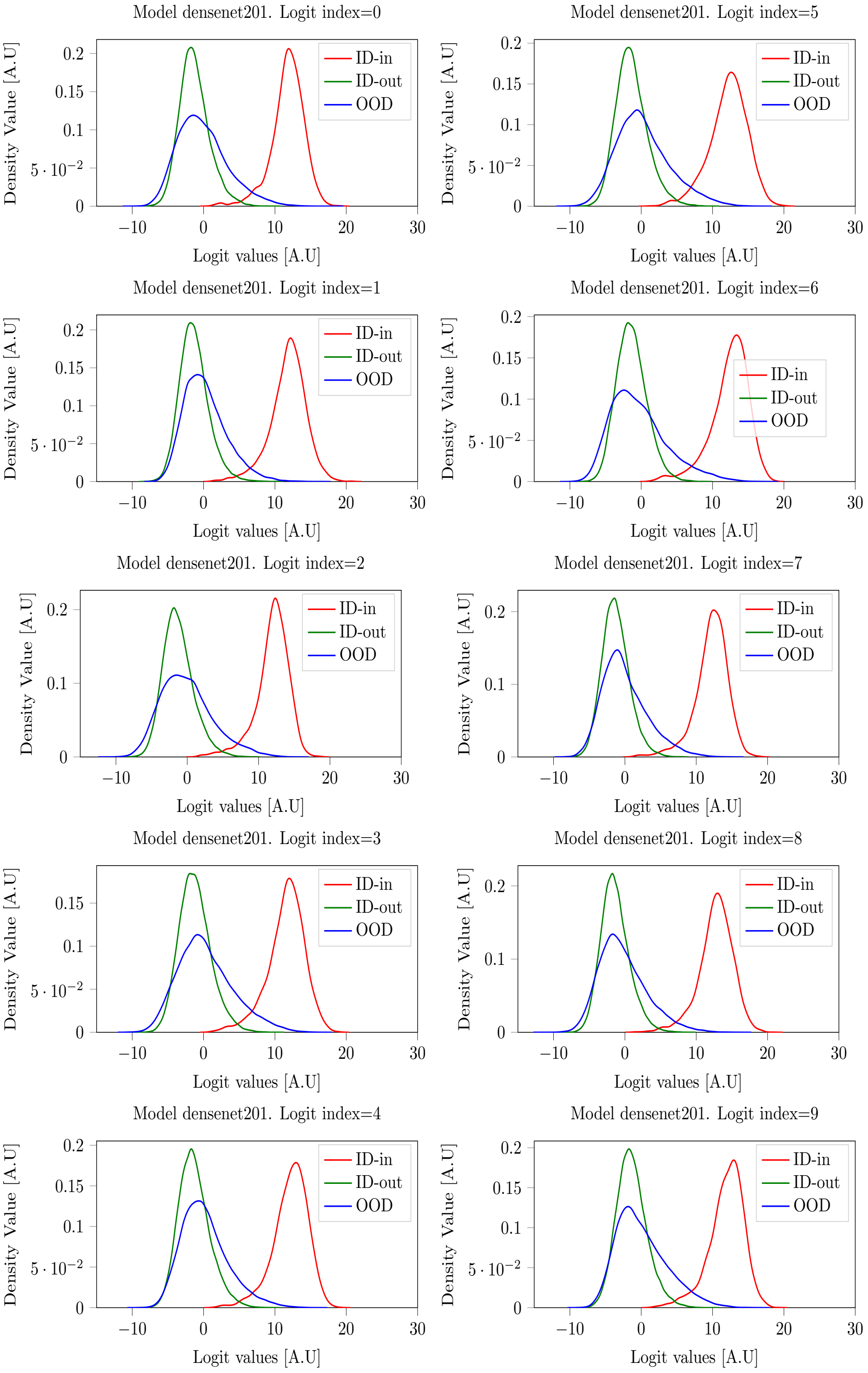


Figure 22: Logit cell densities for CIFAR-10 as ID with Densenet169.


Model densenet201. Logit index=0

Model densenet201. Logit index=5

Model densenet201. Logit index=1

Model densenet201. Logit index=6

Model densenet201. Logit index=2

Model densenet201. Logit index=7

Model densenet201. Logit index=3

Model densenet201. Logit index=8

Model densenet201. Logit index=4

Model densenet201. Logit index=9

Figure 23: Logit cell densities for CIFAR-10 as ID with Densenet201.

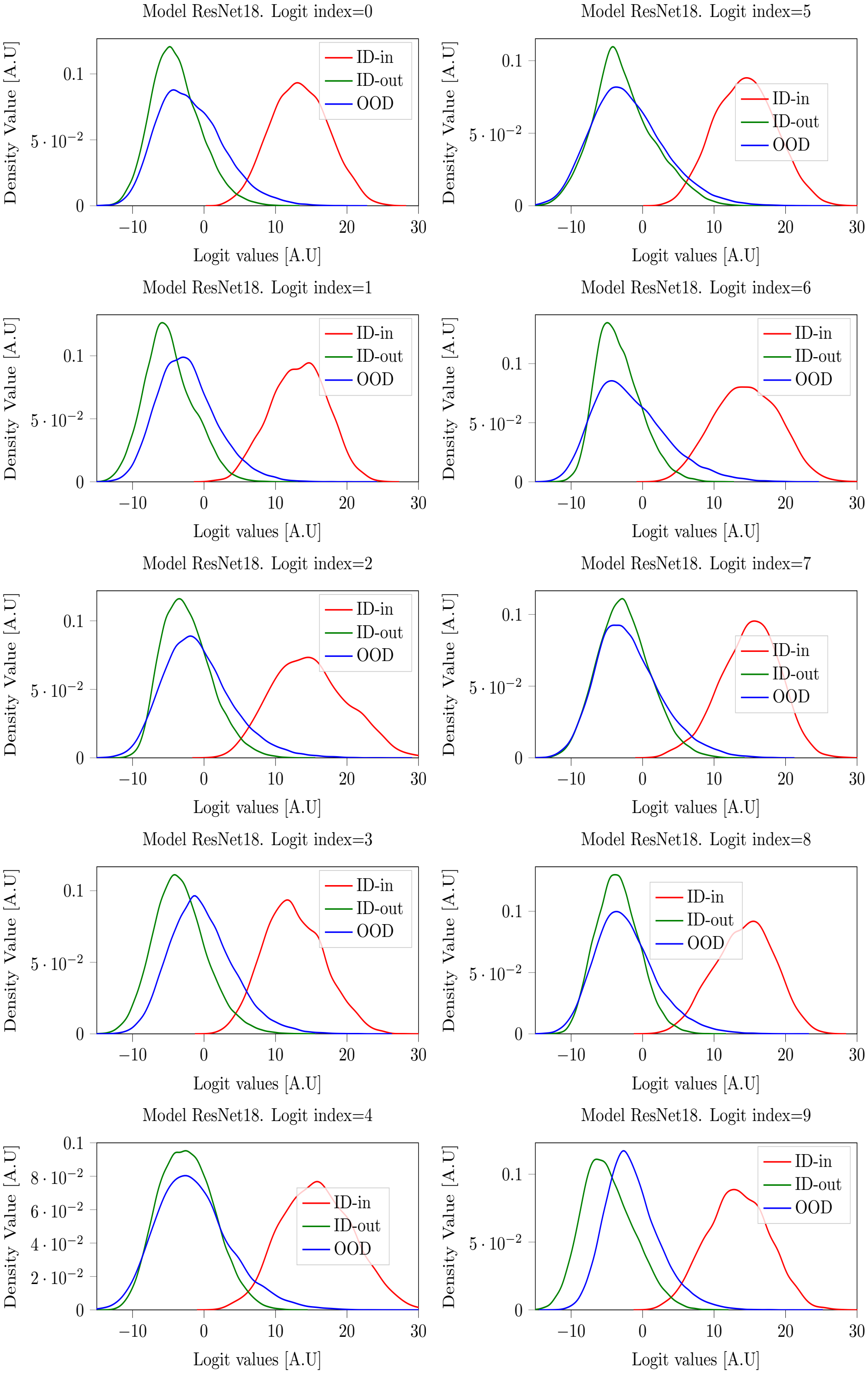


Figure 24: Logit cell densities for CIFAR-10 as ID with ResNet18.

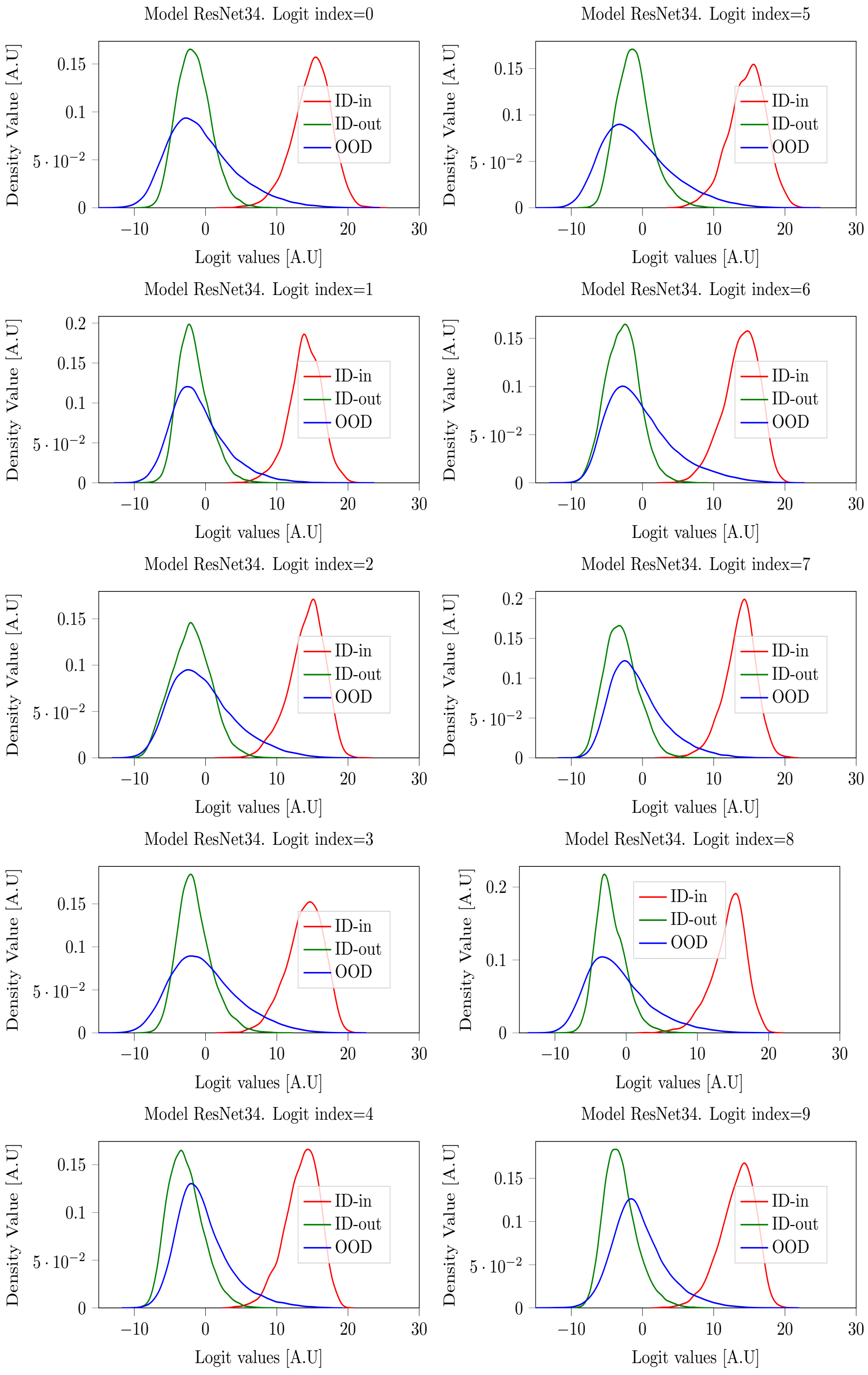


Figure 25: Logit cell densities for CIFAR-10 as ID with ResNet34.

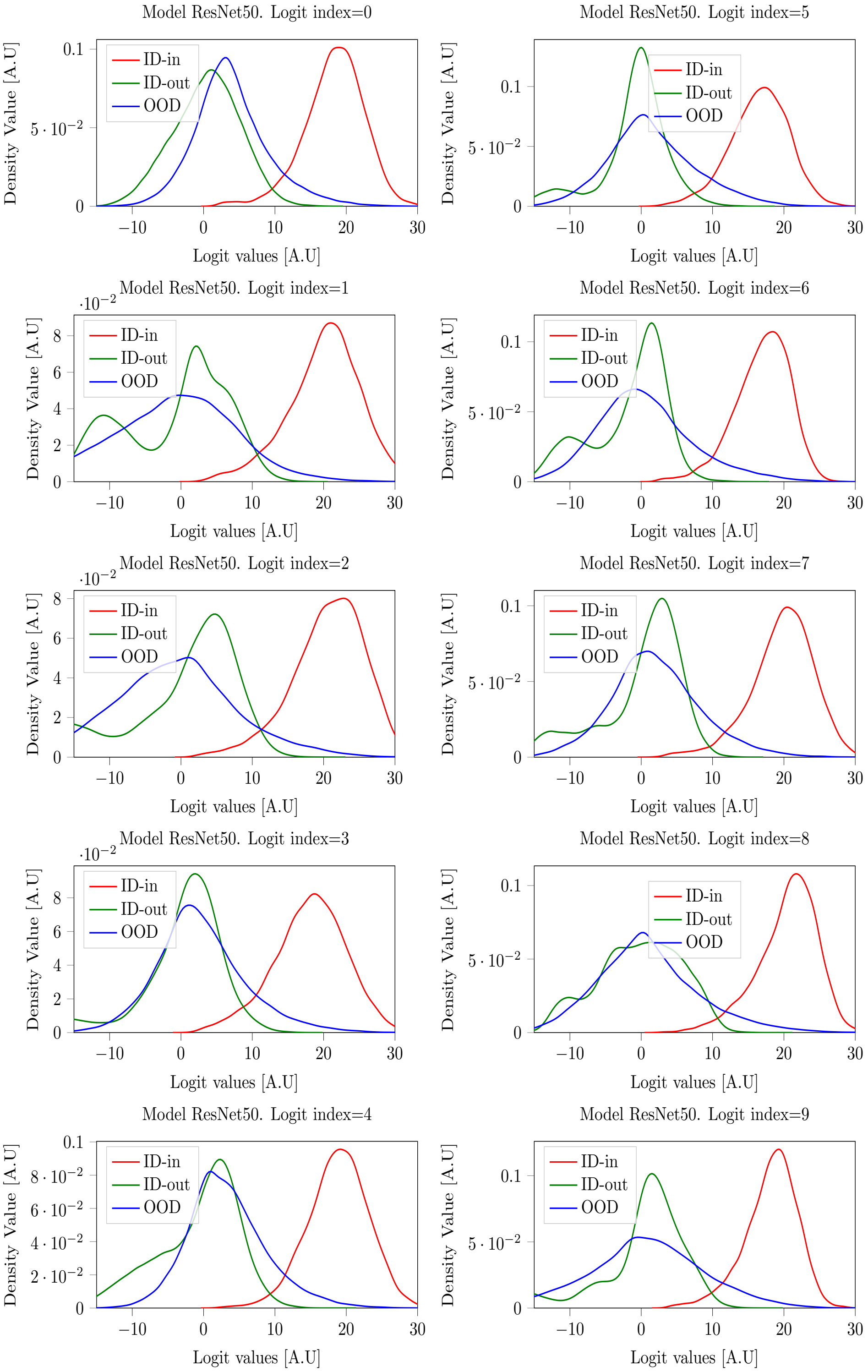
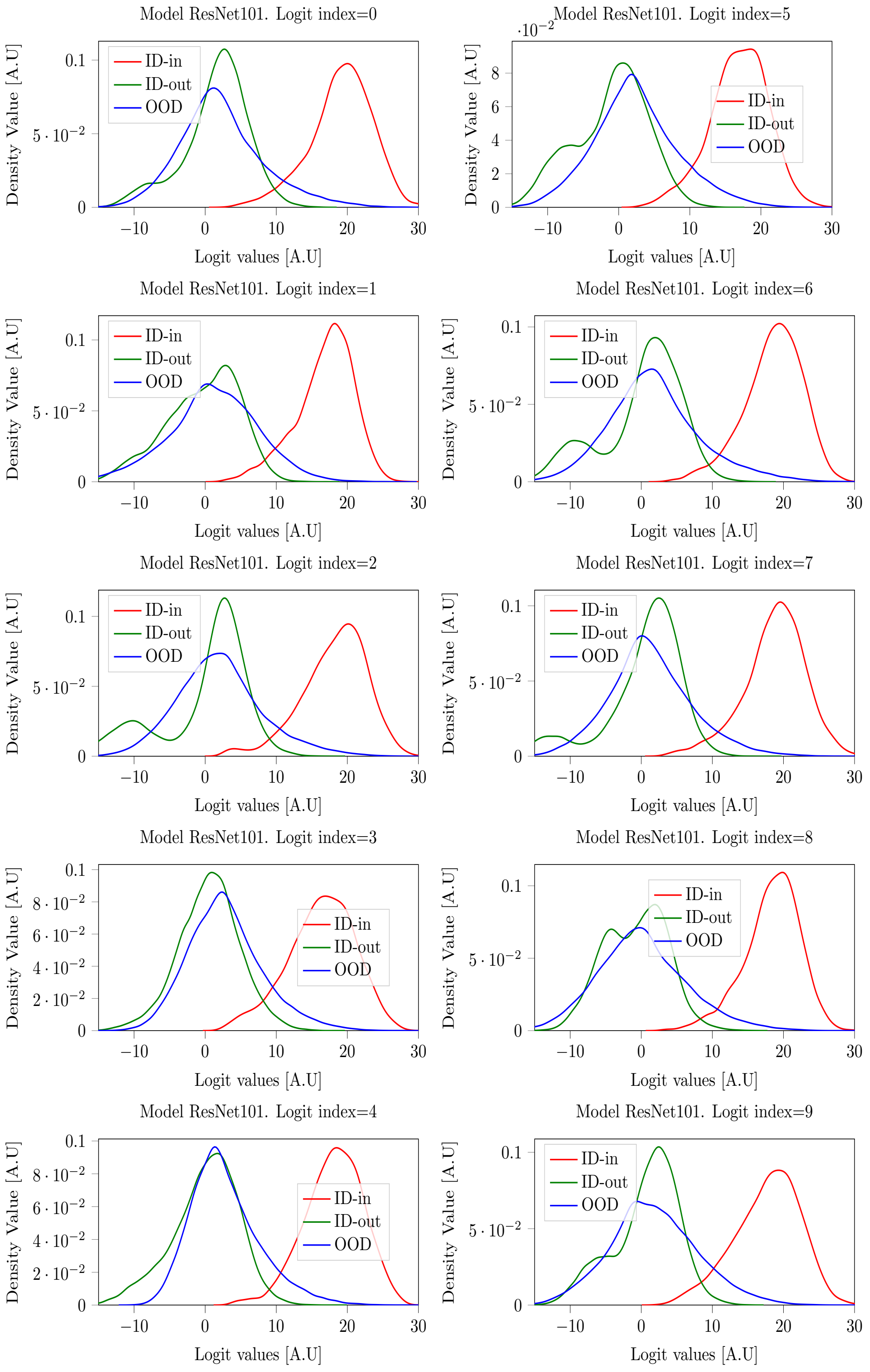


Figure 26: Logit cell densities for CIFAR-10 as ID with ResNet50.



Model ResNet101. Logit index=0

Model ResNet101. Logit index=1

Model ResNet101. Logit index=2

Model ResNet101. Logit index=3

Model ResNet101. Logit index=4

Model ResNet101. Logit index=5

Model ResNet101. Logit index=6

Model ResNet101. Logit index=7

Model ResNet101. Logit index=8

Model ResNet101. Logit index=9

Figure 27: Logit cell densities for CIFAR-10 as ID with ResNet101.

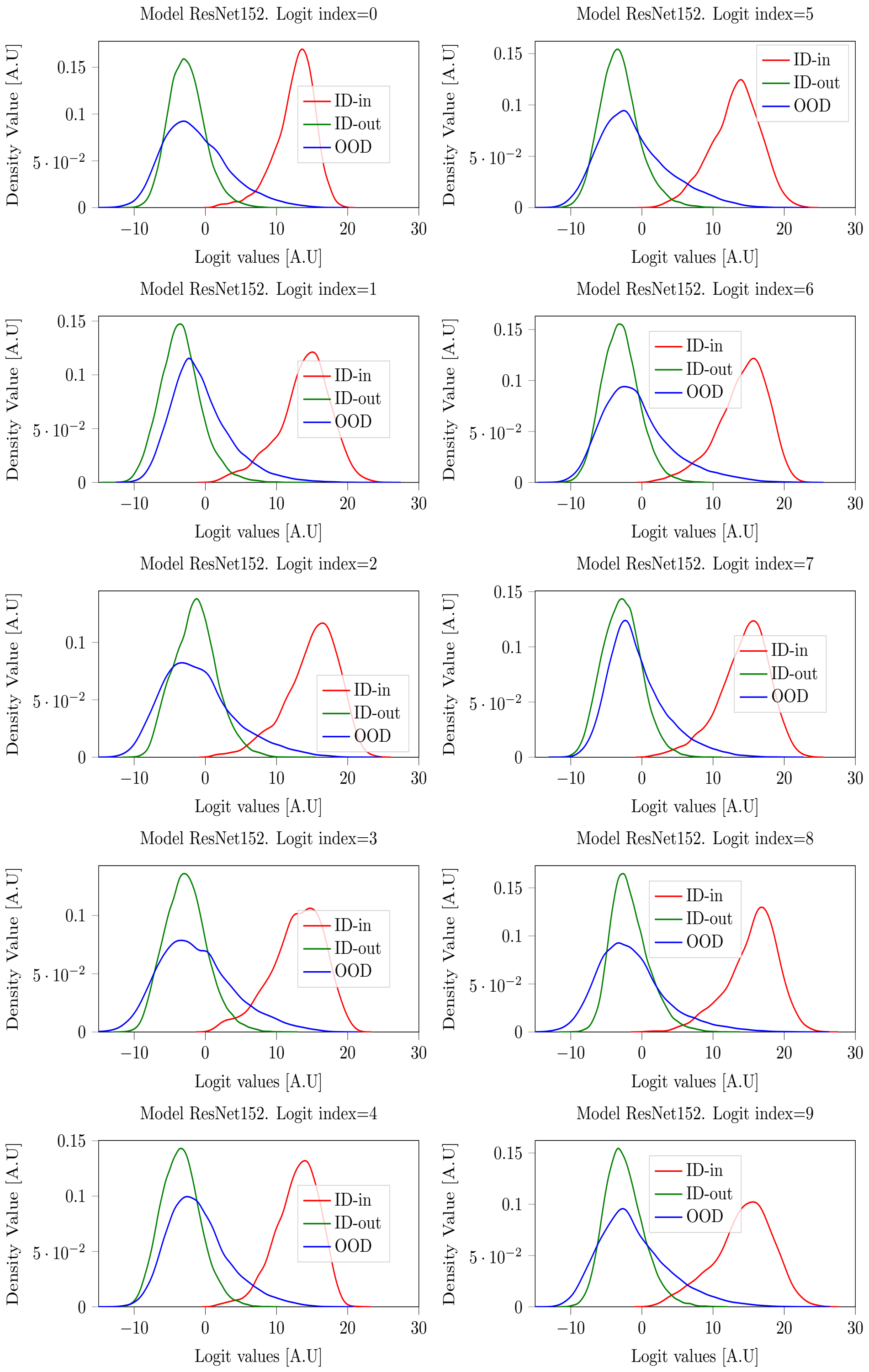


Figure 28: Logit cell densities for CIFAR-10 as ID with ResNet152.

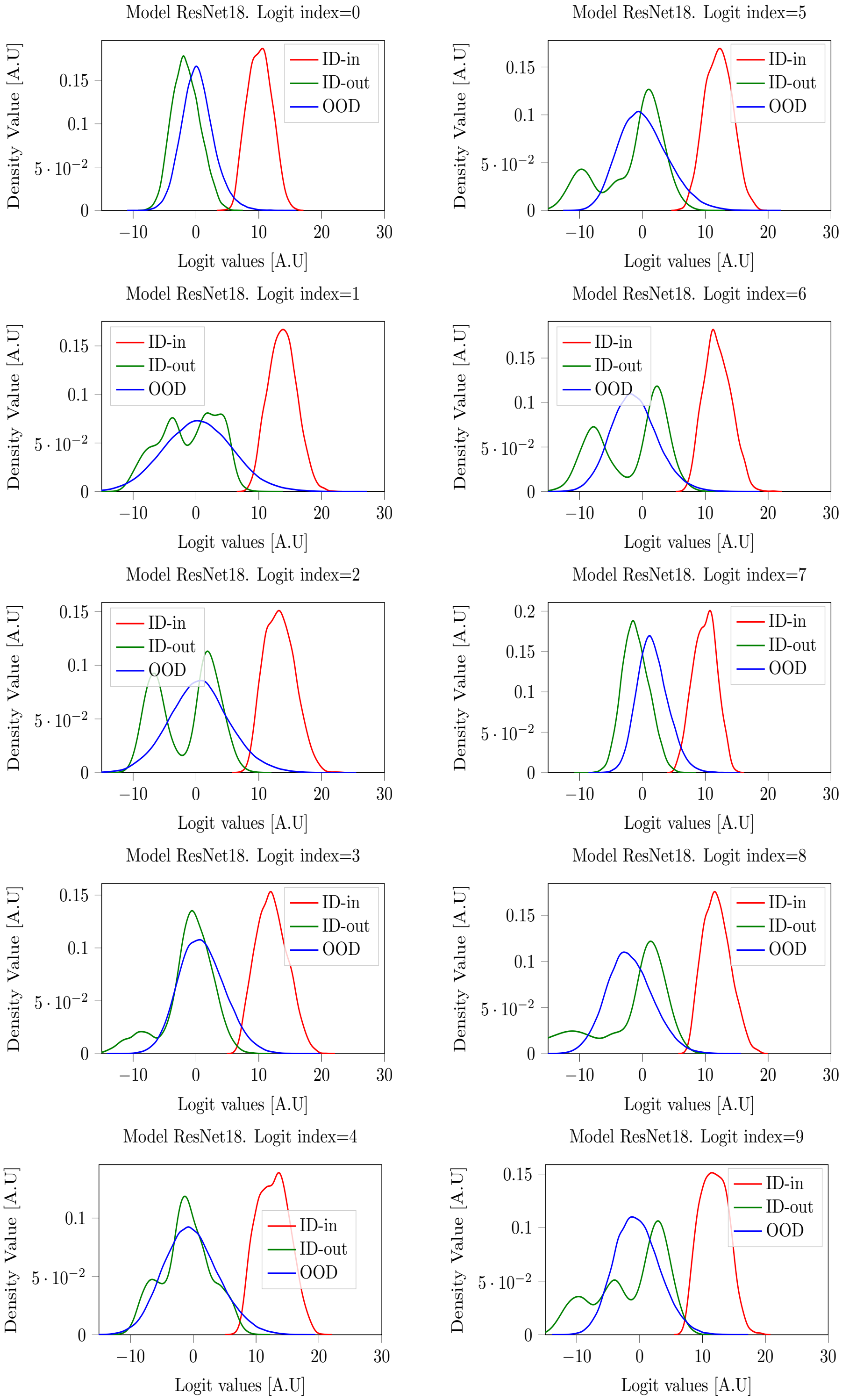


Figure 29: Logit cell densities for SVHN as ID with ResNet18.

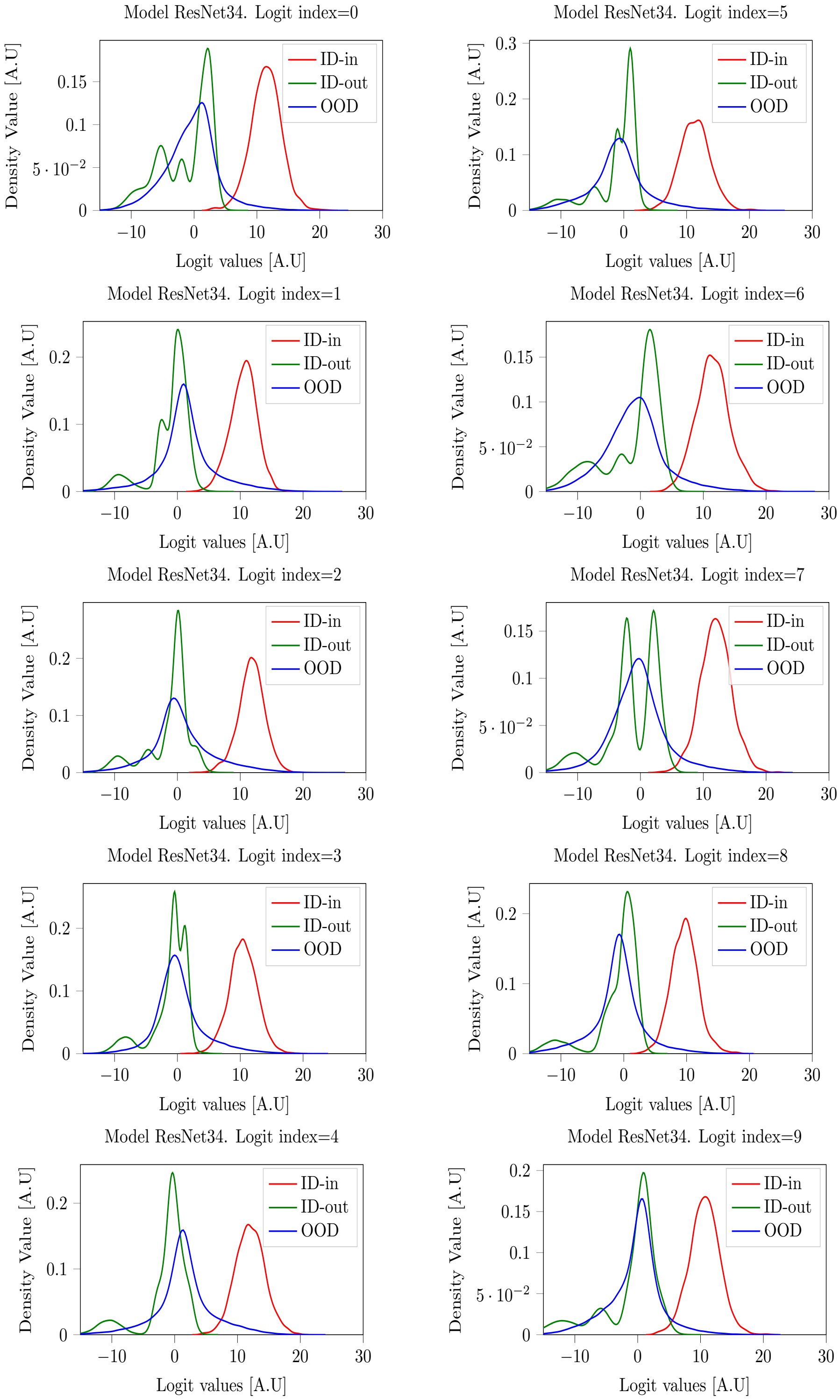


Figure 30: Logit cell densities for SVHN as ID with ResNet34.

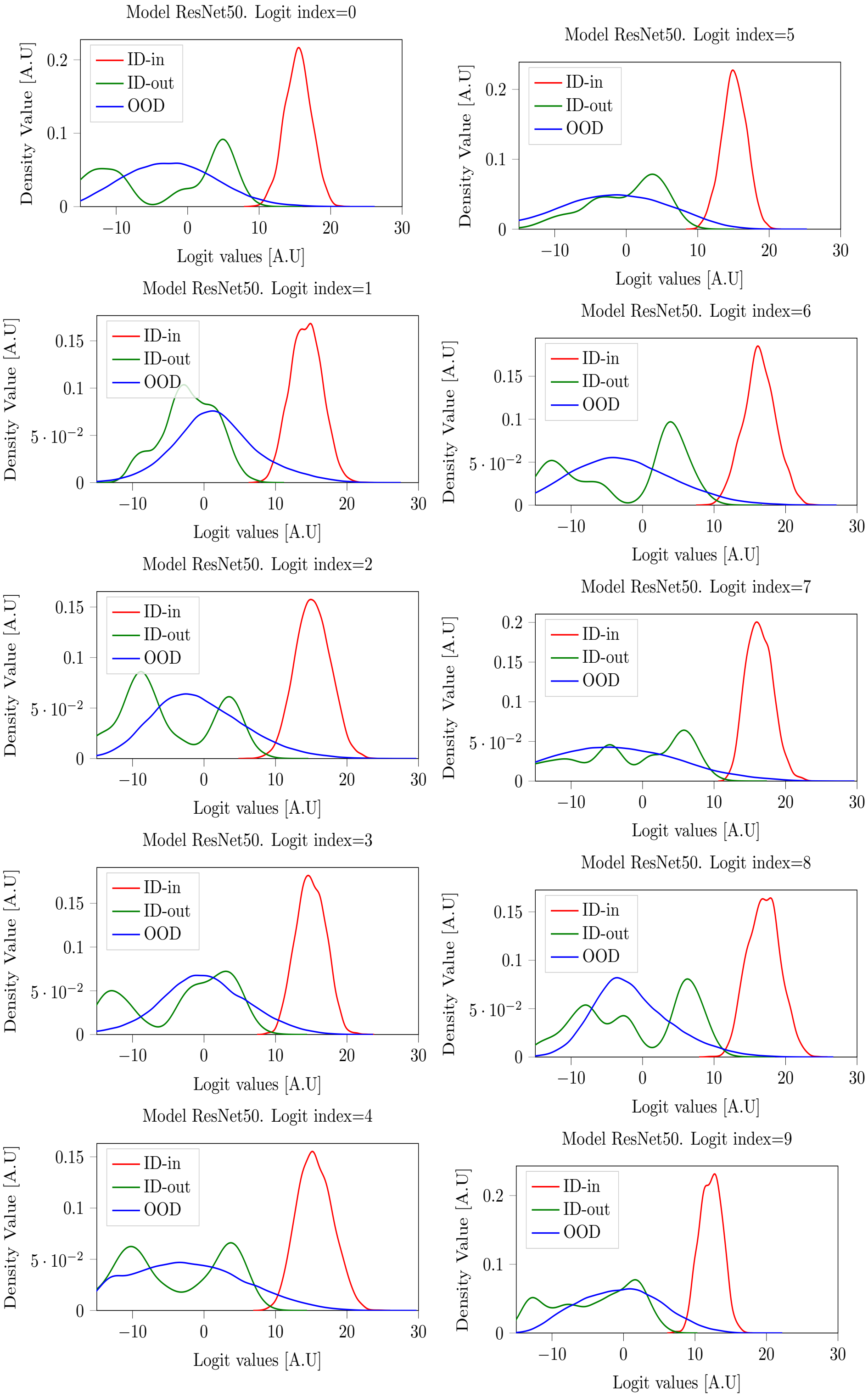
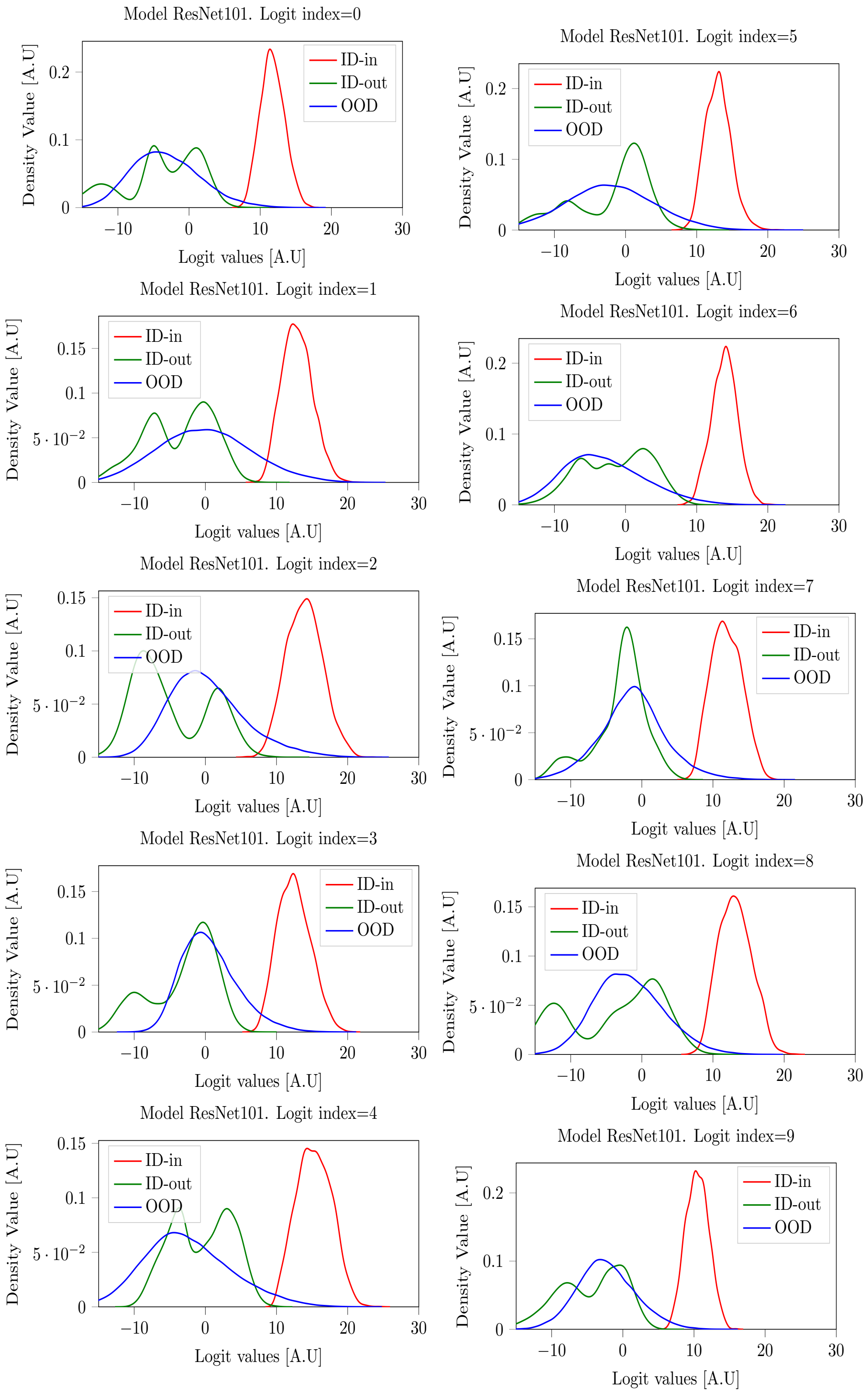


Figure 31: Logit cell densities for SVHN as ID with ResNet50.



Model ResNet101. Logit index=0

Model ResNet101. Logit index=1

Model ResNet101. Logit index=2

Model ResNet101. Logit index=3

Model ResNet101. Logit index=4

Model ResNet101. Logit index=5

Model ResNet101. Logit index=6

Model ResNet101. Logit index=7

Model ResNet101. Logit index=8

Model ResNet101. Logit index=9

Figure 32: Logit cell densities for SVHN as ID with ResNet101.

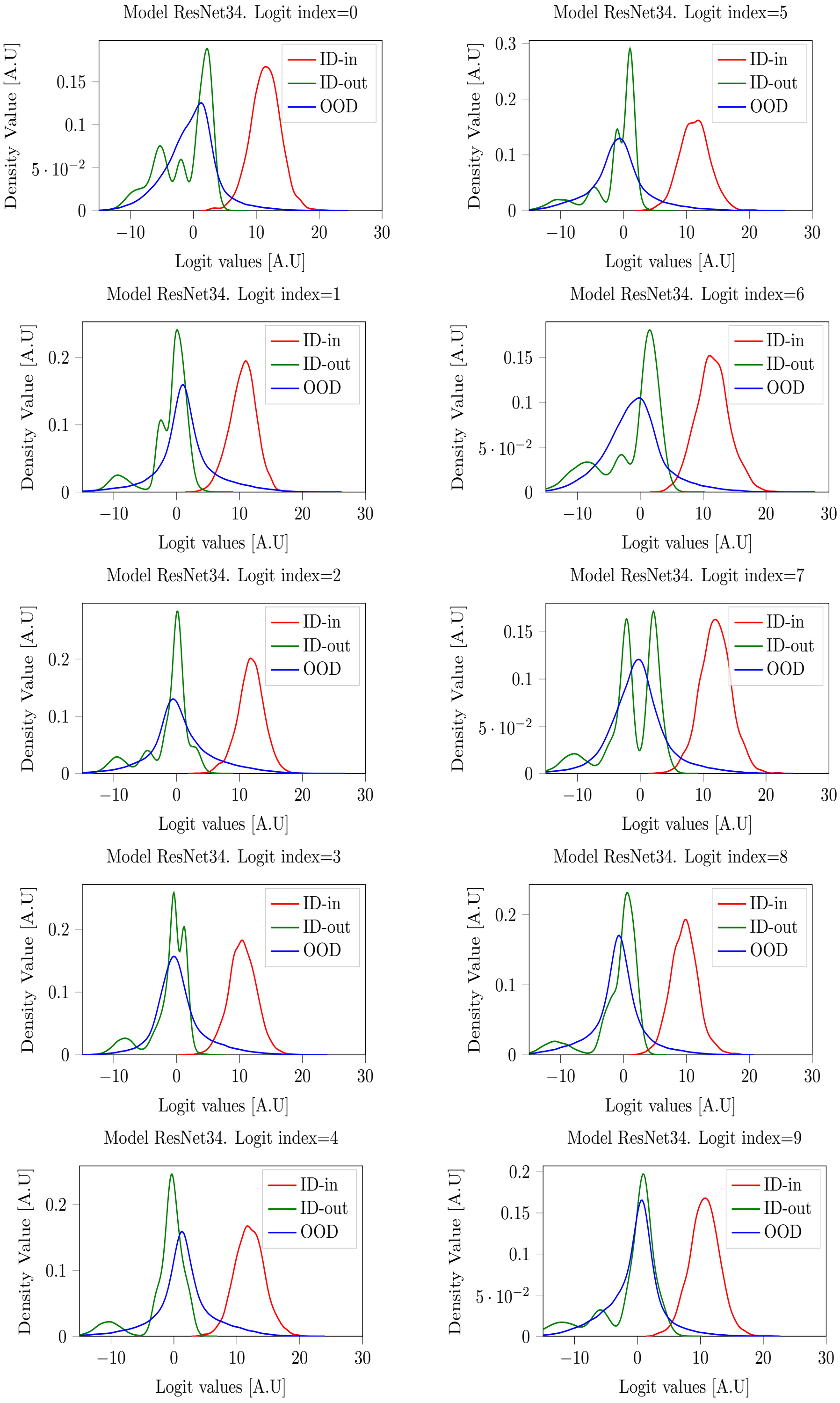


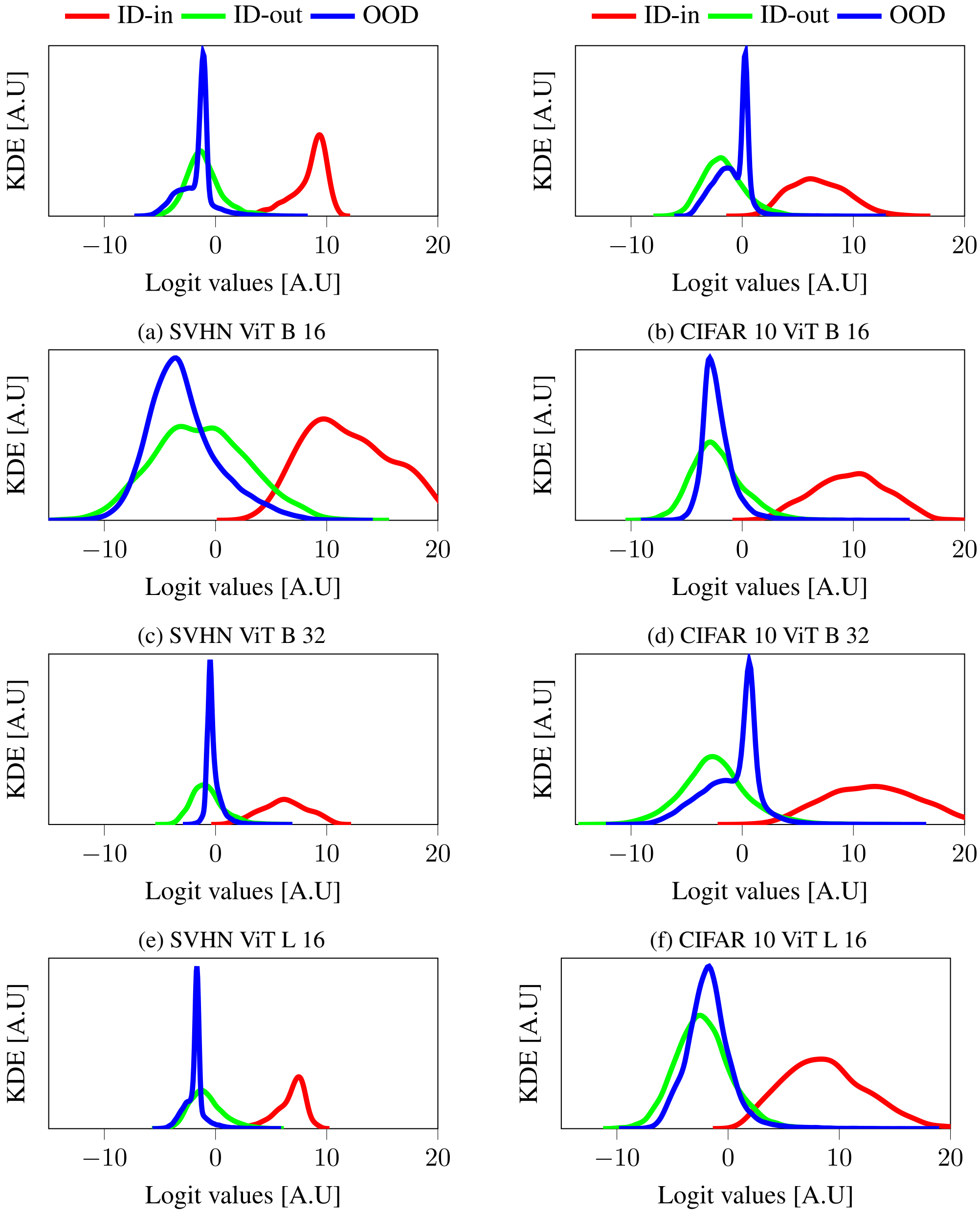
Figure 33: Logit cell densities for SVHN as ID with ResNet152.

G Experiments on different vision transformers

Contrary to traditional convolutional neural networks (e.g., DenseNet, ResNet), which process image patches exclusively on a spatial level, vision transformers (ViT) incorporate an additional component of interleaved processing among patches Dosovitskiy et al. [2021]. To examine the effects of this interleaved processing on the arrangement of OOD and ID logits, we carried out experiments with various ViT configurations, including the base (ViT B) and large (ViT L) models, each with two different patch sizes: 16x16 and 32x32 pixels.

Furthermore, the utlized experimental dataset comprises $\{D\} = \{$ SVHN, CIFAR-100, CIFAR-10, Tiny ImageNet, iSUN, LSUN$\}$. Each model undergoes separate training on CIFAR-10 and SVHN as ID datasets. The remaining datasets are employed as OOD data, specifically $\{D\}/$CIFAR-10 and $\{D\}/$SVHN.

In fig. 34, one can notice that for all versions of the ViT, ID-in logits converge towards higher positive values as expected. Contrarily, the logits for both the ID-out and OOD samples predominantly cluster around zero. Therefore, the intertwined processing among patches in ViT does not alter the anticipated configuration of OOD and ID logits as it is inherently composed of a dot-product operation. Observe that the density plots shown in fig. 34 depict the spread of ID-in, ID-out, and OOD aggregations over all logit cells. For a detailed visual analysis of each logit cell, refer to the various versions of ViT illustrated in figs. 35 to 42.



(a) SVHN ViT B 16

(b) CIFAR 10 ViT B 16

(c) SVHN ViT B 32

(d) CIFAR 10 ViT B 32

(e) SVHN ViT L 16

(f) CIFAR 10 ViT L 16

(g) SVHN ViT L 32

(h) CIFAR 10 ViT L 32

Figure 34: An analysis of the density over logits across distinct ViT architecture trained on the SVHN and CIFAR-10 dataset as the ID data, while the OOD includes $\{D\}/\text{SVHN}$ and $\{D\}/\text{CIFAR-10}$ respectively.

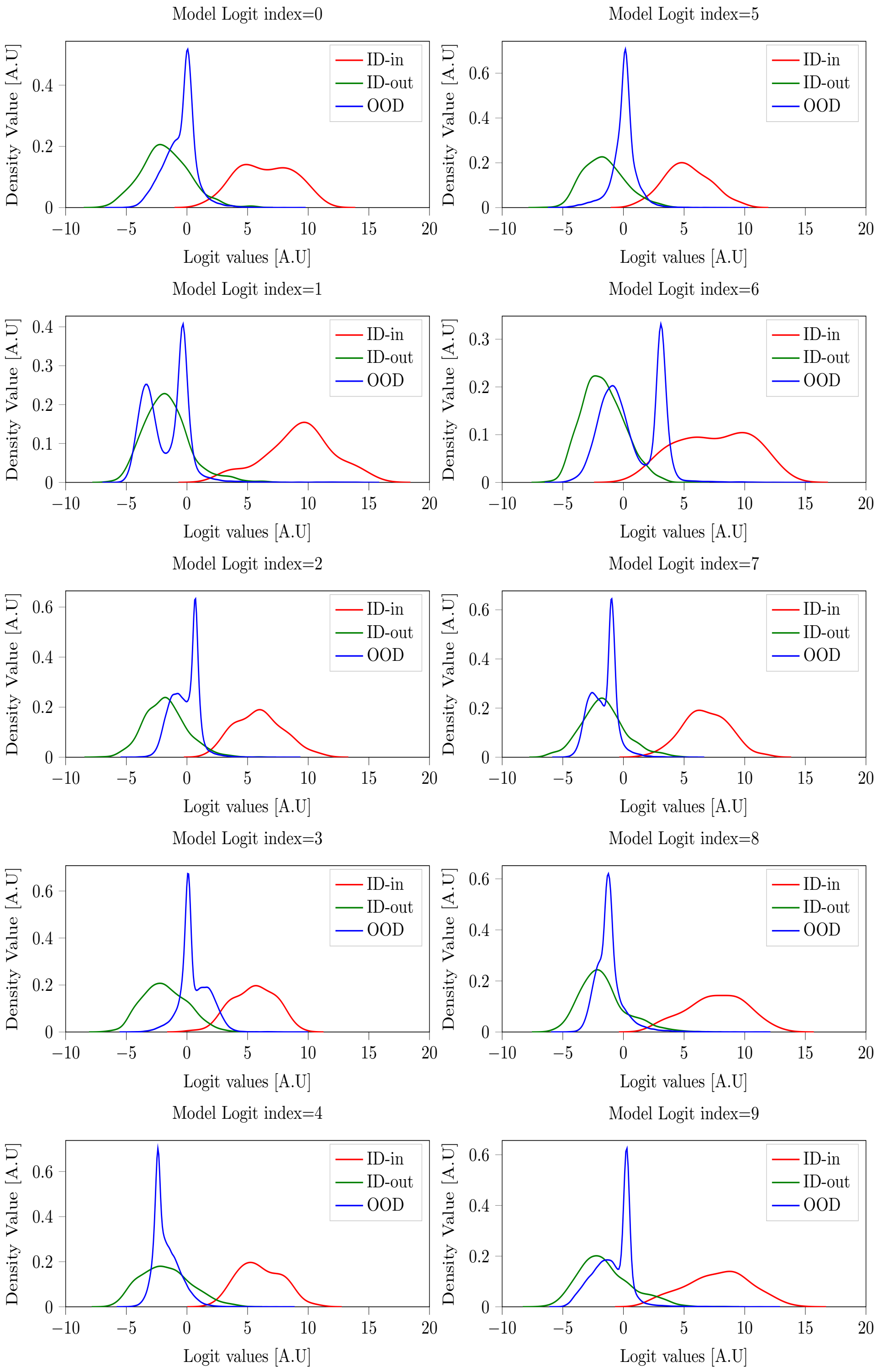


Figure 35: Logit cell densities for CIFAR-10 as ID with ViT-B-16.

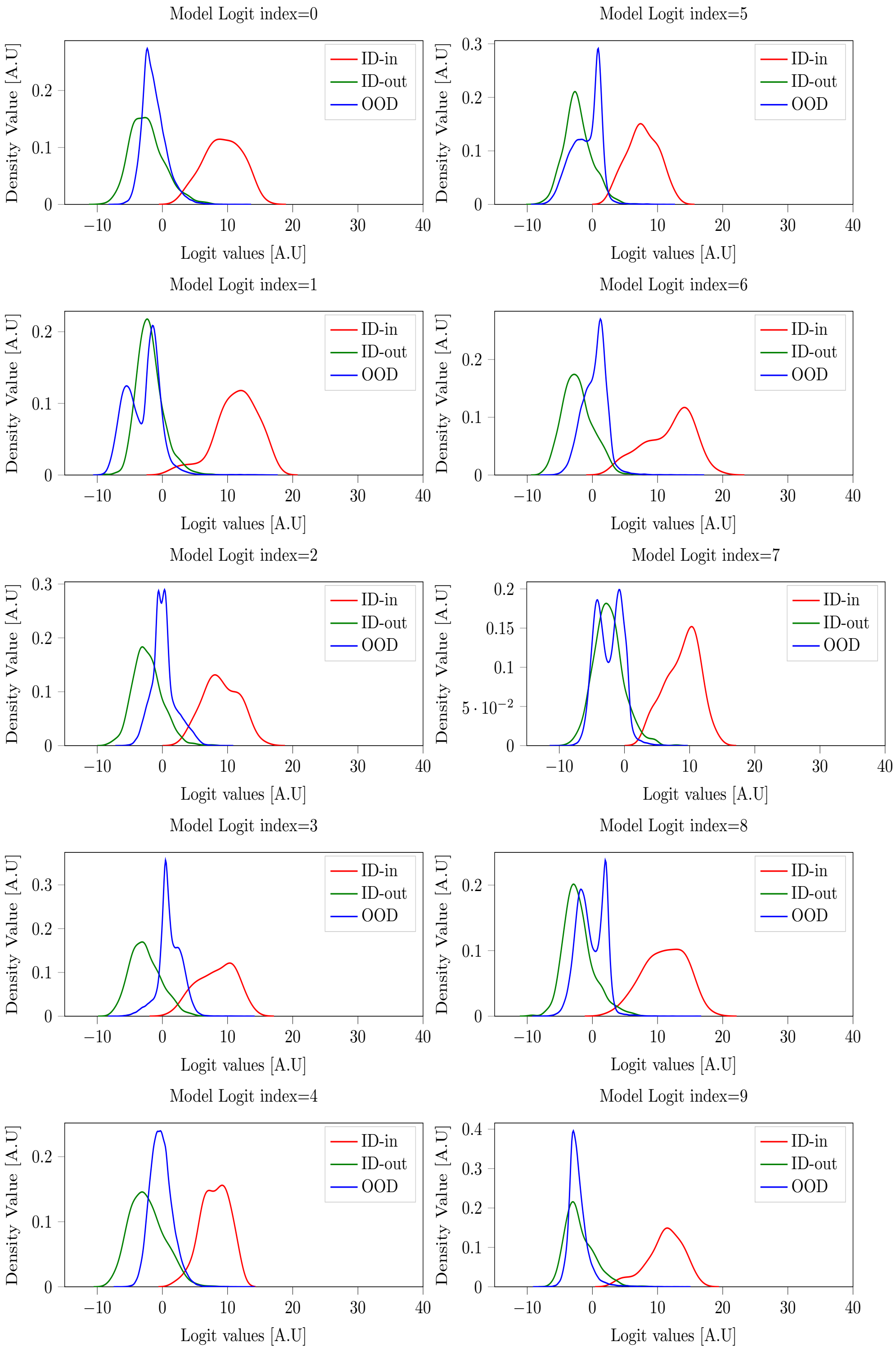


Figure 36: Logit cell densities for CIFAR-10 as ID with ViT-B-32.

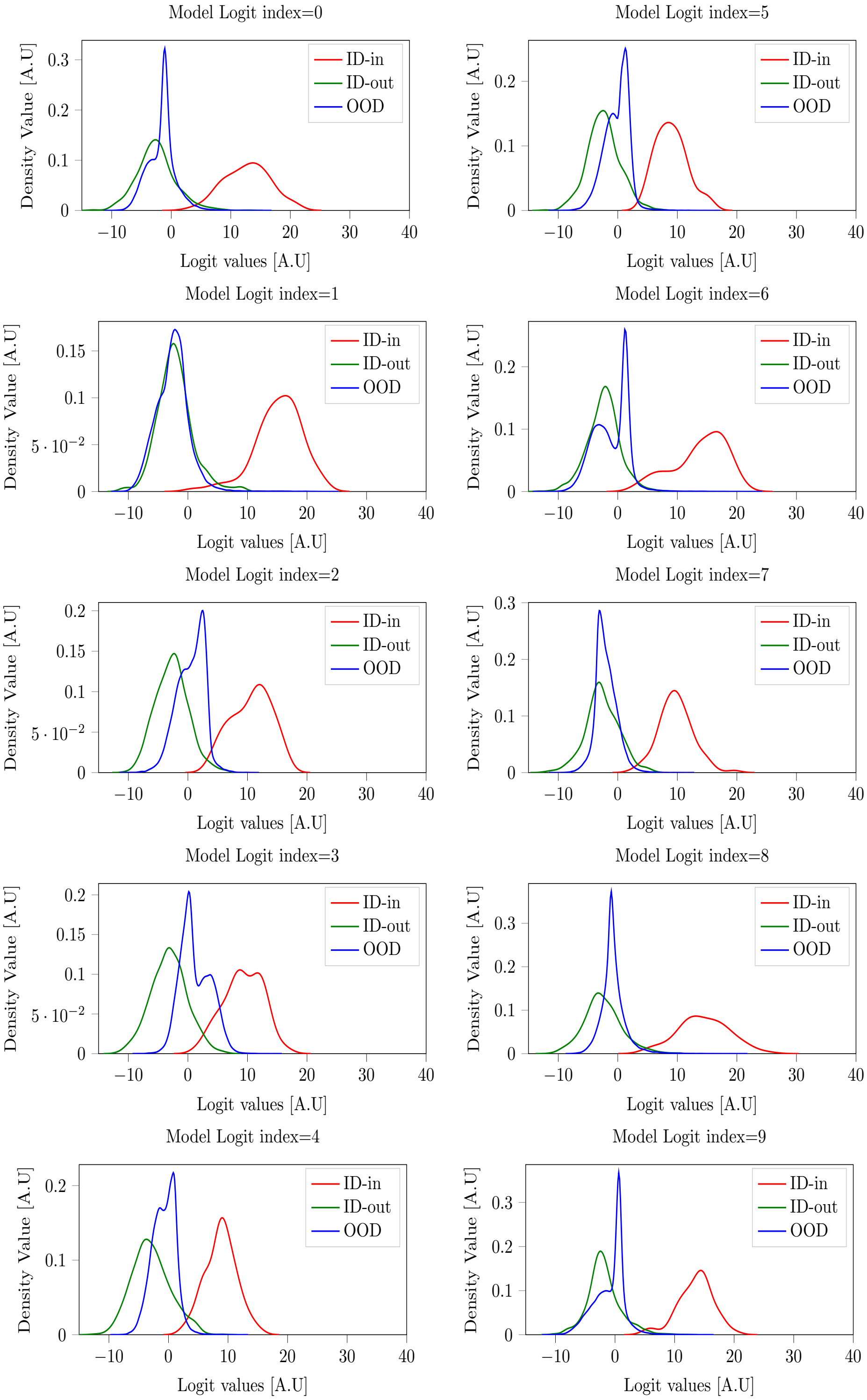


Figure 37: Logit cell densities for CIFAR-10 as ID with ViT-L-16.

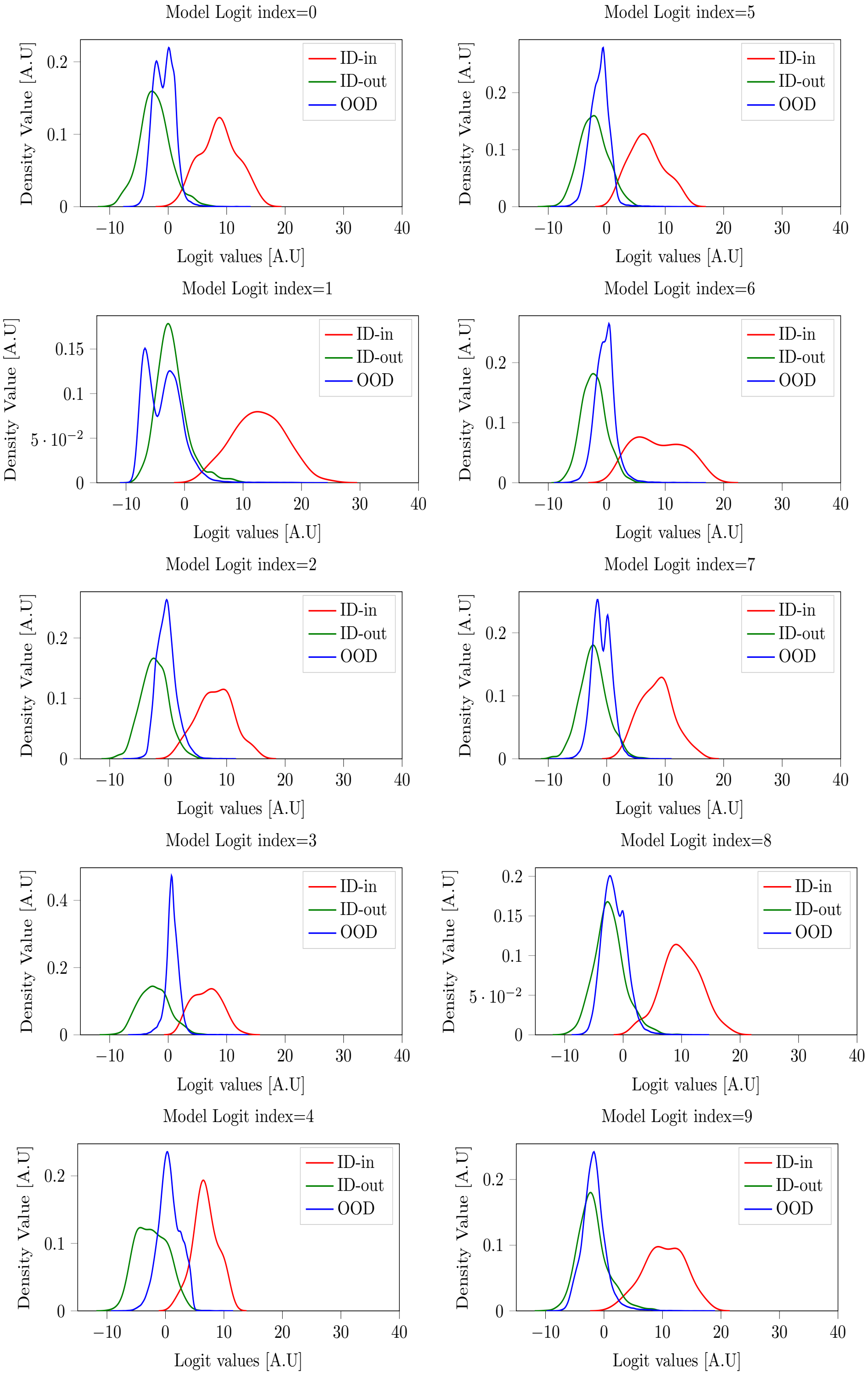


Figure 38: Logit cell densities for CIFAR-10 as ID with ViT-L-32.

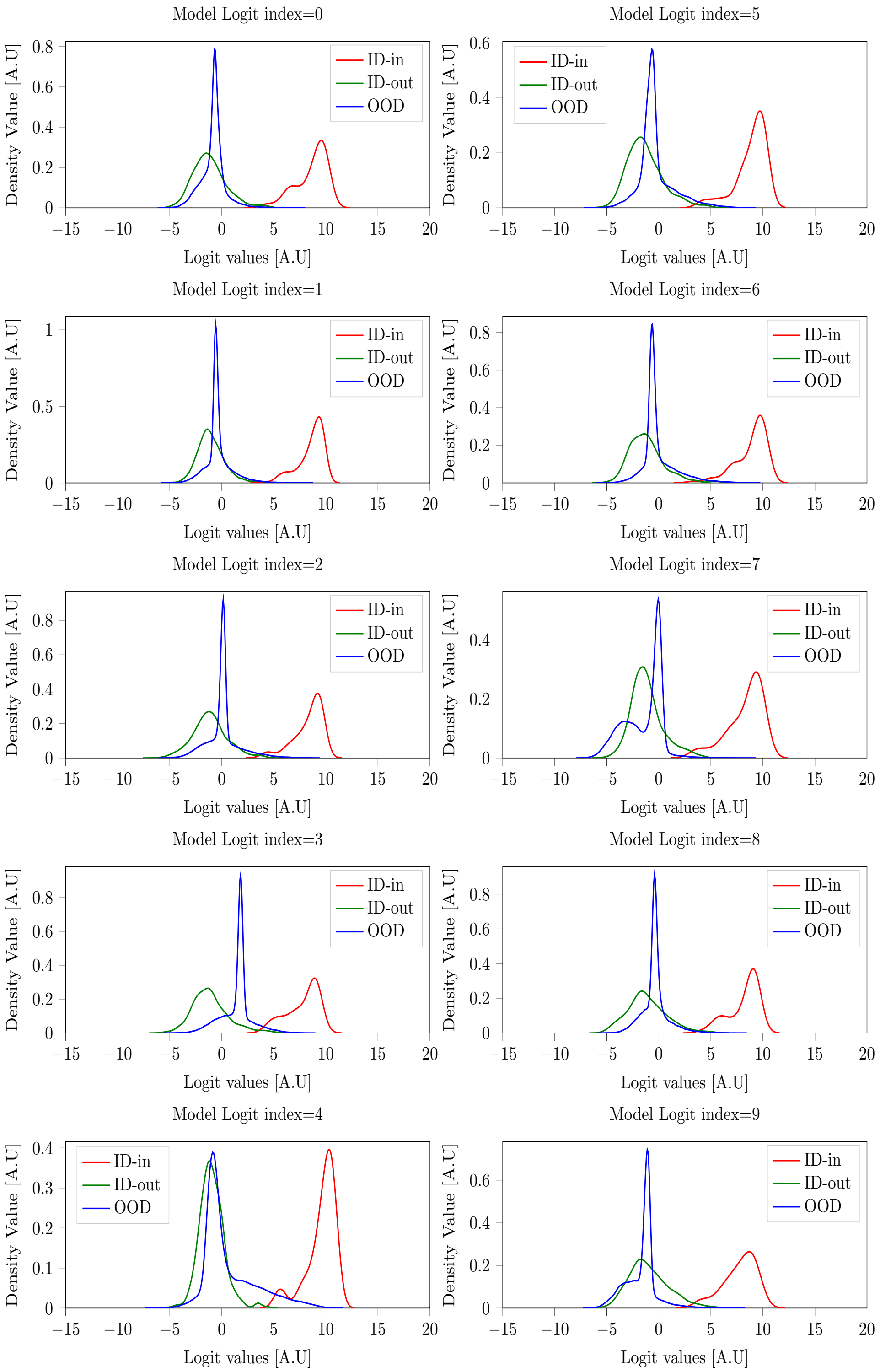


Figure 39: Logit cell densities for SVHN as ID with ViT-B-16.

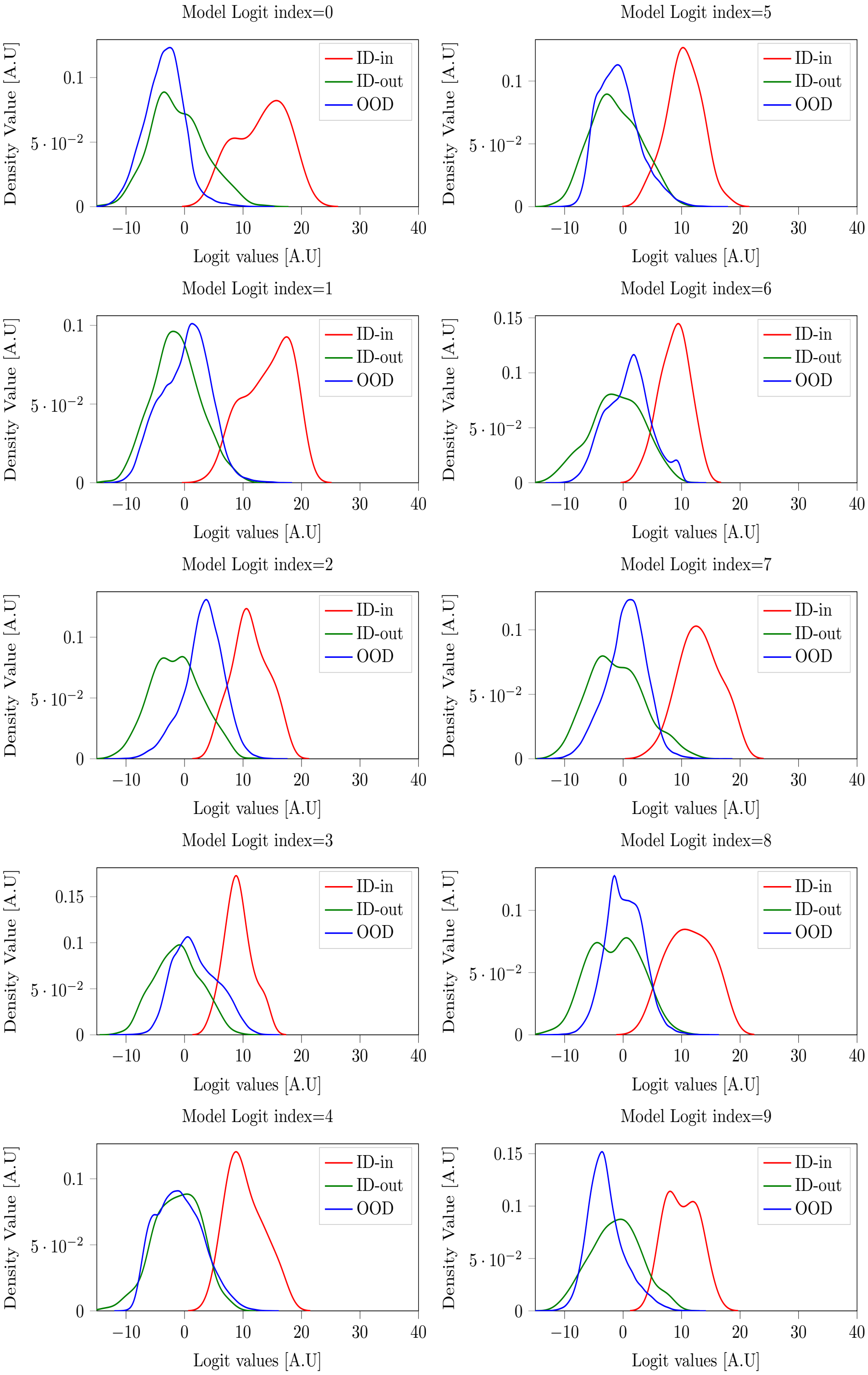


Figure 40: Logit cell densities for SVHN as ID with ViT-B-32.

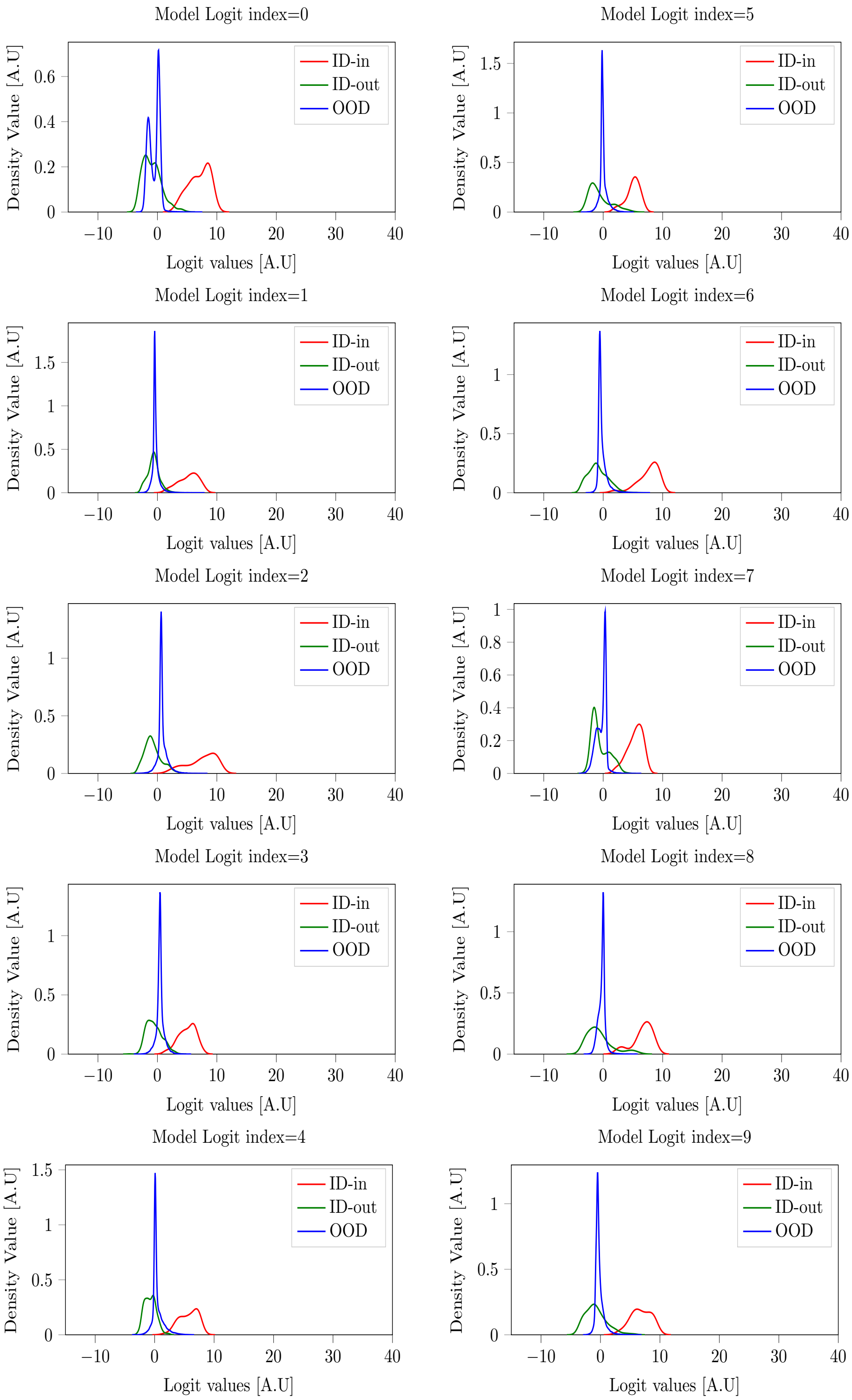


Figure 41: Logit cell densities for SVHN as ID with ViT-L-16.

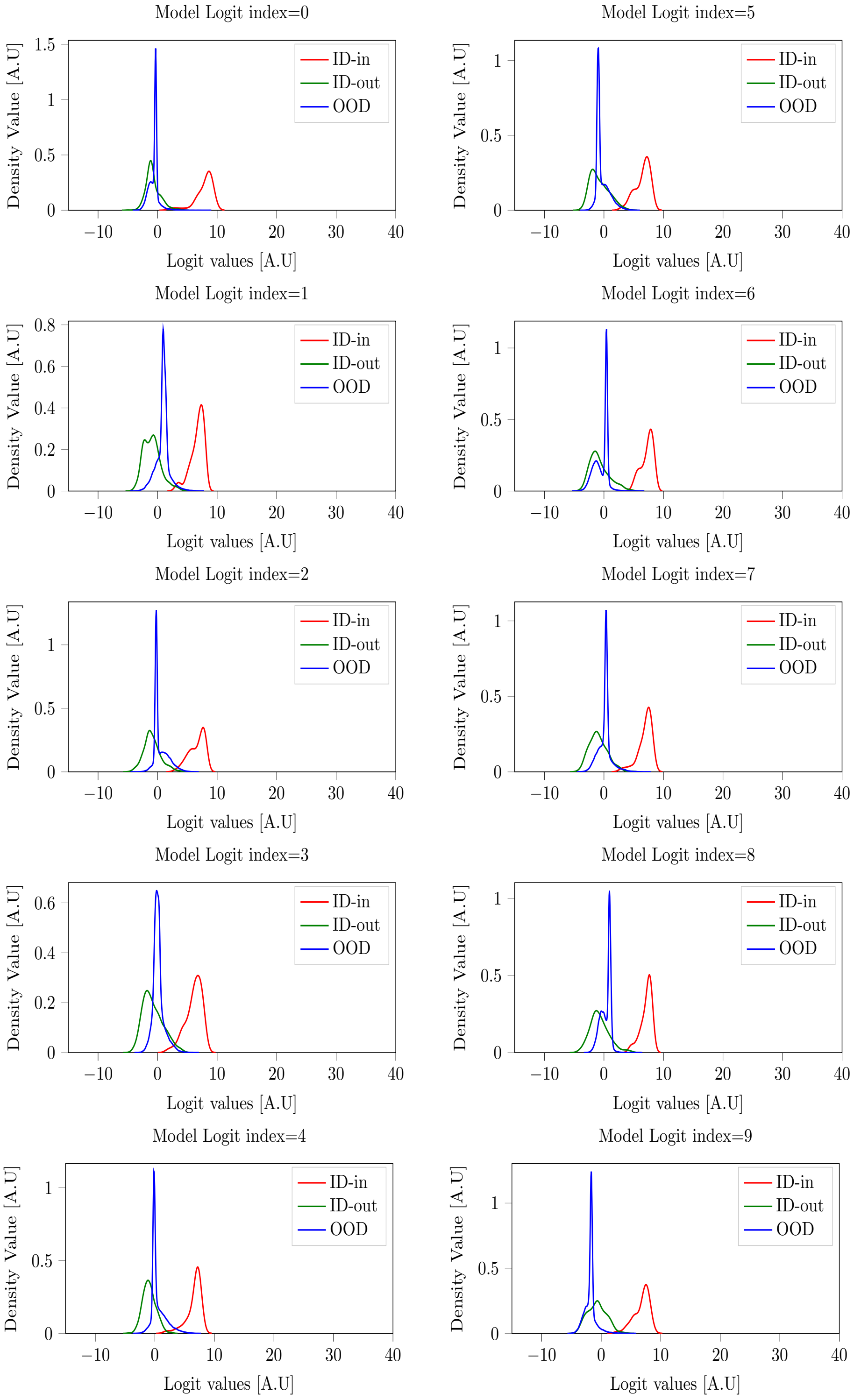


Figure 42: Logit cell densities for SVHN as ID with ViT-L-32.

H Experimentation over grayscale image

The classifier model, which is used for this purpose, consists of three convolutional layers followed by two fully connected layers (see table 2). This model is then trained using the Adam optimizer [Kingma and 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. 43 and 44.

Table 2: A custom-designed model for the experiment on the fashion-MNIST vs. MNIST dataset.

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

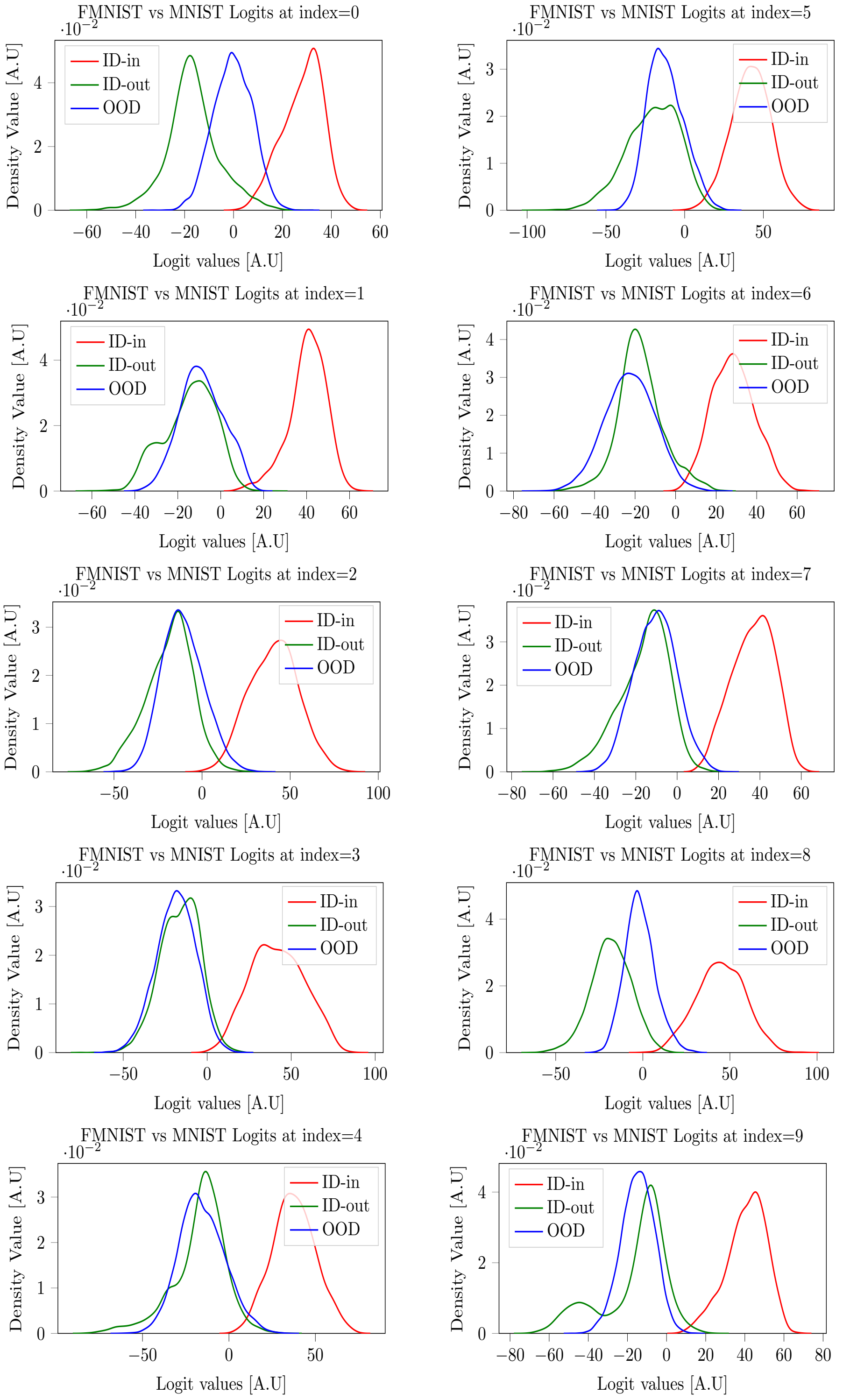


Figure 43: Logit cell densities for FMNIST as ID with model in table 2.

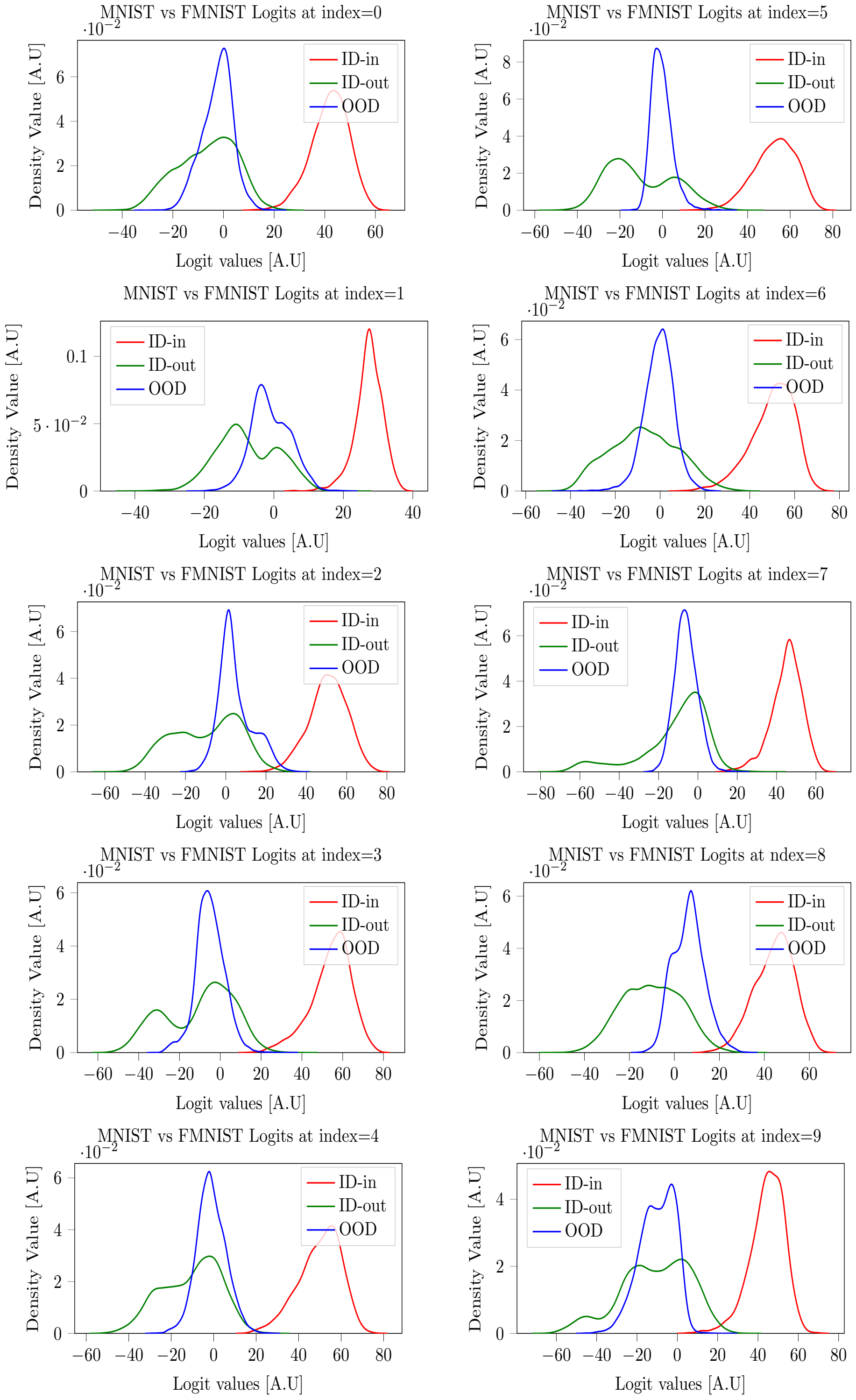


Figure 44: Logit cell densities for MNIST as ID with model in table 2.