A Appendix: OODEEL's Visualization and Benchmarking Features

Visualization OODEEL also comes with quality-of-life features. The first feature includes three useful visualizations that are often used in OOD detection research: Histogram plots for ID and OOD scores, ROC curve, and t-SNE plot of the model's penultimate layer. The first two are used to assess the performance of a detector, and the last is often used to qualitatively evaluate the discriminating capabilities of a given neural network, which often prefigures its Post-hoc OOD detection performances. Examples of these visualizations are found in Figure 9

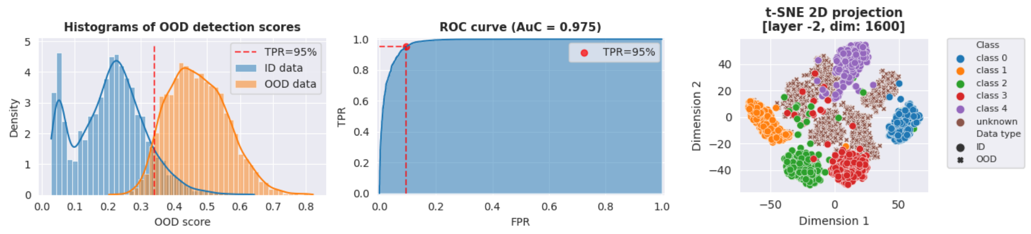
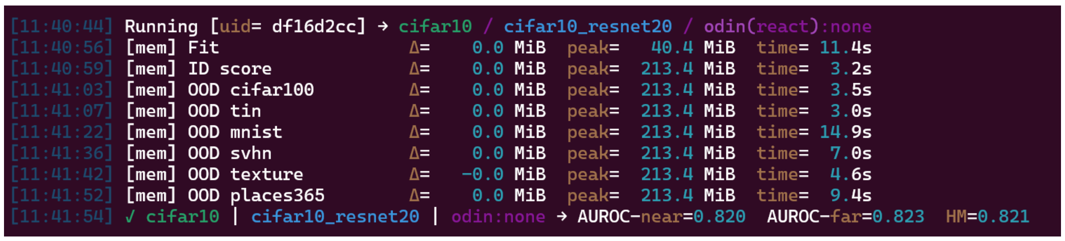


Figure 9: Visualization tools provided by Oodeel. **From left to right:** Histograms and KDE plots for ID vs OOD scores, ROC curve, and t-SNE visualization of a model's feature space.

Benchmarking The last feature is a benchmarking utility that allows for easy testing of the performance of OODEEL's detectors on OpenOOD benchmark. This feature comes with a clean configuration logic, checkpointing, and sharding capabilities, reducing the benchmarking hassle. It also records computational performance metrics, such as VRAM utilisation and execution time. The experiments can be tracked from the terminal through a `rich` interface, and logged to WeightsAndBiases (illustration of the interface in Figure 10). The benchmarking utility is part of another repository called oodeel-banchmark[4] hosted on GitHub.



```
[11:40:44] Running [uid= df16d2cc] → cifar10 / cifar10_resnet20 / odin(react):none
[11:40:56] [mem] Fit                 Δ=    0.0 MiB  peak=   40.4 MiB  time= 11.4s
[11:40:59] [mem] ID score            Δ=    0.0 MiB  peak=  213.4 MiB  time=  3.2s
[11:41:03] [mem] OOD cifar100        Δ=    0.0 MiB  peak=  213.4 MiB  time=  3.5s
[11:41:07] [mem] OOD tin             Δ=    0.0 MiB  peak=  213.4 MiB  time=  3.0s
[11:41:22] [mem] OOD mnist           Δ=    0.0 MiB  peak=  213.4 MiB  time= 14.9s
[11:41:36] [mem] OOD svhn            Δ=    0.0 MiB  peak=  213.4 MiB  time=  7.0s
[11:41:42] [mem] OOD texture         Δ=   -0.0 MiB  peak=  213.4 MiB  time=  4.6s
[11:41:52] [mem] OOD places365       Δ=    0.0 MiB  peak=  213.4 MiB  time=  9.4s
[11:41:54] ✓ cifar10 | cifar10_resnet20 | odin:none → AUROC-near=0.820  AUROC-far=0.823  HM=0.821
```

Figure 10: Caption

B Appendix: Details on the Benchmark Setting

In this appendix, we provide a complete description of the hardware, software, datasets, models, and orchestration logic used in our out-of-distribution (OOD) detection benchmark. Our goal is to enable exact reproduction of all experiments.

B.1 Compute Environment

We ran all experiments on a single machine equipped with:

Hardware

- 2 × NVIDIA RTX 4090 GPUs (24 GB VRAM each)
- CPU RAM: 128 GB

Software

[4] `https://github.com/deel-ai/oodeel-benchmark`

- Python 3.10, PyTorch 2.7.0, torchvision 0.22.0
- OODeel library (dev branch, commit `ff4fa17`, will soon be merged)

Runtime & Logging

- Wall-clock time: $\approx$1 week on 2 × RTX 4090
- Batch size: 64 per GPU; peak VRAM $\leq$ 20 GB
- Metrics streamed to Weights & Biases (and available in "All runs" leaderboard from `https://oodeel-benchmark.streamlit.app/`)

B.2 Repository & Benchmark Orchestration

All code is available at `https://github.com/deel-ai/oodeel-benchmark`. We drive the full sweep with:

```
python -m src.run --num-shards 2 --shard-index 0
```

This script:

1. Loads every YAML in `configs/datasets/`, `configs/models/`, `configs/methods/`.
2. Forms the Cartesian product of (dataset, model, method, hyperparameters) configurations.
3. Shards the list via `-num-shards/-shard-index`.
4. Skips runs with existing `.parquet` outputs in `results/`.
5. Streams raw scores, AUROC, and TPR@5% FPR to W&B.

B.3 Sweep Construction & Config Logic

Each YAML file defines a flat key–value mapping:

- `configs/datasets/*.yaml`: list of near- and far-ood datasets for a given id dataset, model names.
- `configs/models/*.yaml`: architecture name, checkpoint source, feature-extraction layers.
- `configs/methods/*.yaml`: detector type and hyperparameter grid.

The driver merges one file from each folder into a single run configuration. Sharding evenly slices this list across processes, and checkpointing ensures idempotent restarts.

B.4 Datasets & Models Tested

We follow the OpenOOD benchmark in pairing each in-distribution (ID) dataset with semantically *near* and *far* OOD test sets.

CIFAR–10:

- **Near-OOD:** CIFAR–100, Tiny ImageNet
- **Far-OOD:** MNIST, SVHN, Textures, Places365

CIFAR–100:

- **Near-OOD:** CIFAR–10, Tiny ImageNet
- **Far-OOD:** MNIST, SVHN, Textures, Places365

ImageNet–1k:

- **Near-OOD:** SSB-hard, NINCO
- **Far-OOD:** iNaturalist, Textures, OpenImage-O

Model Architectures We evaluate 11 models on CIFAR and 7 on ImageNet, including:

- **ResNets:** ResNet-18, ResNet-32, ResNet-50
- **VGG:** VGG11, VGG16, RepVGG-A0, RepVGG-A2
- **MobileNets:** MobileNet v2, MobileNet v3-Large

- **ShuffleNets:** ShuffleNet v2
- **Transformers:** ViT-B/16, Swin-Tiny
- **RegNet:** RegNetY-16GF

B.5 OOD Detectors & Hyperparameter Sweeps

We evaluate 12 base OOD detection methods (e.g. MSP, Energy, Mahalanobis, ODIN, DKNN, SHE, VIM, . . .) and for each base method construct variants as follows:

- **Logits-based variants:** with and without ReAct clipping, ASH, and SCALE transformations.
- **Feature-based variants:** with and without score aggregation over different feature layers.

This yields a total of 42 distinct method configurations. All hyperparameter search grids are specified in the `configs/methods/` folder of the oodeel-benchmark repository.

B.6 Logging, Results & Reproducibility

Each run writes a Parquet file in `results/` containing raw scores, AUROC, and TPR@5% FPR. Visualizations are provided via console bars (Rich) and W&B dashboards. All code, exact commit hashes, and example launch commands (including seed settings in `src/utils.py`) are publicly available, ensuring full reproducibility.

C Appendix: Complementary benchmark results

In this section, we include complementary plots to the main document for CIFAR-100 and Far-OOD experiments. This section is structured as the experiments section.

C.1 Impact of the Model

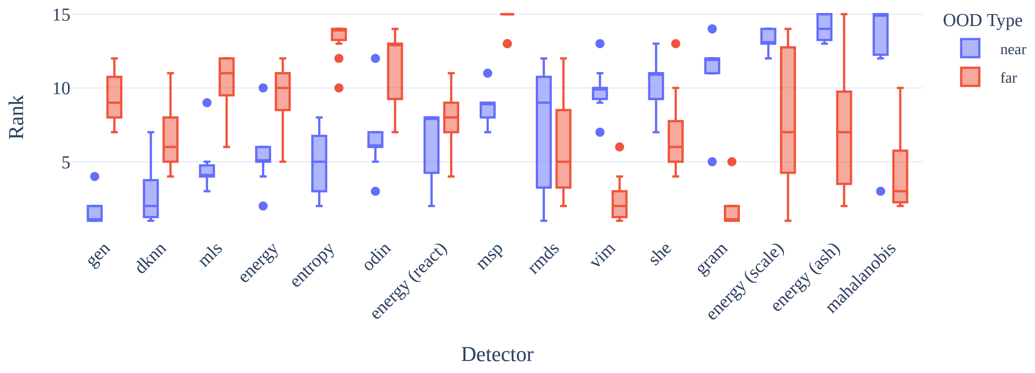


Figure 11: Box plots of the rank for each detector on CIFAR-100. Methods are sorted according to their mean rank on near-OOD. For CIFAR-100, the rank is very different for Near-OOD and Far-OOD, which corroborates the observation of the main paper that performances on Near and Far OOD do not always correlate.

We also report the Spearman rank correlation heatmap between the detectors' rank per model in Figure 12 for CIFAR-10, CIFAR-100, and ImageNet. We can see that there is some correlation in the rank of the detectors for CIFAR-10 and CIFAR-100, even if it is not high enough to alleviate the need to test every detector on various models. For ImageNet, the correlation is significantly lower, emphasizing this need even strongly.

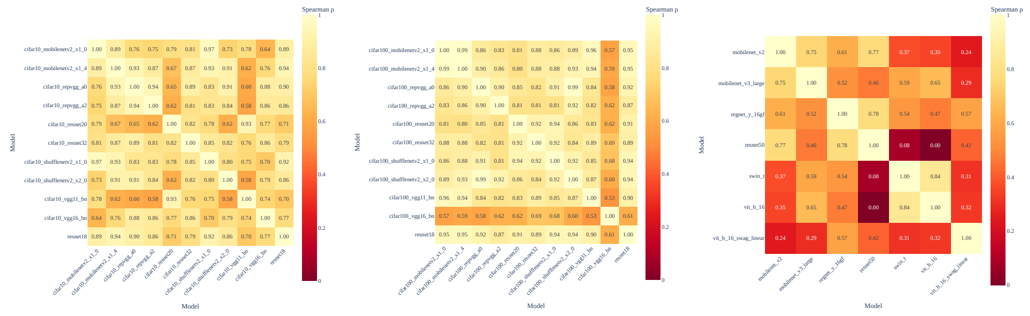


Figure 12: Spearman rank correlation between the AUROC of the detectors for each model. **From left to right:** CIFAR-10, CIFAR-100, ImageNet.

C.2 Impact of Layer-wise Score Aggregation

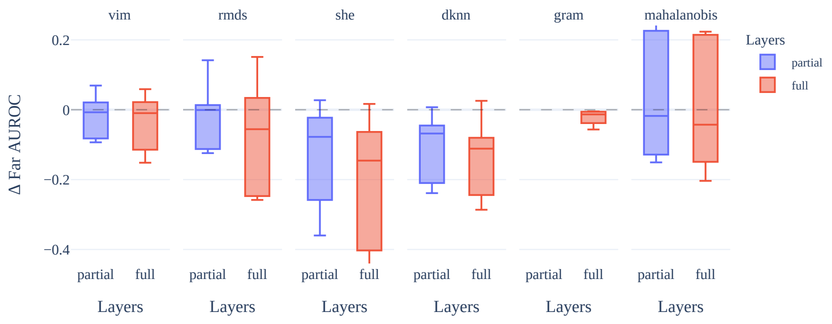


Figure 13: Box plot of the Δ in AUROC between feature-based detector applied on the penultimate layer and either a subset of the internal layers (partial) or all internal layers (full) for **ImageNet Far-OOD**.

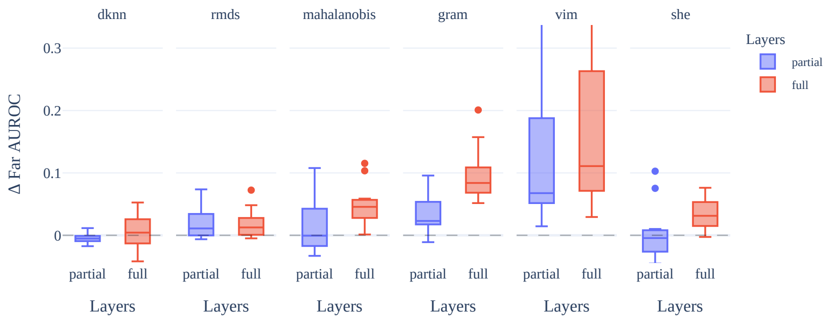


Figure 14: Box plot of the Δ in AUROC between feature-based detector applied on the penultimate layer and either a subset of the internal layers (partial) or all internal layers (full) for **CIFAR-10 Far-OOD**.

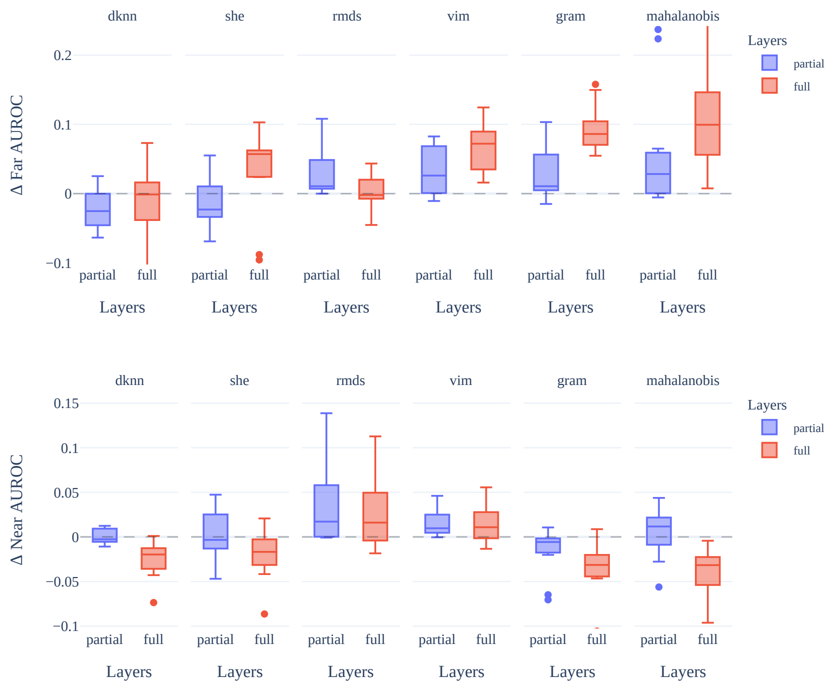


Figure 15: Box plot of the Δ in AUROC between feature-based detector applied on the penultimate layer and either a subset of the internal layers (partial) or all internal layers (full) for **top**: CIFAR-100 Near-OOD, **bottom**: CIFAR-100 Far-OOD

444 ## C.3 Impact of Activation Shaping

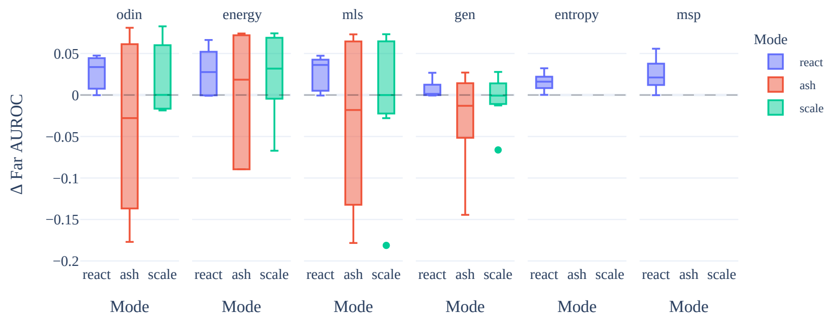


Figure 16: Box plot of the Δ in AUROC between logit-based detectors and their augmentation with activation-shaping techniques (ReAct, SCALE, and ASH) for **ImageNet Far-OOD**.

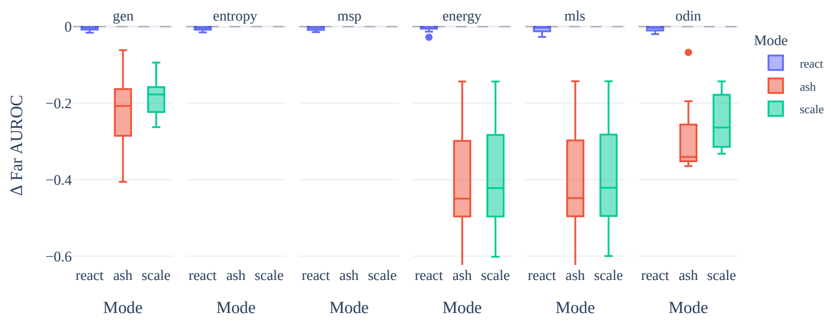


Figure 17: Box plot of the Δ in AUROC between logit-based detectors and their augmentation with activation-shaping techniques (ReAct, SCALE, and ASH) for **CIFAR-10 Far-OOD**.

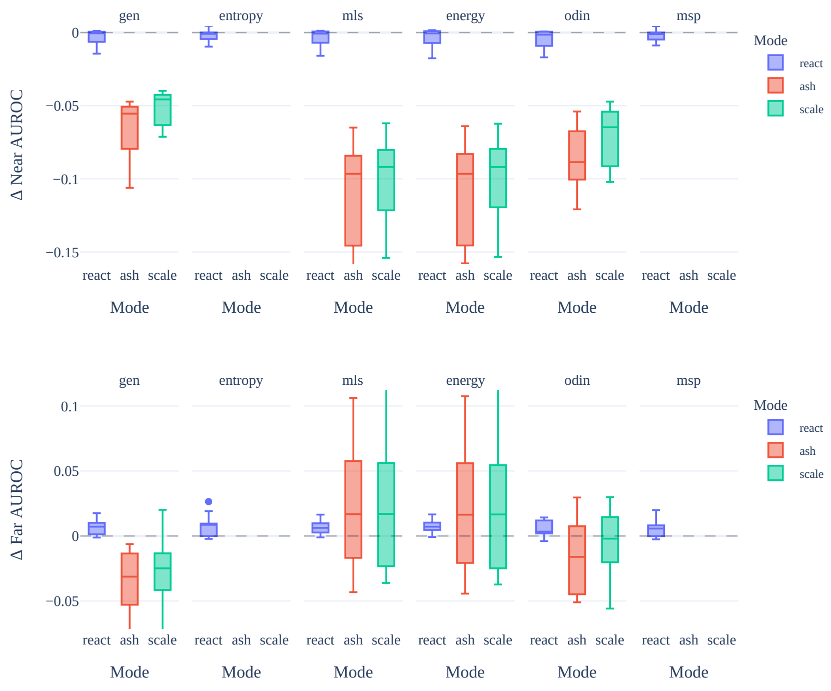


Figure 18: Box plot of the Δ in AUROC between logit-based detectors and their augmentation with activation-shaping techniques (ReAct, SCALE, and ASH) for **top**: CIFAR-100 Near-OOD, **bottom**: CIFAR-100 Far-OOD. Interestingly, the effect of activation shaping on Far-OOD is quite beneficial for CIFAR-100 Far-OOD, whereas it is not for CIFAR-100 Near-OOD or CIFAR-10 Far-OOD.

D Appendix: Sanity check of OODEEL detectors implementation by performance comparison with OpenOOD detectors

To ensure the reliability of our OOD detection results using OODEEL, we conducted a sanity check by comparing our outcomes with those reported on the OpenOOD leaderboard. This validation step focused on both near-OOD detection tasks presented in Table 1.

Note that we obtain a significant improvement for the Gram method from OODEEL with respect to OpenOOD. It stems from a key implementation detail: in OODEEL, the Gram score relies on computing deviations based on quantiles (e.g., the 1st and 99th percentiles) of the features, whereas

OpenOOD uses the absolute minimum and maximum feature values. The use of quantiles in OODEEL leads to a more robust formulation for outliers.

Globally, the obtained AUROC aligns with OpenOOD. The differences are mainly slight, with OODEEL doing better than the OpenOOD implementation for 22 cases, equal for 4, and worse for 16 baselines. For a few exceptions, the difference is significant:

- OODEEL performs significantly better for Gram (three datasets), ODIN (CIFAR-10), and SHE (CIFAR-10)
- OpenOOD performs significantly better for RMDS (CIFAR-10)

Except for Gram, when there is such a large discrepancy, it is only for one dataset. It provides confidence in the correctness and robustness of our implementation.

Table 1: Comparison of **near-OOD** detection AUROC between OpenOOD and OODEEL. We used pretrained ResNet-18 for CIFAR-10 and CIFAR-100, and ResNet-50 for ImageNet.

Method	**CIFAR-10**		**CIFAR-100**		**ImageNet**	
	OpenOOD	**OODEEL**	**OpenOOD**	**OODEEL**	**OpenOOD**	**OODEEL**
MLS	87.52	87.89	81.05	79.89	76.46	76.46
MSP	88.03	88.30	80.27	79.05	76.02	76.02
ODIN	82.87	87.72	79.90	79.72	74.75	75.73
GEN	88.20	89.15	81.31	80.13	76.85	77.71
SHE	81.54	83.16	78.95	77.86	73.78	72.61
Energy	87.58	87.98	80.91	79.80	75.89	75.89
ReAct	87.11	87.93	80.77	79.75	77.38	77.39
SCALE	82.55	82.48	80.99	79.78	81.36	81.36
ASH	75.27	79.99	78.20	79.61	78.17	79.63
DKNN	90.64	91.02	80.18	79.92	71.10	72.24
Mahalanobis	84.20	84.91	58.69	59.43	55.44	55.83
RMDS	89.80	86.82	80.15	79.22	76.99	76.32
VIM	88.68	87.34	74.98	74.30	72.08	72.64
Gram	58.66	88.73	51.66	70.50	61.70	70.72