

The BRAVO Semantic Segmentation Challenge Results in UNCV2024

Tuan-Hung Vu¹, Eduardo Valle¹, Andrei Bursuc¹, Tommie Keressies², Daan de Geus², Gijs Dubbelman², Long Qian^{3,4}, Bingke Zhu^{3,5}, Yingying Chen^{3,5}, Ming Tang^{3,4}, Jinqiao Wang^{3,4,5}, Tomáš Vojtíš⁶, Jan Šochman⁶, Jiří Matas⁶, Michael Smith⁷, Frank Ferrie⁷, Shamik Basu⁸, Christos Sakaridis¹⁰, and Luc Van Gool^{9,10}

¹ valeo.ai, France (Challenge Organizers)

² Eindhoven University of Technology, Netherlands

³ Chinese Academy of Sciences, China

⁴ University of Chinese Academy of Sciences, China

⁵ Objecteye Inc., China

⁶ Czech Technical University in Prague, Czechia

⁷ McGill University, Canada

⁸ University of Bologna, Italy

⁹ Institute for Computer Science, Artificial Intelligence and Technology, Bulgaria

¹⁰ ETH Zurich, Switzerland

Abstract. We propose the unified BRAVO challenge to benchmark the reliability of semantic segmentation models under realistic perturbations and unknown out-of-distribution (OOD) scenarios. We define two categories of reliability: (1) semantic reliability, which reflects the model’s accuracy and calibration when exposed to various perturbations; and (2) OOD reliability, which measures the model’s ability to detect object classes that are unknown during training. The challenge attracted nearly 100 submissions from international teams representing notable research institutions. The results reveal interesting insights into the importance of large-scale pre-training and minimal architectural design in developing robust and reliable semantic segmentation models.

1 BRAVO Challenge

Autonomous vehicles are safety-critical systems operating in a complex open world. As such, they must not only deliver excellent performance in their operational design domain but also be provably robust to adversarial attacks, extreme weather conditions, domain changes, and rare but potentially catastrophic driving situations. The BRAVO Challenge aims to develop test beds to assess and statistically demonstrate the robustness of driving perception models. The challenge employs existing test sets, sometimes with added synthetic augmentations, with novel test metrics to emphasize safety-centered challenges: calibration of models’ outputs and estimation of their uncertainty; detection of out-of-distribution inputs, at scene or object level; assessment of domain shifts. The

BRAVO Challenge 2024 focused on semantic segmentation and was presented at the UNCV workshop ¹¹.

1.1 Main Tracks

In the BRAVO Challenge 2024, we proposed two tracks:

- **Track 1: single-domain Training.** Participants must train their models exclusively on the Cityscapes [6] dataset. This track evaluates the robustness of models trained with limited supervision and geographical diversity when facing unexpected corruptions observed in real-world scenarios.
- **Track 2: multi-domain Training.** Participants may train their models over a mix of datasets, whose choice is strictly limited to the list provided below, comprising both natural and synthetic domains. This track assesses the impact of fewer constraints on the training data on robustness. The accepted datasets are: Cityscapes [6], BDD100K [31], Mapillary Vistas [17], India Driving Dataset [25], WildDash 2 [32], GTA5 [20] and SHIFT [22].

1.2 BRAVO Dataset

The BRAVO Challenge 2024 aimed to benchmark semantic segmentation models on urban scenes undergoing diverse forms of natural degradation and realistic-looking synthetic corruptions. To this end, we repurposed existing datasets [3, 11, 21] and combined them with newly generated data. The BRAVO Dataset 2024 comprised images from ACDC [21], SegmentMeIfYouCan (SMIYC) [3], Out-of-context Cityscapes [11], and new synthetic data. We organized the dataset into six subsets, two with real data and four based on the validation set of Cityscapes with synthetic augmentations:

- *bravo-ACDC*: real scenes captured in adverse weather conditions, *i.e.*, fog, night, rain, and snow [21];
- *bravo-SMIYC*: real scenes featuring out-of-distribution (OOD) objects rarely encountered on the road [3];
- *bravo-synrain*: 500 augmented scenes with synthesized raindrops on the camera lens [19];
- *bravo-synobjs*: 656 augmented scenes with inpainted synthetic OOD objects from 26 classes [14];
- *bravo-synflare*: 308 augmented scenes with synthesized light flares [29];
- *bravo-outofcontext*: 329 augmented scenes with random backgrounds for road and sidewalk [11].

¹¹ <https://uncertainty-cv.github.io/2024/>

1.3 Metrics

The BRAVO Challenge 2024 evaluated methods on various metrics to assess their performance in semantic segmentation and out-of-distribution (OOD) detection. The semantic metrics assessed the quality of the semantic segmentation predictions on both accuracy and calibration. The OOD metrics assess the model’s ability to detect whether the objects are OOD, *i.e.*, to distinguish between *known* classes seen during training *vs.* *unknown* classes seen at test time. The BRAVO Index combines the semantic and OOD metrics to rank the models.

Semantic metrics are computed on all subsets, except SMIYC, for valid pixels only. Valid pixels are those not invalidated by extreme uncertainty, such as pixels obscured by the brightest areas of a flare or covered by an OOD object.

- Mean Intersection over Union (mIoU): Proportion of correctly labeled pixels among all pixels. Only semantic metric that does not rely on prediction confidence. Higher values indicate better segmentation accuracy.
- Expected Calibration Error (ECE): Difference between predicted confidence and actual accuracy. Lower values indicate better calibration.
- Area Under the ROC Curve (AUROC): Area Under the ROC Curve over the binary criterion of a pixel being accurate, ranked by the predicted confidence level for the pixel. Higher values indicate better calibration, as the confidence ranking matches the correctness of the pixels.
- False Positive Rate at 95% True Positive Rate (FPR@95): False positive rate when the true positive rate is 95% in the ROC curve above. Lower values indicate better calibration at the tail of the confidence distribution: the ability to reject false positives even when we reach the most true positives.
- AUPR-Success: Area Under Precision-Recall curve, over the same data as the AUROC. Higher values indicate the ability of higher confidence to match correct pixels and, thus, better calibration.
- AUPR-Error: Uses the reversed data (pixel being inaccurate, ranked by $1 - \text{confidence}$). Higher values indicate the ability of lower confidence to match incorrect pixels and, thus, better calibration. That tends to be stricter than AUPR-Success, since incorrect pixels tend to be rarer.

OOD metrics are computed on the SMIYC and SynObjs subsets only for invalid pixels, *i.e.*, those obscured by OOD objects. The OOD metrics are computed over the binary criterion of a pixel being invalid, ranked by the reversed predicted confidence level for the pixel, and include:

- Area Under the ROC Curve (AUROC).
- False Positive Rate at 95% True Positive Rate (FPR@95).
- Area Under the Precision-Recall Curve (AUPRC).

Aggregated metrics. For model ranking, the metrics above are aggregated as follows:

- Semantic. The harmonic mean of all semantic metrics, with ECE and FPR@95 reversed (as $1 - x$).

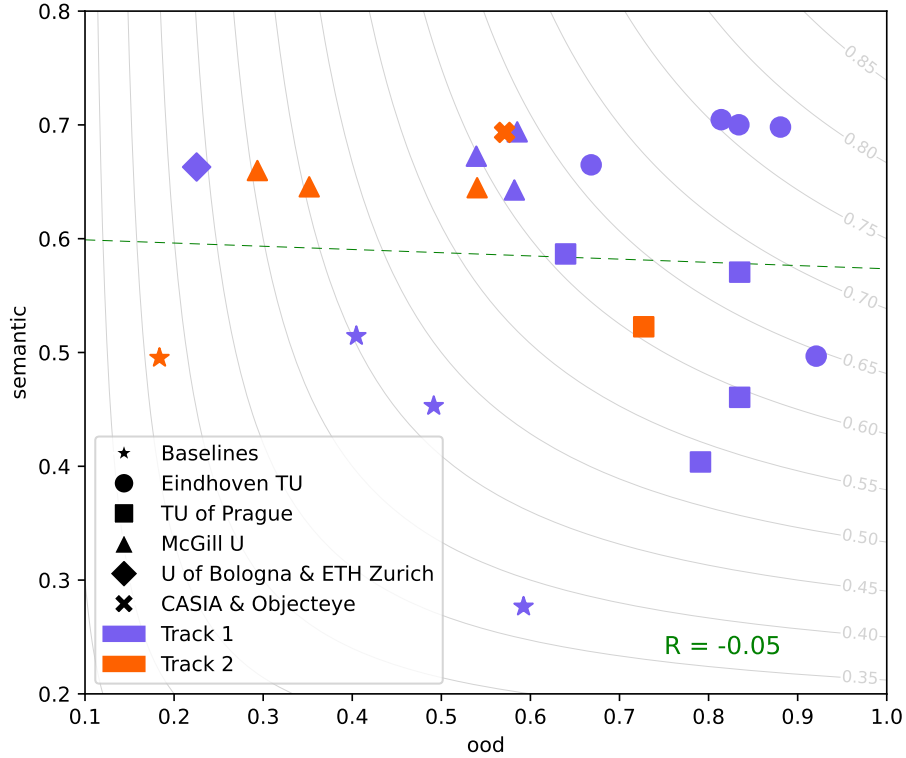


Fig. 1: All submissions. Aggregated metrics (out-of-distribution and semantic) on axes, ranking metric (BRAVO Index) on level set. More freedom on the training dataset (Task 2, in orange) did not translate into better results.

- OOD. The harmonic mean of all OOD metrics, with FPR@95 reversed.
- BRAVO Index: The harmonic mean of Semantic and OOD, used as the official ranking metric for the challenge.

In Appendix A, we provide more information on the rules and the submission format of the BRAVO challenge 2024.

2 Submissions digest

This section collects all the solutions from the two challenge tracks, along with interesting findings reported by the participants. We primarily retain the notation used by the participants in their reports. Notations are, thus, not consistent across subsections. The first person “we” in the submission subsections is that of the respective authors, summarized and sometimes paraphrased by the challenge organizers. Due to space limitations, the approach figures and some training details are provided in Appendix B.

Method	Ref.	BRAVO \uparrow	Semantic \uparrow	OOD \uparrow
DINOv2-OOD	Sec. 2.2	77.9	69.8	88.1
PixOOD w/ ResNet-101 DeepLabv3 [26]	Sec. 2.3	61.2	58.7	64.0
Ensemble	Sec. 2.4	61.1	64.3	58.2
PhyFea [1]	Sec. 2.5	33.6	66.3	22.5
<i>Baseline: SegFormer-B5 [30]</i>	-	47.1	45.3	49.2
<i>Baseline: ObsNet-ResNet101 [2]</i>	-	45.3	51.5	40.5
<i>Baseline: RbA Swin-B [16]</i>	-	37.7	27.7	59.2

Table 1: Track 1 – Performance metrics across different approaches.

Method	Ref.	BRAVO \uparrow	Semantic \uparrow	OOD \uparrow
InternImage-OOD	Sec. 2.6	62.6	69.3	57.1
PixOOD [26]	Sec. 2.7	60.8	52.3	72.7
Ensemble	Sec. 2.8	58.8	64.5	54.0
<i>Baseline: FAMix [10]</i>	-	26.8	49.5	18.4

Table 2: Track 2 – Performance metrics across different approaches.

2.1 Quantitative summary

The results are summarized¹² in Tabs. 1 and 2, which show the best submission for each team. Fig. 1 shows all public submissions at the end of the Challenge.

Fig. 1 shows the submissions considerably improved over the proposed baselines. Due to the strictness of the harmonic mean, submissions that only enhanced one criterion were penalized on the ranking BRAVO Index. The most surprising collective finding is that multiple training datasets (Track 2) did not improve the metrics using only Cityscapes (Track 1).

The OOD and semantic metrics were uncorrelated among the submissions (regression line in green in Fig. 1, $R = -0.05$). On the other hand, we observed varying degrees of correlation among the metrics aggregated by the subsets listed in Sec. 1.2. The correlogram appears in the Appendix B.

All top two solutions in both tracks leverage vision foundation models (VFMs) pretrained on massive data corpus, *i.e.*, DINOv2 [18] and InternImage [27]. For semantic segmentation, it was surprising to find that a simple linear decoder (DINOv2-OOD) outperformed more sophisticated decoders by a large margin, according to the unified BRAVO score.

Those findings emphasize the importance of robust VFM backbone and may temporarily shift researchers’ focus toward training more robust VFMs, rather than concentrating on sophisticated network design for downstream tasks. However, the current best BRAVO score is still far from perfect and using larger models is not showing clear beneficial signals, we believe that advancing reliable segmentation requires progress from both sides: developing more robust VFM backbones and designing more efficient architectures to better exploit the knowledge encapsulated in pre-trained VFMs.

¹² Detailed results are available at the challenge server and at the repository: https://github.com/valeoai/bravo_challenge.

2.2 Track 1: DINOv2-OOD – Eindhoven University of Technology

Authors: Tommie Kerssies, Daan de Geus, Gijs Dubbelman

This solution fine-tunes pre-trained Vision Foundation Models (VFMs) for semantic segmentation, leveraging their robust representations. Given a pre-trained VFM, we attach an off-the-shelf segmentation decoder and fine-tune the entire model for semantic segmentation. We evaluate this meta-architecture in several different configurations. Our primary solution uses the DINOv2 VFM [18], selected due to its effectiveness in domain generalized semantic segmentation for urban scenes [9, 13]. DINOv2, built upon the Vision Transformer (ViT) architecture [8], is pre-trained using self-supervised learning on a vast, curated dataset. We experiment with all available sizes of DINOv2.

For our default segmentation decoder, we use a simple linear layer that transforms the patch-level features $\mathbf{F} \in \mathbb{R}^{E \times \frac{H}{P} \times \frac{W}{P}}$ into segmentation logits $\mathbf{L} \in \mathbb{R}^{C \times \frac{H}{P} \times \frac{W}{P}}$, where H and W represent the height and width of the input image, P denotes the patch size, E is the feature dimension, and C is the number of classes in the dataset. We choose a linear layer trusting that the strong representations learned by the VFM forgo a more advanced decoder (which could also overfit to the training distribution).

We evaluate the impact of large-scale pre-training with DINOv2 by contrast with a DeiT-III [24] ViT pre-trained on ImageNet-1K [7] and fine-tuned on Cityscapes. We assess the impact of a more advanced decoder by contrast with a Mask2Former decoder [5]. We also contrast the default patch size (16×16) to a more expensive 8×8 .

Training. When training the model with a linear decoder, we bilinearly up-sample the segmentation logits $\mathbf{L} \in \mathbb{R}^{C \times \frac{H}{P} \times \frac{W}{P}}$ to $\mathbf{L}' \in \mathbb{R}^{C \times H \times W}$, and then apply a categorical cross-entropy loss to those logits and the semantic segmentation ground truth to fine-tune the model. When using Mask2Former, the decoder outputs a set of mask logits $\mathbf{M} \in \mathbb{R}^{N \times \frac{H}{P} \times \frac{W}{P}}$ and corresponding class logits $\mathbf{C} \in \mathbb{R}^{N \times (C+1)}$, where N is the number of masks and C includes an additional “no-object” class. Following Mask2Former, during training those mask and class logits are matched to the ground truth using bipartite matching. The predicted masks are then supervised with a cross-entropy loss and a Dice loss, and the predicted classes are supervised with a categorical cross-entropy loss.

Testing. During inference with the linear decoder, we compute per-pixel class confidence scores with softmax on the upsampled class logits \mathbf{L}' , using the highest score to predict the pixel class. For the Mask2Former decoder [5], we bilinearly up-sample the mask logits $\mathbf{M} \in \mathbb{R}^{N \times \frac{H}{P} \times \frac{W}{P}}$ to the original resolution, resulting in $\mathbf{M}' \in \mathbb{R}^{C \times H \times W}$. We then obtain the mask scores $\mathbf{P}_{\mathbf{M}} = \text{sigmoid}(\mathbf{M}')$.

Overall per-pixel class confidence scores $\mathbf{P}' \in \mathbb{R}^{C \times H \times W}$ are computed by multiplying the mask scores with the class scores across all masks. Specifically, for each class c and pixel (h, w) , we have: $\mathbf{P}'_{c,h,w} = \sum_{n=1}^N \mathbf{P}_{\mathbf{M}_n,h,w} \cdot \mathbf{P}_{\mathbf{C}_n,c}$. For each pixel, the predicted class is the one with the highest value in \mathbf{P}' , and we also output this maximum value as the confidence score.

Results. As shown in Tab. 3. Our best performing model, DINOv2 with a ViT-L/8 backbone and a linear decoder, achieves the highest BRAVO index of

Method	BRAVO \uparrow	Semantic \uparrow	OOD \uparrow
DINOv2, ViT-L, 8x8 patch size, linear decoder	77.9	69.8	88.1
DINOv2, ViT-L, 16x16 patch size, linear decoder	77.2	70.8	84.8
DINOv2, ViT-g, 16x16 patch size, linear decoder	76.1	70.0	83.4
DINOv2, ViT-B, 16x16 patch size, linear decoder	75.5	70.5	81.4
DINOv2, ViT-S, 16x16 patch size, linear decoder	69.9	69.1	70.6
DINOv2, ViT-g, 16x16 patch size, Mask2Former decoder	64.5	49.7	92.1
DeiT III (IN1K), ViT-S, 16x16 patch size, linear decoder	54.1	62.8	47.6

Table 3: Track 1 – DINOv2-OOD – Ablated models.

Method	mIoU \uparrow	AUPR-Error \uparrow	AUPR-Success \uparrow	AUROC \uparrow	ECE \downarrow	FPR@95 \downarrow
DINOv2, ViT-L, 8x8 patch size, linear decoder	76.7	40.0	99.4	91.4	2.0	38.8
DINOv2, ViT-L, 16x16 patch size, linear decoder	75.9	41.2	99.5	92.3	1.7	37.8
DINOv2, ViT-g, 16x16 patch size, linear decoder	77.6	39.3	99.5	92.3	1.8	37.6
DINOv2, ViT-B, 16x16 patch size, linear decoder	71.7	43.3	99.4	92.3	2.1	40.3
DINOv2, ViT-S, 16x16 patch size, linear decoder	66.5	45.1	99.2	91.8	2.5	44.3
DINOv2, ViT-g, 16x16 patch size, Mask2Former decoder	78.2	23.2	99.2	87.9	5.0	63.6
FixOOD w/ ResNet-101 DeepLabv3 [26]	43.2	58.5	93.5	84.0	15.1	54.6
Ensemble C	73.9	47.4	99.1	92.5	32.7	34.7
DeiT III (IN1K), ViT-S, 16x16 patch size, linear decoder	44.9	54.7	97.9	89.2	1.7	54.2
<i>Baseline:</i> SegFormer-B5 [30]	67.4	24.1	97.2	77.2	30.7	71.9
<i>Baseline:</i> ObsNet-R101-DLV3plus [2]	65.3	32.1	98.5	87.8	45.6	63.4
<i>Baseline:</i> Mask2Former-SwinB [5]	67.2	13.2	90.4	47.0	55.1	82.8

Table 4: Track 1 – DINOv2-OOD – Semantic metrics for valid pixel predictions and their confidence, averaged across all subsets except SMIYC, computed for ablated models and other approaches.

77.9, which is +16.7 higher than the next best method, *i.e.*, PixelOOD presented in Sec. 2.3.

We observe in Tab. 4 that the model with the highest mIoU, DINOv2 with a ViT-g/16 backbone and a Mask2Former decoder, performs relatively poorly on the other metrics. As the other metrics take into account the confidence score, this suggests that while Mask2Former is effective at predicting the correct class, it is less adept at estimating the confidence of its predictions, at least in the out-of-the-box manner in which we used it. In our setup, models with a simple linear decoder provide the best trade-off between segmentation accuracy and confidence estimation.

A similar result is observed when changing the patch size. A smaller patch size of 8×8 results in better mIoU, but the other metrics are worse. This indicates that a smaller patch size allows the model to capture more fine-grained details, which improves the accuracy of the predicted class labels, but that this somehow makes the confidence scores less reliable.

Another noteworthy observation is that all our models have relatively low ECE values, indicating that they are well-calibrated, even though no explicit calibration techniques were applied. Even the DeiT-III-based model, which scores low on the overall BRAVO score, achieves a low ECE value of 1.7. Therefore, further investigation is needed to understand why the ECE values are so low.

Finally, the results suggest that more accurate models in terms of mIoU tend to be worse at identifying their own errors, as indicated by the AUPR-Error metric. However, they excel at identifying correct predictions, as shown by the AUPR-Success metric. It is possible that this happens simply because errors by accurate models are rarer, making it harder to identify them.

Method	AUPRC \uparrow	AUROC \uparrow	FPR@95 \downarrow
DINOv2, ViT-L, 8x8 patch size, linear decoder	81.7	97.7	12.9
DINOv2, ViT-L, 16x16 patch size, linear decoder	76.7	97.1	15.0
DINOv2, ViT-g, 16x16 patch size, linear decoder	74.3	96.9	15.3
DINOv2, ViT-B, 16x16 patch size, linear decoder	70.6	96.6	15.1
DINOv2, ViT-S, 16x16 patch size, linear decoder	58.9	94.9	20.2
DINOv2, ViT-g, 16x16 patch size, Mask2Former decoder	84.1	98.8	4.5
DeiT III (IN1K), ViT-S, 16x16 patch size, linear decoder	30.0	86.5	38.6

Table 5: Track 1 – DINOv2-ODD – OOD metrics for detecting OOD objects by identifying invalid pixels based on prediction confidence, averaged over the SMIYC and Synobj subsets, computed for ablated models.

Overall, the results show that mIoU, which does not depend on prediction confidence, does not correlate well with the other metrics that do.

OOD results are detailed in Tab. 5. Surprisingly, our configuration with worst confidence estimation for valid pixels, DINOv2 with ViT-g/16 and Mask2Former, achieves the highest AUPRC of 84.1, the highest AUROC of 98.8, and the lowest FPR@95 of 4.5 for detecting invalid pixels. This suggests that the mask classification framework used by Mask2Former, where per-class masks are predicted separately, allows this decoder to more accurately identify which pixels belong to the mask of an in-distribution class and which do not. Additionally, while a smaller patch size results in worse confidence estimation for valid pixels (see Fig. 3), it helps in identifying invalid pixels. Qualitative analyses show that the smaller patch size enables the model to better separate valid and invalid pixels, as it can capture more fine-grained details. Finally, while scaling from ViT-L to ViT-g improves mIoU for valid pixels (see Tab. 4), OOD detection performance shows a noticeable degradation.

Overall, the results indicate that the models best at identifying invalid pixels are not necessarily the same ones that excel at correctly classifying valid pixels or accurately estimating their confidence for valid pixels

2.3 Track 1: PixOOD – Czech Technical University in Prague

Authors: Tomáš Vojtíš, Jan Šochman and Jiří Matas

We describe how the semantic segmentation and the confidence scores are computed for all submitted methods. We also discuss the training details with the focus on differences to the original PixOOD [26].

Semantic Segmentation. The semantic class $c \in \{1, 2, \dots, C\}$ for each pixel $p \in \{(y, x)\}^{H \times W}$ of an image $I \in \mathbb{R}^{H \times W \times 3}$ is computed from logits $l \in \mathbb{R}^{H \times W \times C}$ simply as: $c_p^* = \arg \max_c (l_p^c)$. The logits are computed using different decoders in each variant as discussed below.

Confidence. The confidence of the semantic segmentation (*i.e.* 1 – OOD score) in all variants of the PixOOD method is computed as s_I by Eq. (3) from PixOOD [26]. Because of the quantization required for saving the results to a 16-bit PNG format (*i.e.*, into the 65,536 values), the score is re-normalized so that the “effective range” of the score is well represented. Since the score is calibrated and directly corresponds to the false positive rate of the in-distribution

Method	BRAVO [†]	Semantic [†]	OOD [†]
PixOOD	53.5	40.4	79.1
PixOOD w/ DeepLabv3 Decoder	59.4	46.1	83.5
PixOOD w/ ResNet-101 DeepLabv3	61.2	58.7	64.0

Table 6: Track 1 – PixOOD – Analysis.

data (training data) the “effective range” is mostly limited to the first 5% – *i.e.*, (0.0, 0.05), thus, the re-normalization maps the score piece-wise linearly as follows: $[0.0, 0.05] \rightarrow [0.0, 0.8]$ and $[0.05, 1.0] \rightarrow [0.8, 1.0]$ where the square brackets denote inclusion of the boundary in the range. Note that this re-normalization is only useful if we need to quantize the results for some reason. The calculation of the s_I score in PixOOD is performed on a per-class basis, we report the score associated with the predicted class c_p^* .

Models. The following three PixOOD variants were submitted to the challenge.

- *PixOOD*. This is the default method exactly as described in the PixOOD paper with the checkpoints used from the published codebase.
- *PixOOD w/ DeepLabv3 Decoder*. This method replaces the simple MLP segmentation head in PixOOD by a more complex DeepLab v3 [4] decoder. The input to the decoder are concatenated features from layers (17, 23) and (5, 11) of DINOv2 [18] encoder for the *ASPP* and *fine* bottleneck layers respectively. The output of the decoder is used to generate the logits for the N-P task and the computation of the OOD and segmentation scores.
- *PixOOD w/ ResNet-101 DeepLabv3*. This method replaces the simple MLP segmentation head in PixOOD by a complete DeepLab v3 network [4] with a ResNet-101 [12] backbone. It takes an image as input instead of the latent representation of the DINOv2 [18] image encoder used in the previous two methods. The DeepLab v3 output logits (*i.e.*, output of the last layer of the network before any **argmax** operation) are used for the N-P task and the computation of the OOD and segmentation scores.

Training. All variants of the segmentation part of the PixOOD method were trained using the same regime for 30 epochs on the Cityscapes dataset with the learning rate set to 0.0001 using the AdamW optimizer without scheduling the learning rate. The submitted variants used basic image augmentations, *i.e.*, random crop of size 1,792 of the longer side while keeping the aspect ratio and random horizontal flip with probability 0.5. These augmentations were used only during training of the head that produces the logits. The calculation of the Condensation algorithm and the N-P decision strategies were the same for all methods and follows the default settings described in PixOOD [26].

2.4 Track 1: Ensemble – McGill University

Authors: Michael Smith and Frank Ferrie

The solutions to both tracks involve ensembles, albeit in different configurations. For both of them, we use ensembles in a standard configuration where

Method	BRAVO↑	Semantic↑	OOD↑
Ensemble A	59.9	67.3	53.9
Ensemble C	61.1	64.3	58.2
HMSA	36.0	70.6	24.2

Table 7: Track 1 – Ensemble – Analysis.

we have Q models [15]: $\{P(y | x^*, \theta^{(q)})\}_{q=1}^Q$, $\theta^{(q)} \sim p(\theta | \mathcal{D})$. Each one of these models is capable of generating a prediction y from a test input x^* with weights $\theta^{(q)}$ constrained by the prior $p(\theta)$. The predictions of these models can be aggregated through the predictive posterior as the mean across models, *i.e.*: $P(y | x^*, \mathcal{D}) = \frac{1}{Q} \sum_{q=1}^Q P(y | x^*, \theta^{(q)})$. With $P(y | x^*, \mathcal{D})$, we now have a confidence assigned to each class, with the maximum across the set of classes providing the predicted class and associated confidence in the prediction as required by the BRAVO challenge.

For Track 1, we address both aspects of our hypothesis. For the first part on model diversity, we chose to use ensembles of two models: Mask2Former [5] and HMSA (Hierarchical Multi-Scale Attention for Semantic Segmentation) [23]. In all cases for this track, pretrained Cityscapes models available online are used. Note that for the Mask2Former approach, we build off the code and use models provided by the authors of RbA (Rejected By All) [16], whose main contribution is a scoring function that takes as input the output of a Mask2Former model. We explicitly denote the approaches where we use said scoring function as being RbA, but otherwise refer to the approach as Mask2Former. Ideally, we would have used more models, but this was not possible due to time and resource constraints. The two models were chosen as they have very different architectures, and thus are good candidates for exploring model diversity in terms of architectures. They also satisfy the other part of our hypothesis as they are known to perform very well in terms of semantic segmentation performance and out-of-distribution performance, on the Cityscapes [6] and SMIYC [3] leaderboards, providing a good starting point in terms of performance.

Models. Ensemble Configuration A & C involve combining the predictions of Mask2Former [5] and HMSA [23]. Configuration A consists of two models: the Swin-L model of Mask2Former and the `nimble-chihuahua` model from HMSA. Configuration C adds one additional model in the form of the Swin-B model for Mask2Former, for a total of three models. We apply the softmax operator to the logits of each model independently, which then gives us $Q = \{2, 3\}$ models, giving us a prediction and associated confidence in each class for every pixel.

Given the results presented in Tab. 7, we can make a few observations. The first is that straightforward ensembles using the mean achieves very respectable results, doing relatively well in combining the divergent semantic and OOD performance of the Mask2Former and HMSA models. It is not quite able however to achieve the best of both worlds, as demonstrated by the greater performance of the RbA and HMSA baselines. It is clear as well that other approaches, such as PixOOD (Sec. 2.3), suffer from some trade-offs in terms of semantic and OOD performance when making architecture changes. In this particular case, it is safe

to say that network architecture can play a very significant role in performance, and the performance of ensembles can be heavily influenced by it. Our second observation is that it is clear model diversity can play an important role as well, as the performance of our approach is entirely due to being able to combine two approaches that specialize in different metrics. However, with only two models at play for Track 1, we cannot make any definitive statements.

2.5 Track 1: PhyFea – University Of Bologna & ETH Zurich

Authors: Shamik Basu, Christos Sakaridis and Luc Van Gool

Our approach PhyFea (Physically Feasible Semantic Segmentation) enhances the performance of the baseline segmentation architectures $\phi(X)$ as described in our paper by retraining it with physical priors “inclusion constraint” and “discontinued class” incorporated by PhyFea. After retraining, we can observe an improvement in the mIOU score during inference by the baseline architectures. For this challenge, the baseline architecture we have taken is Segformer-B4 [30]. The training overview is explained in [1].

Semantic segmentation. The semantic class $c \in \{1, 2, 3, \dots, C\}$ for each pixel $p \in (y, x)^{H \times W}$ of an image $I \in \mathbb{R}^{(3,H,W)}$ is computed from logits $\phi(I) \in \mathbb{R}^{(C,H,W)}$ as: $S^* = \arg \max_s(\phi(I))$.

Confidence. The confidence of the semantic segmentation (*i.e.*, 1 – OOD score) in the baseline model $\phi(X)$ is computed on its output $\phi(I)$. First, $\phi(I)$ is bounded as $0 \leq \phi(I) \leq 1$ in order to represent the effective range in a better way. Then, quantization is performed and saved in 16-bit PNG format.

Training. In [1], we show the architectural overview of PhyFea. A 2D semantic segmentation model as baseline network $\phi(X)$ takes an image $\mathbf{I} \in \mathbb{R}^{(3,H,W)}$ as input and produces the raw output $\phi(\mathbf{I}) \in \mathbb{R}^{(C,H,W)}$ where C is the number of classes present in the dataset. PhyFea takes $\phi(\mathbf{I})$ as input and produces an absolute difference of two loss values $l_{opening}$ and $l_{dilation}$ generated by the two operations performed in PhyFea, namely opening and selective dilation. Opening solves the inclusion constraint problem and selective dilation solves the discontinued class problem. The absolute difference $|l_{opening} - l_{dilation}|$ is then added to the cross-entropy loss (denoted by $l_{cross-entropy}$) of $\phi(X)$ to obtain the total loss. Here α is a hyperparameter and it is used to balance the loss of PhyFea and the baseline network. $l_{total-loss}$ is backpropagated to optimize the weights of the baseline network by obtaining the **argmin** of $l_{total-loss}$ as: $l_{total-loss} = l_{cross-entropy} + \alpha * |l_{opening} - l_{dilation}|$, $0 < \alpha < 1$ and $S^* = \arg \min_s(l_{total-loss})$. PhyFea is end-to-end differentiable in order to incorporate the physical priors (*i.e.*, inclusion constraint and discontinued class) while re-training the baseline network and it is free of any parameterized component like convolution kernel or MLP.

2.6 Track 2: InternImage-OOD – CASIA & Objecteye

Authors: Long Qian, Bingke Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang

We introduce InternImage-OOD, which integrate the power of general large-scale vision foundation model and the efficiency of simple clustering algorithm, to solve the challenge. We fine-tuned the vision foundation model to enhance its semantic segmentation capabilities, and integrated a simple clustering algorithm to improve the model’s OOD detection performance, all while maintaining efficiency. We employ K-Means clustering as a post-processing technique to improve OOD detection followed by PixOOD [26] (Sec. 2.3 and Sec. 2.7). Our method consists of two stages. First, the input image is processed using the InternImage model to perform basic semantic segmentation, resulting in the predicted class map P , the confidence map C , and the image feature embeddings F . Then, we apply a K-Means-based OOD detection method for post-processing the results.

As the core of the pipeline, we apply the InternImage to perform basic semantic segmentation. $P = \{p_i \in \mathbb{R}^{H \times W}\}_{i=1}^K = \text{InternImage}(I)$, $C = \{c_j \in \mathbb{R}^{H \times W}\}_{j=1}^K = \text{Confidence}(I)$ and $F = \{f_k \in \mathbb{R}^{D \times H \times W}\}_{k=1}^M = \text{FeatureEmbedding}(I)$ where: P is the set of predicted pixel-wise class labels from the InternImage model, C is the set of confidence maps associated with each class prediction, F is the set of feature embeddings extracted from the feature pyramid, H and W represent the height and width of the image, respectively, K denotes the number of classes, D is the dimensionality of the feature embedding for each pixel, and M represents the number of feature levels in the feature pyramid.

In the second stage, K-Means clustering is applied to the feature embeddings for OOD detection and further refinement, with $OODMask = \text{K-Means}(F_k)$ and $UpdatedConfidence(C) = \text{UpdateMask}(OODMask, C)$. $OODMask$ is the mask, which shows the region of OOD, obtained from K-Means clustering applied to feature embeddings. UpdateMask is the function that updates the confidence map C based on the OOD mask.

Here, *feature embeddings* are taken from different depths of the model, representing the model from shallow to deep features. For each class in the *predicted class map* P , where regions are recognized as low-confidence regions, the corresponding *feature embeddings* F are passed to the trained KMeans models. By calculating the distances between the embeddings and the cluster centroids, OOD regions are identified based on how far they deviate from the known clusters. Regions with distances significantly higher than the average are marked as OOD, and the corresponding areas in the *confidence map* C are updated with lower values to reflect the uncertainty.

Datasets. Considering the time constraints of the competition and in order to demonstrate the advantages of this method to still achieve excellent performance on a small number of datasets, we only used Mapillary Vistas and Cityscapes datasets to complete this experiment, which have about 50,000 images in total.

Implementation Details. We choose the *Upernet_InternImage_XL* [27] as our backbone, and train the model on Mapillary Vistas and Cityscapes datasets one by one, during which we choose the AdamW optimizer with a learning rate of $2e-5$, a batch size of 8, and a weight decay of 0.05. Besides, we use K-Means

Method	BRAVO↑	Semantic↑	OOD↑
InternImage	62.1	69.3	56.2
+ KMeans-Based OOD	62.6	69.3	57.1

Table 8: Track 2 – InternImage-OOD – Ablation Study.

as our optimization method on OOD, in which we only use a few images and a very small number of the cluster centroids, considering the limited time.

Results. Our solution achieved BRAVO-Index of 62.6 (see Tab. 2), and ranked 1st in the Multi-domain training Track. To show the effectiveness of our solution, we conduct an ablation study. As show in Tab. 8, we achieved a modest improvement with a very small amount of data and clustering centers.

2.7 Track 2: PixOOD – Czech Technical University in Prague

Authors: Tomáš Vojtř, Jan Šochman and Jiří Matas

We refer to Sec. 2.3 for details of the method and different model variants. For Track 2, only the *PixOOD w/ DeepLabv3 Decoder* variant is submitted (see Tab. 2). The method was trained on the combined Cityscapes and BDD100K datasets. The BDD100K data set was randomly sub-sampled such that the number of images is roughly equal (taking 1/3 of the data) to the size of Cityscapes. As the resolution of these two datasets is different, a smaller random crop of size 1036 (of the longer side while keeping the aspect ratio) was used during training.

2.8 Track 2: Ensemble – McGill University

Authors: Michael Smith and Frank Ferrie

We refer to Sec. 2.4 for the common theory. Here we present the methodology differences adopted for Track 2 and the corresponding results.

For Track 2, our primary goal was to evaluate the potential use of different datasets as a source of model diversity for the ensembles. The BRAVO challenge is set up to evaluate only the standard 19 Cityscapes evaluation classes, and Track 2 allows for the use of multiple datasets. With all of these datasets placing a clear focus on autonomous driving in some way, they are all formatted to either use the aforementioned 19 Cityscapes classes or use classes similar enough such that they can be mapped to the Cityscapes ones. This presents an opportunity where we can train models on different datasets with different characteristics (including some synthetic ones) while maintaining compatibility with one another. Here, we use the same models as Track 1, with a particular focus on HMSA as we train a model with that architecture for each allowed dataset.

Training. Before we could evaluate any models, we first needed to obtain one trained model per dataset with the HMSA architecture [23]. The models for each were generated as follows:

- *Cityscapes:* We used the author-provided `nimble-chihuahua` model [23].

Method	BRAVO↑	Semantic↑	OOD↑
Ensemble A	40.6	66.0	29.4
Ensemble B	45.5	64.6	35.2
Ensemble C	58.8	64.5	54.0

Table 9: Track 2 – Ensemble – Analysis.

- *Mapillary*: After converting the dataset to use Cityscapes labels, we trained the model using transfer learning from the `fast-rattlesnake` model [23] as provided by the authors and with the same training configuration as they provide with the code for their Mapillary model, except the number of epochs, which we set to 15 as we needed to adapt the model to the new class scheme.
- *GTA5*: We first removed some corrupted images and then resized all images and ground truth masks to (1914, 1052). The model was then trained with transfer learning from the `fast-rattlesnake` model, with the same training settings as used by [23] for their `cityscapes_sota` training configuration.
- *SHIFT*: We used the dataset author-provided label mapping to Cityscapes and trained the model with transfer learning from the `outstanding-turtle` model from [23]. The training parameters are the same as with the GTA5 model, except for the learning rate, which is set to 0.005.
- *BDD100K*: This model is trained via transfer learning from the `industrious-chicken` model [23] with the same settings as the GTA5 model.
- *IDD*: This transfer learning source for this model is the `outstanding-turtle` model. Parameters are the same as the GTA5 model.
- *Wild Dash 2*: The `nimble-chihuahua` model is used for pretraining in this case, with parameters the same as the SHIFT model.

Tab. 2 and Tab. 9 reports results in Track 2. We can see that while using several models helps the BRAVO score for Multi-dataset ensemble configuration A over the HMSA baseline, it does so by improving the OOD score at the expense of the semantic score. Multi-dataset configurations B and C, however, show that adding the two Mask2Former models, with their ability to do better at OOD detection, is much more impactful than using more models of the same HMSA architecture trained on different datasets. More generally, neither of the two approaches tested on both Track 1 and 2 (our approach with Ensembles and PixOOD) show any notable improvement from using more datasets.

Acknowledgements

We extend our heartfelt gratitude to the authors of ACDC [21], SegmentMeIfY-ouCan [3], and Out-of-context Cityscapes [11] for generously permitting us to repurpose their benchmarking data. We are also thankful to the authors of GuidedDisent [19], Flare Removal [29], and GenVal [14] for providing the excellent toolboxes that helped synthesize realistic-looking raindrops, light flares, and inpainted objects. All have collectively contributed to creating BRAVO, a unified benchmark for robustness in autonomous driving.

The BRAVO Challenge is an initiative within ELSA — European Lighthouse on Secure and Safe AI, a network of excellence funded by the European Union. This work was supported by ELSA and was funded by the European Union under grant agreement No. 101070617.

We provide in this document additional details of the BRAVO challenge and of the solutions.

A BRAVO Challenge

A.1 General Rules

For the BRAVO Challenge 2024 challenge, the following rules applied:

- a) The task is semantic segmentation, with pixel-wise evaluation performed on the 19 semantic classes of Cityscapes.
- b) Models in each track must be trained using only the datasets allowed for that track.
- c) Employing generative models for synthetic data augmentation is strictly forbidden.
- d) All results must be reproducible. Participants must submit a white paper containing comprehensive technical details alongside their results.
- e) Participants must make models and inference code accessible.
- f) Evaluation considers the 19 classes of Cityscapes: ‘road’, ‘sidewalk’, ‘building’, ‘wall’, ‘fence’, ‘pole’, ‘traffic light’, ‘traffic sign’, ‘vegetation’, ‘terrain’, ‘sky’, ‘person’, ‘rider’, ‘car’, ‘truck’, ‘bus’, ‘train’, ‘motorcycle’ and ‘bicycle’.
- g) Teams must register a single account for submitting to the evaluation server. An organization (e.g. a University) may have several teams with independent accounts only if the teams are not cooperating.

A.2 Submissions

For each input image, two files were required: one for the semantic predictions and one for the confidence values.

The class prediction file must be in PNG format, 8-bit grayscale, with each pixel assigned a value from 0 to 19, representing the 19 classes of Cityscapes. The confidence file must also be in PNG format, but 16-bit grayscale, with each pixel’s value ranging from 0 to 65,535, representing the confidence level of the predicted class. Confidence values are evaluated across the entire subset of the dataset simultaneously and, therefore, should be comparable across all images in the subset. Each prediction and confidence file must have the exact same dimensions as the corresponding input image. Evaluation is performed on a pixel-by-pixel basis.

B Submissions digest

From the correlogram in Fig. 2, we observed varying degrees of correlation among the metrics aggregated by the BRAVO subsets.

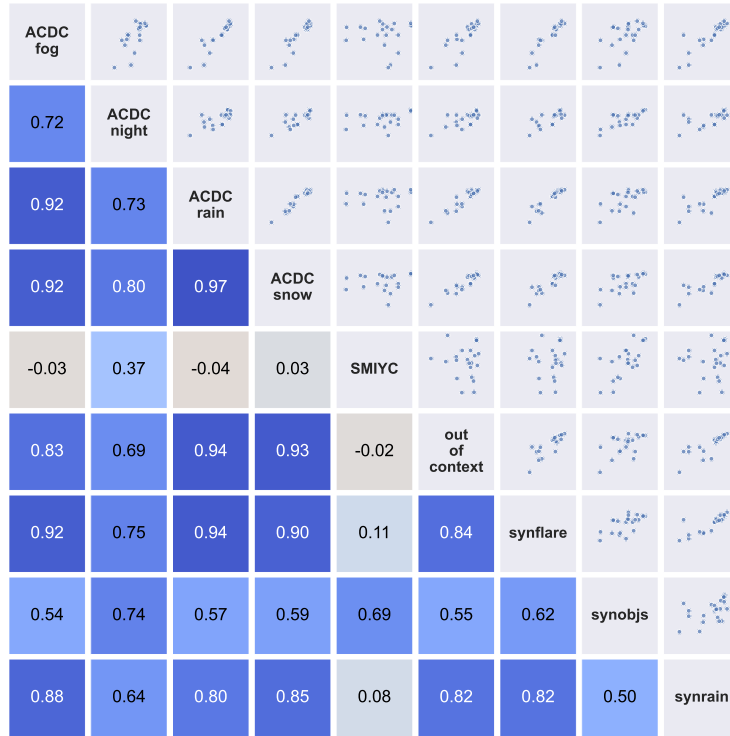


Fig. 2: Analysis showing the correlation of the summary metric of each BRAVO subset.

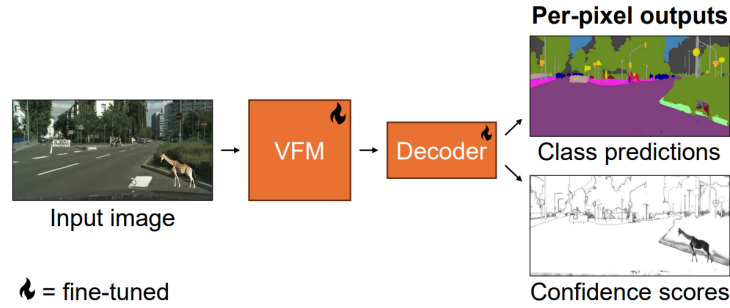


Fig. 3: DINOv2-ODD Meta-approach. We take a pre-trained Vision Foundation Model (VFM), attach a simple segmentation decoder, and fine-tune the entire model for semantic segmentation. The segmentation decoder outputs both the per-pixel classification predictions and the associated confidence scores.

B.1 Track 1: DINOv2-ODD – Eindhoven University of Technology

Authors: Tommie Kerssies, Daan de Geus, Gijs Dubbelman

Method. Figure 3 overviews the DINOv2-ODD approach.

Implementation details. We use the following models from the `timm` library [28] to initialize the VFM:

```

- deit3_small_patch16_224.fb_in1k;
- vit_small_patch14_dinov2;
- vit_base_patch14_dinov2;
- vit_large_patch14_dinov2;
- vit_giant_patch14_dinov2.

```

The models are fine-tuned for 40 epochs using two A6000 GPUs, with a batch size of 1 per GPU and gradient accumulation over 8 steps, resulting in an effective batch size of 16. Our implementation follows the details provided in [13]. Notably, the learning rate for the VFM weights is set to be $10\times$ smaller than the overall learning rate, as this configuration empirically yields better results. For the Mask2Former decoder, we employ a variant specifically adapted for use with a single-scale ViT encoder, as introduced in [13].

Method	ACDC \uparrow	SMIYC \uparrow	Out-of-context \uparrow	Synflare \uparrow	Synobjs \uparrow	Synrain \uparrow
DINOv2, ViT-L, 8x8 patch size, linear decoder	67.3	89.9	71.0	72.7	76.7	73.9
DINOv2, ViT-L, 16x16 patch size, linear decoder	69.4	89.3	70.4	72.4	75.1	73.8
DINOv2, ViT-g, 16x16 patch size, linear decoder	67.7	88.2	71.0	73.2	74.7	73.2
DINOv2, ViT-B, 16x16 patch size, linear decoder	68.5	87.9	71.2	72.8	74.0	73.0
DINOv2, ViT-S, 16x16 patch size, linear decoder	66.9	83.1	70.2	70.6	68.6	72.9
DINOv2, ViT-g, 16x16 patch size, Mask2Former decoder	49.1	94.4	40.9	53.9	64.3	60.1
DeiT III (IN1K), ViT-S, 16x16 patch size, linear decoder	58.8	50.1	65.1	71.4	58.5	65.6

Table 10: Track 1 – DINOv2-OOD – Harmonic means of semantic and OOD metrics for each subset in the BRAVO benchmark dataset, computed for ablated models.

B.2 Track 1: PixOOD – Czech Technical University in Prague

Authors: Tomáš Vojtř, Jan Šochman and Jiří Matas

Method. Figure 4 visualizes the three PixOOD variants that were submitted to the BRAVO challenge 2024.

B.3 Track 1: PhyFea – University Of Bologna & ETH Zurich

Authors: Shamik Basu, Christos Sakaridis and Luc Van Gool

Method. Figure 5 overviews the PhyFea approach.

B.4 Track 2: InternImage-OOD – CASIA & Objecteye

Authors: Long Qian, Bingke Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang

Method. Figure 6 overviews the InternImage-OOD solution.

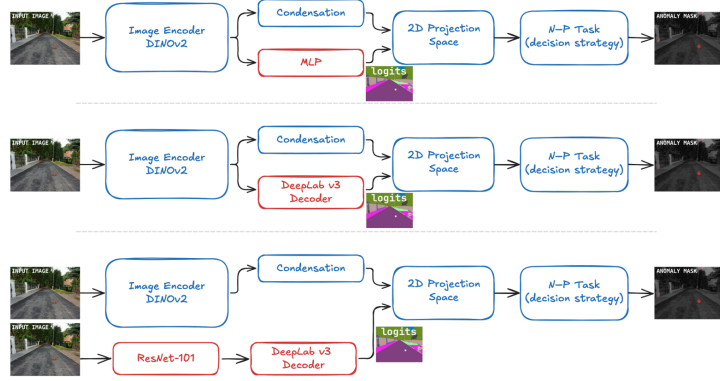


Fig. 4: PixOOD Variants. Simplified block representation of the PixOOD framework for different submitted variants. From top to bottom: PixOOD, PixOOD w/ DeepLab Decoder and PixOOD w/ ResNet101 DeepLab. The blue color denotes blocks that are the same for all variants and are described in the PixOOD. The red color denotes the differences between the methods in the semantic segmentation branches.

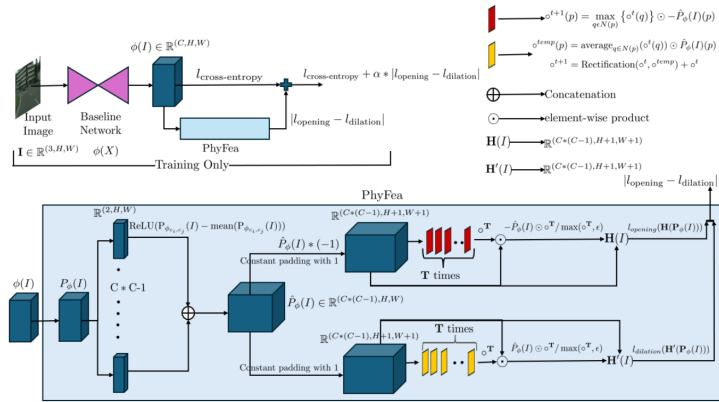


Fig. 5: PhyFea approach. Top left: illustration of the complete network architecture, where the cross-entropy loss of the baseline network is added to the losses of PhyFea. Bottom: the pipeline of PhyFea, where red-colored boxes are iterations for opening and yellow colored boxes are for selective dilation. Top right: legends for various components of PhyFea, such as the operations we apply in iterative manner for area opening and for selective dilation and the two functions to calculate the losses.

B.5 Track 2: Ensemble – McGill University

Authors: Michael Smith and Frank Ferrie

Below are a number of settings which we set when training all models on each dataset but do not explicitly enumerate in the interest of brevity:

- **Splits:** In all cases, we used dataset author-provided training splits.

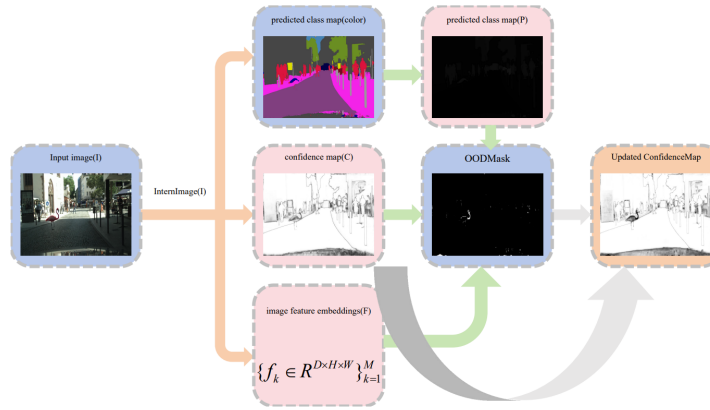


Fig. 6: InternImage-OOD. The diagram illustrates the process of OOD detection and confidence map refinement. Starting with an input image I , the InternImage model generates both the predicted class map P and the confidence map C . Image feature embeddings F are extracted, and K-Means clustering is applied to detect OOD regions, forming the $OOMask$. Finally, the $OOMask$ is used to update the confidence map, resulting in the updated confidence map for further refinement.

- **Tile size:** The training process uses tiling during training to try and ensure a better class distribution when training for some classes that are more rare. We try and set this such that each image can be decomposed into two tiles as best as possible, depending on the size(s) of images contained in the dataset.
- **Crop size/resizing:** We set the image resizing and cropping to match the image size(s) in the dataset as best as possible so as to minimize data loss.
- **Epochs:** All models are trained for 175 epochs unless otherwise indicated. However, the final model used was the one which achieved the best results on the respective validation set of each dataset during the entire training run and thus may have been trained for fewer epochs.
- **Batch sizes:** Training and validation batch sizes are set as high as possible given the VRAM capacity of the GPUs being used for training.

References

1. Basu, S., Sakaridis, C., Van Gool, L.: Physically feasible semantic segmentation. arXiv preprint arXiv:2408.14672 (2024)
2. Besnier, V., Bursuc, A., Picard, D., Briot, A.: Triggering failures: Out-of-distribution detection by learning from local adversarial attacks in semantic segmentation. In: ICCV (2021)
3. Chan, R., Lis, K., Uhlemeyer, S., Blum, H., Honari, S., Siegart, R., Fua, P., Salzmann, M., Rottmann, M.: Segmentmeifyoucan: A benchmark for anomaly segmentation. In: NeurIPS (2021)
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)

5. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: CVPR (2022)
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
8. Dosovitskiy, A.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
9. Englert, B.B., Piva, F.J., Kerssies, T., De Geus, D., Dubbelman, G.: Exploring the benefits of vision foundation models for unsupervised domain adaptation. In: CVPRW (2024)
10. Fahes, M., Vu, T.H., Bursuc, A., Pérez, P., de Charette, R.: A simple recipe for language-guided domain generalized segmentation. In: CVPR (2024)
11. Franchi, G., Belkhir, N., Ha, M.L., Hu, Y., Bursuc, A., Blanz, V., Yao, A.: Robust semantic segmentation with superpixel-mix. In: BMVC (2021)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
13. Kerssies, T., De Geus, D., Dubbelman, G.: How to benchmark vision foundation models for semantic segmentation? In: CVPRW (2024)
14. Loiseau, T., Vu, T.H., Chen, M., Pérez, P., Cord, M.: Reliability in semantic segmentation: Can we use synthetic data? In: ECCV (2024)
15. Malinin, A.: Uncertainty estimation in deep learning with application to spoken language assessment. Ph.D. thesis (2019)
16. Nayal, N., Yavuz, M., Henriques, J.F., Güney, F.: Rba: Segmenting unknown regions rejected by all. In: ICCV (2023)
17. Neuhold, G., Ollmann, T., Rota Bulò, S., Kotschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: ICCV (2017)
18. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: DINOv2: Learning robust visual features without supervision. TMLR (2023)
19. Pizzati, F., Cerri, P., de Charette, R.: Physics-informed guided disentanglement in generative networks. T-PAMI (2023)
20. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: ECCV (2016)
21. Sakaridis, C., Dai, D., Van Gool, L.: Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In: ICCV (2021)
22. Sun, T., Segu, M., Postels, J., Wang, Y., Van Gool, L., Schiele, B., Tombari, F., Yu, F.: Shift: a synthetic driving dataset for continuous multi-task domain adaptation. In: CVPR (2022)
23. Tao, A., Sapra, K., Catanzaro, B.: Hierarchical multi-scale attention for semantic segmentation. arXiv preprint arXiv:2005.10821 (2020)
24. Touvron, H., Cord, M., Jégou, H.: Deit iii: Revenge of the vit. In: ECCV (2022)
25. Varma, G., Subramanian, A., Nambodiri, A., Chandraker, M., Jawahar, C.: Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In: WACV (2019)
26. Vojtř, T., Šochman, J., Matas, J.: PixOOD: Pixel-level out-of-distribution detection. In: ECCV (2024)
27. Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., Hu, X., Lu, T., Lu, L., Li, H., et al.: InternImage: Exploring large-scale vision foundation models with deformable convolutions. In: CVPR (2023)

28. Wightman, R.: Pytorch image models. <https://github.com/rwightman/pytorch-image-models> (2019). <https://doi.org/10.5281/zenodo.4414861>
29. Wu, Y., He, Q., Xue, T., Garg, R., Chen, J., Veeraraghavan, A., Barron, J.T.: How to train neural networks for flare removal. In: ICCV (2021)
30. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. In: NeurIPS (2021)
31. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: CVPR (2020)
32. Zende, O., Schörghuber, M., Rainer, B., Murschitz, M., Beleznaï, C.: Unifying panoptic segmentation for autonomous driving. In: CVPR (2022)

Supplementary Material

The BRAVO Semantic Segmentation Challenge

Results in UNCV2024

Tuan-Hung Vu¹, Eduardo Valle¹, Andrei Bursuc¹, Tommie Kerssies², Daan de Geus², Gijs Dubbelman², Long Qian^{3,4}, Bingke Zhu^{3,5}, Yingying Chen^{3,5}, Ming Tang^{3,4}, Jinqiao Wang^{3,4,5}, Tomáš Vojtíš⁶, Jan Šochman⁶, Jiří Matas⁶, Michael Smith⁷, Frank Ferrie⁷, Shamik Basu⁸, Christos Sakaridis¹⁰, and Luc Van Gool^{9,10}

¹ valeo.ai, France (Challenge Organizers)

² Eindhoven University of Technology, Netherlands

³ Chinese Academy of Sciences, China

⁴ University of Chinese Academy of Sciences, China

⁵ Objecteye Inc., China

⁶ Czech Technical University in Prague, Czechia

⁷ McGill University, USA

⁸ University of Bologna, Italy

⁹ Institute for Computer Science, Artificial Intelligence and Technology, Bulgaria

¹⁰ ETH Zurich, Switzerland

We provide in this document additional details of the BRAVO challenge and of the solutions.

1 BRAVO Challenge

1.1 General Rules

For the BRAVO Challenge 2024 challenge, the following rules applied:

- a) The task is semantic segmentation, with pixel-wise evaluation performed on the 19 semantic classes of Cityscapes.
- b) Models in each track must be trained using only the datasets allowed for that track.
- c) Employing generative models for synthetic data augmentation is strictly forbidden.
- d) All results must be reproducible. Participants must submit a white paper containing comprehensive technical details alongside their results.
- e) Participants must make models and inference code accessible.
- f) Evaluation considers the 19 classes of Cityscapes: ‘road’, ‘sidewalk’, ‘building’, ‘wall’, ‘fence’, ‘pole’, ‘traffic light’, ‘traffic sign’, ‘vegetation’, ‘terrain’, ‘sky’, ‘person’, ‘rider’, ‘car’, ‘truck’, ‘bus’, ‘train’, ‘motorcycle’ and ‘bicycle’.
- g) Teams must register a single account for submitting to the evaluation server. An organization (e.g. a University) may have several teams with independent accounts only if the teams are not cooperating.

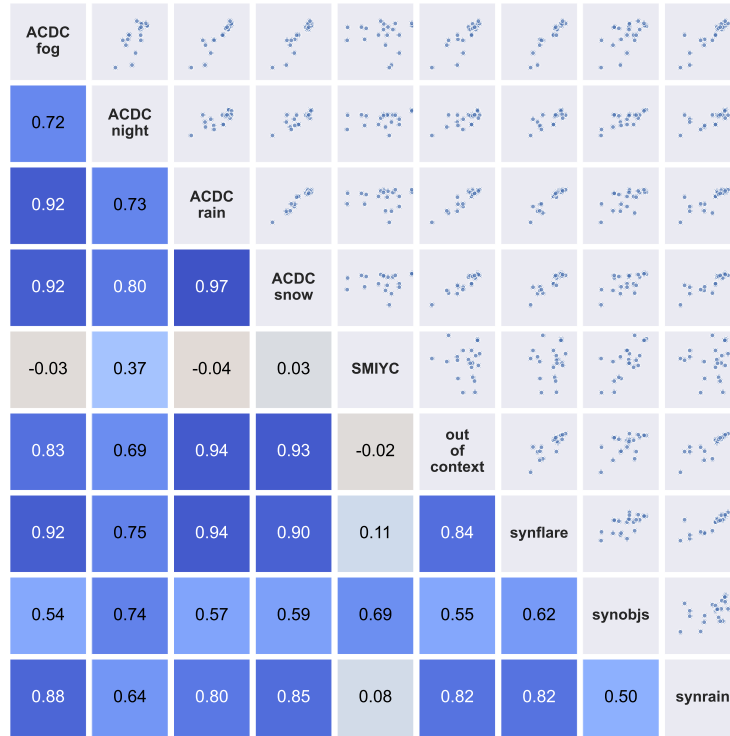


Fig. 2: Analysis showing the correlation of the summary metric of each BRAVO subset.

1.2 Submissions

For each input image, two files were required: one for the semantic predictions and one for the confidence values.

The class prediction file must be in PNG format, 8-bit grayscale, with each pixel assigned a value from 0 to 19, representing the 19 classes of Cityscapes. The confidence file must also be in PNG format, but 16-bit grayscale, with each pixel's value ranging from 0 to 65,535, representing the confidence level of the predicted class. Confidence values are evaluated across the entire subset of the dataset simultaneously and, therefore, should be comparable across all images in the subset. Each prediction and confidence file must have the exact same dimensions as the corresponding input image. Evaluation is performed on a pixel-by-pixel basis.

2 Submissions digest

From the correlogram in Fig. 2, we observed varying degrees of correlation among the metrics aggregated by the BRAVO subsets.

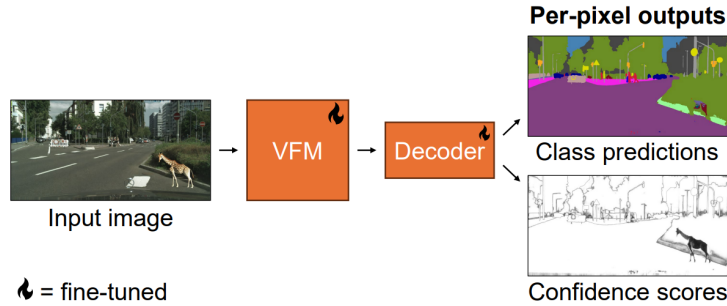


Fig. 3: DINOv2-OOD Meta-approach. We take a pre-trained Vision Foundation Model (VFM), attach a simple segmentation decoder, and fine-tune the entire model for semantic segmentation. The segmentation decoder outputs both the per-pixel classification predictions and the associated confidence scores.

2.1 Track 1: DINOv2-OOD – Eindhoven University of Technology

Authors: Tommie Kerssies, Daan de Geus, Gijs Dubbelman

Method. Figure 3 overviews the DINOv2-OOD approach.

Implementation details. We use the following models from the `timm` library [?] to initialize the VFM:

- `deit3_small_patch16_224.fb_in1k`;
- `vit_small_patch14_dinov2`;
- `vit_base_patch14_dinov2`;
- `vit_large_patch14_dinov2`;
- `vit_giant_patch14_dinov2`.

The models are fine-tuned for 40 epochs using two A6000 GPUs, with a batch size of 1 per GPU and gradient accumulation over 8 steps, resulting in an effective batch size of 16. Our implementation follows the details provided in [?]. Notably, the learning rate for the VFM weights is set to be $10\times$ smaller than the overall learning rate, as this configuration empirically yields better results. For the Mask2Former decoder, we employ a variant specifically adapted for use with a single-scale ViT encoder, as introduced in [?].

Method	ACDC \uparrow	SMIYC \uparrow	Out-of-context \uparrow	Synflare \uparrow	Synobjjs \uparrow	Synrain \uparrow
DINOv2, ViT-L, 8x8 patch size, linear decoder	67.3	89.9	71.0	72.7	76.7	73.9
DINOv2, ViT-L, 16x16 patch size, linear decoder	69.4	89.3	70.4	72.4	75.1	73.8
DINOv2, ViT-g, 16x16 patch size, linear decoder	68.9	90.4	71.1	73.2	74.7	73.2
DINOv2, ViT-B, 16x16 patch size, linear decoder	66.5	87.9	71.2	72.8	74.0	73.0
DINOv2, ViT-S, 16x16 patch size, linear decoder	66.9	88.9	72.0	72.6	74.3	73.5
DINOv2, ViT-g, 16x16 patch size, Mask2Former decoder	49.1	66.7	49.0	53.9	49.9	50.0
DeiT III (IN1K), ViT-S, 16x16 patch size, linear decoder	47.0	62.1	48.4	65.1	47.6	56.4

Table 9: Track 1 – DINOv2-OOD – Harmonic means of semantic and OOD metrics for each subset in the BRAVO benchmark dataset, computed for ablated models.

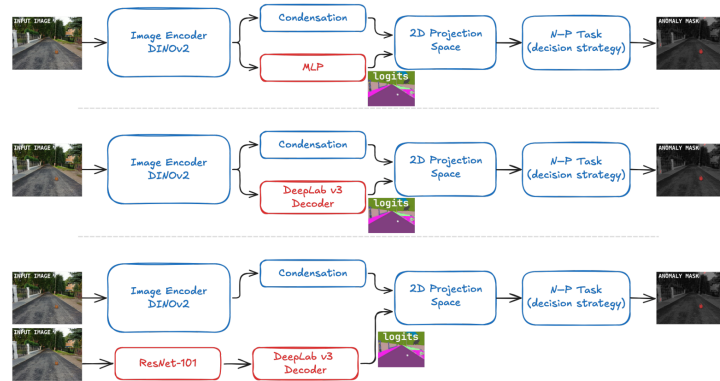


Fig. 4: PixOOD Variants. Simplified block representation of the PixOOD framework for different submitted variants. From top to bottom: PixOOD, PixOOD w/ DeepLab Decoder and PixOOD w/ ResNet101 DeepLab. The blue color denotes blocks that are the same for all variants and are described in the PixOOD. The red color denotes the differences between the methods in the semantic segmentation branches.

2.2 Track 1: PixOOD – Czech Technical University in Prague

Method. Figure 4 visualizes the three PixOOD variants that were submitted to the BRAVO challenge 2024.

2.3 Track 1: PhyFea – University Of Bologna & ETH Zurich

Method. Figure 5 overviews the PhyFea approach.

2.4 Track 2: InternImage-OOD – CASIA & Objecteye

Method. Figure 6 overviews the InternImage-OOD solution.

2.5 Track 2: Ensemble – McGill University

Authors: Michael Smith and Frank Ferrie

Below are a number of settings which we set when training all models on each dataset but do not explicitly enumerate in the interest of brevity:

- **Splits:** In all cases, we used dataset author-provided training splits.
- **Tile size:** The training process uses tiling during training to try and ensure a better class distribution when training for some classes that are more rare. We try and set this such that each image can be decomposed into two tiles as best as possible, depending on the size(s) of images contained in the dataset.
- **Crop size/resizing:** We set the image resizing and cropping to match the image size(s) in the dataset as best as possible so as to minimize data loss.

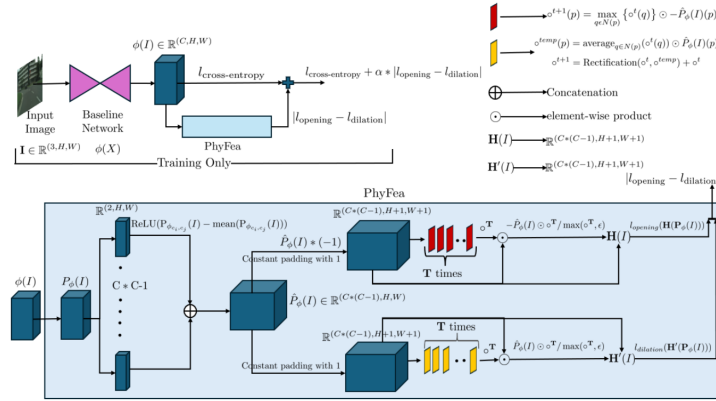


Fig. 5: PhyFea approach. Top left: illustration of the complete network architecture, where the cross-entropy loss of the baseline network is added to the losses of PhyFea. Bottom: the pipeline of PhyFea, where red-colored boxes are iterations for opening and yellow colored boxes are for selective dilation. Top right: legends for various components of PhyFea, such as the operations we apply in iterative manner for area opening and for selective dilation and the two functions to calculate the losses.

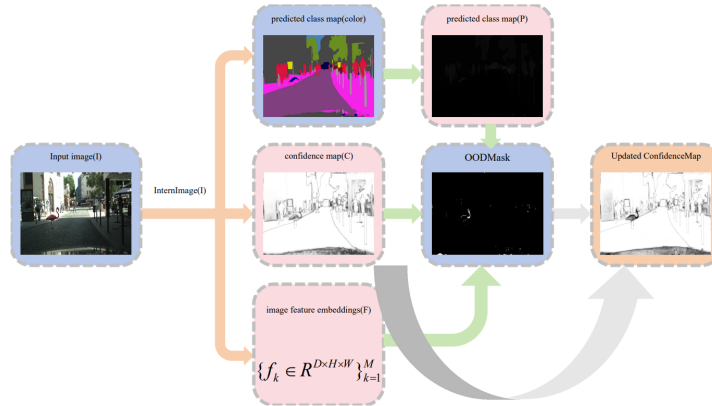


Fig. 6: InternImage-OOD. The diagram illustrates the process of OOD detection and confidence map refinement. Starting with an input image I , the InternImage model generates both the predicted class map P and the confidence map C . Image feature embeddings F are extracted, and K-Means clustering is applied to detect OOD regions, forming the $OODMask$. Finally, the $OODMask$ is used to update the confidence map, resulting in the updated confidence map for further refinement.

- **Epochs:** All models are trained for 175 epochs unless otherwise indicated. However, the final model used was the one which achieved the best results on the respective validation set of each dataset during the entire training run and thus may have been trained for fewer epochs.

- **Batch sizes:** Training and validation batch sizes are set as high as possible given the VRAM capacity of the GPUs being used for training.