An Empirical Analysis of the Embedding Space of the DINOv2 Vision Transformer for Out-of-Distribution Robustness

Bachelor Thesis

Bachelor of Science in Computer Science: Software System Science

Alexander Fröhling

January 16, 2026

Supervisor:

1st: Prof. Dr. Christian Ledig
2nd: Jonas Alle, M.Sc.

Chair of Explainable Machine Learning
Faculty of Information Systems and Applied Computer Sciences
Otto-Friedrich-University Bamberg

Abstract

Vision foundation models developed in recent years have shown convincing results in out-of-distribution (OOD) robustness. Despite this success, one drawback of FMs is the low interpretability of their behaviour.

In this thesis, potential approaches for interpreting the behavior of a vision FM under covariate shift are investigated. The investigated approaches are based on the results of the embedding space of the DINOv2 vision Foundation Model.

As part of this work a methodology was developed that enables the comparison between embeddings of multiple classes across multiple datasets. The presented methodology aims to evaluate covariate shift induced changes on the embedding space in relation to the embeddings of in-distribution (ID) samples. The methodology was applied to the following four datasets: ImageNet-1k, ImageNet-R, ImageNet-V2, and ImageNet-C. A total of 200 comparisons were conducted between the embeddings of one ID class and the embeddings of the corresponding 99 OOD classes. Four different measures were applied to evaluate the difference between the embeddings of two classes.

Based on the obtained results, several patterns could be discerned. For samples from the ImageNet-C dataset, an increase in corruption severity was followed by an increase in the observed shift in the embedding space. When comparing the ImageNet-C results per corruption category the weather and digital categories were found to have less impact on the embedding space than the noise and blur categories. More specifically, the results varied greatly depending on the corruption type of the ImageNet-C dataset. For the samples included in ImageNet-V2, relatively small changes in the embedding space were generally observed. In contrast, the changes for samples from the ImageNet-R dataset were more significant.

Overall, the evaluation of the impact of each OOD dataset on the embedding space differed depending on the measure used.

The code used to produce the reported results is available at: `https://github.com/alexfrhling/dino_v2_ood_robustness`

Abstract

Die in den letzten Jahren entwickelten vision Foundation Models (FMs) zeigen eine überzeugende Robustheit gegenüber Testdaten die außerhalb der Verteilung der Trainingsdaten liegen (OOD Daten). Allerdings besteht ein Nachteil dieser Modelle in der schwierigen Interpretierbarkeit ihres Verhaltens.

In dieser Arbeit werden mögliche Ansätze zur Interpretation des Verhalten von vision FMs bei Vorliegen eines covariate-shifts untersucht. Die Ergebnisse des Embedding Spaces des DINOv2 vision FMs wurden als Grundlage für die erprobten Ansätze verwendet.

Teil dieser Arbeit ist eine entwickelte Methodik, die es ermöglicht die Embeddings mehrerer Klassen über mehrere Datensätzen hinweg zu vergleichen. Die vorgestellte Methodik hat zum Ziel die durch covariate shifts verursachten Veränderungen im Embedding Space relativ zu den Embeddings von In-Distribution (ID) Daten zu bewerten. Die Methodik wurde auf folgenden vier Datensätzen angewendet: ImageNet-1k, ImageNet-R, ImageNet-V2, und ImageNet-C. Insgesamt wurden 200 Vergleiche zwischen den Embeddings jeweils einer ID Klasse und den Embeddings von dazu korrespondieren 99 OOD Klassen durchgeführt. Der Unterschied zwischen den Embeddings zweier Klassen wurden mittels vier unterschiedlicher Bewertungsmaßen quantifiziert.

Auf Grundlage der gewonnen Ergebnisse konnten einige Muster an Veränderungen im Embedding Space beobachtet werden. Für Beispiele des ImageNet-C Datensatzes konnte eine Zunahme der Verschiebung im Embedding Space mit der Zunahme der Korruptionsschwere nachgewiesen werden. Beim Vergleich der Ergebnisse je Korruptionskategorien, die in ImageNet-C enthalten sind, zeigten die weather und digital Kategorien kleinere Auswirkungen auf den Embedding Space zu haben als die Kategorien noise und blur. Im Detail waren die Ergebnisse je Korruptiontyp des ImageNet-C Datensatzes sehr unterschiedlich. Für die Beispiele die in ImageNet-V2 enthalten sind konnten allgemein relativ kleine Veränderungen im Embedding Space beobachtet werden. Die Veränderungen für Beispiele des ImageNet-R Datensatzes waren dagegen deutlicher im Embedding Space zu messen.

Insgesamt konnten Veränderungen in der Bewertung je OOD Datensatz zwischen den vier verwendeten Bewertungsmaßen festgestellt werden.

Der zur Erzeugung der berichteten Ergebnisse verwendete Code ist verfügbar unter: `https://github.com/alexfrhling/dino_v2_ood_robustness`

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
ID	In-Distribution
OOD	Out-of-Distribution
ML	Machine Learning
FM	Foundation Model
NLP	Natural Language Processing
ViT	Vision Transformer
F-OOD detection	Full-spectrum Out-of-Distribution detection
ResNet	Residual Network

1 Introduction

In controlled environments machine learning models achieved performances that even surpassed human capabilities (He et al., 2015). However, under real-world conditions it is often the case that the performance of ML models deteriorates. A study conducted by Beede et al. (2020) demonstrates the practical consequences of this. In their study they analyzed the deployment of an AI system for detecting diabetic retinopathy, a severe disease which can cause blindness. One of the findings of the study was that the AI system often rejected to make predictions on the made images (21% rejection rate) because of low image quality and to keep the number of false predictions at a minimum. One of the reasons made out for the high rejection rate were varying lighting conditions in the room where the pictures were taken (Beede et al., 2020). In contrast to the AI systems, nurses regarded the rejected images as good enough for disease analysis performed by a humans (Beede et al., 2020). The point is, although ML models outperform humans on a specific task they miss the ability to adapt to varying input conditions.

In ML, this problem is known as distribution shift. According to (Quiñonero-Candela et al., 2008), a potential scenario in which this problem is likely to occur is a pattern recognition software developed for a specific camera. Generally, an ML model, including the pattern recognition software, is trained on a set of samples that all stem from the same data distribution, referred to as the In-Distribution (ID) (Yang et al., 2024). The performance of an ML model is expected to stay stable on samples originating from the same data distribution as the training data. However, if a test sample originates from a different distribution than the training samples a, deterioration in performance can be expected. (Zhang et al., 2020) For the pattern recognition software, this would be the case when the software is deployed on a new camera. Images captured by the new camera may differ in certain characteristics from those captured by the camera used to generate the training data. In practice, test samples that deviate from the training data distribution belong to the set of Out-of-Distribution (OOD) data.

The field of ML Robustness is generally considered when discussing counteract measurements on the so far described performance deterioration of ML models for OOD data. Actually ML Robustness is not restricted to the problem of OOD. According to Freiesleben and Grote (2023) ML Robustness is defined as a causally connection between a robustness modifier and a robustness target. Following Freiesleben and Grote (2023) ML Robustness requires that for a change of the modifier the target deviation as a consequence should remain within a certain range of tolerance. Most relevant for the OOD problematic is the connection between deployment distribution (as robustness modifier) and deployment performance (as robustness target). In conclusion, when the deployment distribution changes, the model's performance should remain within a certain range of the performance achieved on ID data.

There is evidence that a more robust model performance can be achieved when using one of the currently emerging Foundation Models (FMs).

According to Bommasani et al. (2022) there are two major points that make a ML model a FM. First, they note that a FM model is initially trained on a large and diverse dataset that is not limited to examples from a specific ML task. Usually, this pretraining is accomplished in a self-supervised manner using unlabeled data (Bommasani et al., 2022). The second important characteristic of an FM according to Bommasani et al. (2022) is, that after pretraining, these models can be adapted to numerous user-specific tasks with comparably less effort.

In practice vision FMs like DINOv2 - which is topic of this thesis - or CLIP showed a significant performance improvement on OOD data compared with a ML model trained in a supervised fashion. The experimental results obtained from Radford et al. (2021) and Oquab et al. (2024) enable a comparison of the performance of CLIP, DINO and ResNet101 on various OOD datasets, all of which are derived of ImageNet-1k. The experimental results are presented in Table 1. The results indicate that there is a significantly smaller accuracy drop for DINOv2 and CLIP on the OOD datasets than for ResNet-101. In conclusion, the results speak for an higher OOD robustness of DINO and CLIP compared with the ResNet-101 model according to the definition of OOD robustness by Freiesleben and Grote (2023).

ML Model	Accuracies			
	ImageNet-1k	ImageNet-R	ImageNet Sketch	ImageNet-A
DINOv2	86.5	78.8	62.5	75.9
CLIP	76.2	88.9	60.2	77.1
ResNet101	76.2	37.7	25.2	2.7

Table 1: Accuracies achieved by DINOv2, CLIP, and ResNet101 on ImageNet-1k as ID dataset and derived OOD datasets, respectively. Data taken from (Oquab et al., 2024) and (Radford et al., 2021).

On the one hand FMs offer great potential to improve on OOD robustness, on the other hand missing understanding on them discourages their use. One reason for the lack of understanding is the complexity of FMs. The complexity of FMs can be quantified for example in the number of learnable parameters. The biggest DINOv2 model has 1.1 billion parameters what is significantly more compared to the previously mentioned ResNet-101 model, which has 44.5 million parameters Oquab et al. (2024); Zagoruyko and Komodakis (2016). Another point is that in most cases it's opaque on which data a FM was trained and how this was done. According to Bommasani et al. (2022) research in FM interpretability is necessary to assess if the use of FMs in real-world applications can be justified.

This work investigates approaches for interpreting the OOD robustness behaviour of DINOv2. The focus of the investigations is on DINOv2‘s embedding space. An embedding of the DINOv2 model is as a higher-dimensional model output which contains general-purpose features that were extracted from an input image. These extracted general-purpose features form the basis for subsequent downstream tasks. Oquab et al. (2024) As all DINOv2 models are Vision Transformers, two different

types of embeddings are returned. One type of returned embeddings is the CLS token, and the other type is known as the patch token. These two output types differ by the fact that the CLS token is explicitly intended for representing global information of the input image. (Dosovitskiy et al., 2021) This work analyses the CLS token behaviour of ViT-S/14 (a specific version of DINOv2) under covariate shift, a particular form of OOD.

Structure of work In chapter 2 an in-depth explanation of the various types of OOD is provided, followed by the technical description of the ViT architecture and details of the DINO models. Chapter 3 begins with the presentation of the comparison approach that was used in this work for evaluating the impact of covariate shifts on the embedding space. The definition of the measures used for embedding comparison concludes this chapter. Chapter 4 discusses all datasets used in this work and their characteristics with respect to OOD. In chapter 5, first the application of the described comparison method is explained step-by-step, followed by the presentation of the experimental results. Finally, in chapter 6, the achieved results are evaluated in relation to the objective of the work.

2 Theoretical Background

2.1 OOD Definition

The field of OOD research lacks a clear taxonomy what makes it difficult to understand the relationships among the variant research fields and to make a systematic investigation. Moreover, a research field's name is not enough to understand which scenarios it actually addresses. For a clearer understanding, first the OOD problematic is formalized in a mathematical way then the relevant research terms are explained based on this formalization subsequently. After this general explanation of OOD in ML follows a focused description of OOD in ML in computer vision.

To understand the theory behind OOD it's helpful to imagine the case of a supervised image classification task. A ML model intended for this task will be trained on a training dataset $D = \{(x, y), ...\}$ (Gulrajani and Lopez-Paz, 2020). Each element of this training dataset includes a picture x and the corresponding ground truth label y. In theory, all training examples are independently drawn from the same joint probability distribution $P(X, Y)$ (Zhang et al., 2020). Following the theoretical idea, after model training there is only a guarantee for a test instance to be correctly classified when it belongs to the identical distribution as the training examples (Zhang et al., 2020). In this work OOD is defined as the case where a test instance is from a different distribution than the training distribution, what is named as a dataset shift in (Moreno-Torres et al., 2012). This shift in the distribution $P(X, Y)$ that comes with an OOD instance is the reason why ML models suffer from a performance decrease (Zhang et al., 2020).

Moreno-Torres et al. (2012) categorize distributional shifts of $P(X, Y)$ in four categories, which are considered a good basis for a more detailed understanding of the problem. Therefore first these four categories by Moreno-Torres et al. (2012) are described followed by the explanation of additional OOD types and important terms.

Moreno-Torres et al. (2012) introduces at first two different decompositions of $P(X, Y)$, which are relevant for understanding the four categories:

- $P(x, y) = P(y|x)P(x)$
- $P(x, y) = P(x|y)P(y)$

The term $P(y|x)P(x)$ considers tasks where the input x determines the value of the class label y. The already mentioned supervised image classification task belongs to this type. In the other case, y defines the value of x what is described by the term $P(x|y)P(y)$. For this case the diagnosis of a disease would be a potential example, where x (the symptoms) is the result of y (the disease). (Moreno-Torres et al., 2012)

1) *Covariate shift*, is relevant for problems of the type $P(y|x)P(x)$. In case of a covariate shift the probability $P(x)$ changes but not the conditional probability $P(y|x)$. An example for this is where an image classification model is deployed to a

new camera, which encodes the made pictures in a different way compared with the encoding of the pictures used for training the model. (Moreno-Torres et al., 2012)

2) *Prior probability shift*, is relevant for problems of the type $P(x|y)P(y)$. Here the distribution of the class labels $P(y)$ changes but again not the conditional distribution, which is $P(x|y)$. (Moreno-Torres et al., 2012) A practical scenario which is affected by this shift is a ML model intended to detect spam e-mails (Quiñonero-Candela et al., 2008). When the ratio of spam e-mails in the training set is lower than in a real mailbox, then there is a shift of $P(y)$.

3) *Concept shift*, is relevant for problems of both types $P(y|x)P(x)$ and $P(x|y)P(y)$. In this type of shift the marginal probabilities ($P(x)$ for former, $P(y)$ for the latter term) remain fixed and the conditional probabilities change. (Moreno-Torres et al., 2012) Lane and Brodley (1998) describe in their paper following learning task that is affected by a concept shift. The goal is to train a model that recognizes anomalies in a computer user's behaviour what would indicate the doing of an malicious actor. In practice the behaviour of the user will change over time what causes a concept drift (Lane and Brodley, 1998). Expressed in a mathematical way the conditional probability $P(x|y)$ changes, because a user's ordinary behaviour will be observable with different patterns over time.

4) *Further shifts*, are relevant for problems of both types (Moreno-Torres et al., 2012). The last shift defined by (Moreno-Torres et al., 2012) considers a shift of the marginal probability as well of the conditional probability. In the view of (Moreno-Torres et al., 2012) these kind of shift has less importance for practical ML.

Despite this comprehensive definition of distributional changes, important OOD terms have not yet been addressed.

Semantic shift, is another distributional shift which isn't covered by any of the introduced shifts so far. In case of a semantic shift, a ML model is faced with an instance which belongs to a class the model wasn't trained on to classify (Yang et al., 2024). For example a semantic shift would be present if a ML model trained to classify pictures of dogs and cats encounters a picture of a penguin during the test phase. Regardless whether the ML model categorizes the picture of the penguin as an instance of a cat or of a dog it would be a false prediction. With the new encountered class the probability distribution of $P(y)$ during test phase is different from $P(y)$ during training phase (Yang et al., 2024). Because the images belonging to the new class are not part of the image collective of the training phase, $P(x)$ also changes in the event of a semantic shift (Yang et al., 2024). Moreover $P(y|x)$ is also different, because $P(y|x)$ of the training distribution doesn't describe the relation between input pictures and class label for the new encountered class. Additionally some research papers distinguish between sub-categories of semantic shifts or classifying the semantic shift of interest for their research as of special kind. Yang et al. (2023) differentiate the group of semantically shifted classes between instances that have a smaller covariate shift to the ID class instances and ones with a greater covariate shift to the ID class instances. They aimed to achieve a more thorough evaluation when assessing methods for detecting semantically shifted instances. In another

paper by Noda et al. (2025) the semantic relationships among Imagenet-1k classes were exploited in order to create one ID dataset and one OOD dataset from the Imagenet-1k dataset. By this approach they created an OOD dataset which classes are semantically very close to the ID classes.

Domain shift, is a another often used term in the context of ML model robustness. Quiñonero-Candela et al. (2008) explains domain shift as a scenario where the way something is described changes but not what is described. More formally: In case of a domain shift "the measurement system, or the method of description" (Quiñonero-Candela et al., 2008) has changed. An example for this definition would be the measurement of the same distance in meters and in inches (Quiñonero-Candela et al., 2008). The measurement in meters would be different from the measurement in inches, but both describe the same distance. Another example from the field of image classification would be the change of the form of class instances from photographs to sketches (Yang et al., 2024). In the context of the already discussed dataset shift types, Yang et al. (2024) sees domain shift as a form of a covariate shift. This categorization is better understandable in a mathematical form of description. In case of a domain shift as well as in the case of a covariate shift the marginal distribution $P(x)$ changes. Also in both cases there is no new class introduced and so $P(y)$ remains constant. Because domain shift is in a special form of a covariate shift, by the definition for a covariate shift $P(y|x)$ remains fixed in case of a domain shift.

Spurious correlation is not a form of dataset shift but is one reason why ML models make false predictions for OOD instances. Spurious correlation describes the general problem when the inference a ML model makes is based on non-causal features Ye et al. (2025). Non-causal features could be for example: background of images, textures, or secondary objects Ye et al. (2025). In a more visual way, such an over-reliance on spurious features is given when a ML model correctly classifies the image of a cow when there is green grass in the background but isn't able to classify images of cows when there is no green grass in the background (Bommasani et al., 2022). Spurious correlation may be a problem for model robustness in case of covariate shift and semantic shift. By definition of a covariate shift $P(x)$ changes and consequently the non-causal features may dissipate. When the non-causal features are no longer present the ML models learned inference my fail to correctly classify the examples from the shifted dataset Ye et al. (2025). As discussed before, in case of a semantic shift an example of a not trained class was encountered and the distribution $P(y)$ shifted by this. In such a scenario expectations on a robust ML model would be to not predict this test instance as an instance of one of the trained classes with high confidence. However spurious correlation induces this undesired behaviour, as spurious features in examples with a semantic shift cause the ML model to classify these examples as one of the trained classes Ming et al. (2022).

The so far described definitions and relevant terms were aimed to give a general overview for OOD in ML. According to published papers considering OOD for ML in computer vision the most often mentioned kinds of OOD in practical research are covariate shifts and semantic shifts (Noda et al., 2025; Yang et al., 2024). The

definitions for these two OOD types are given above but unclear is yet what a robust response of a ML vision model to these OOD types respectively would be. To elaborate on this question it's worth considering the idea behind full-spectrum out-of-distribution detection (F-OOD detection), first mentioned by Yang et al. (2023). The overarching idea of F-OOD detection is that covariate shifted instances should be ideally dealt as ID instances by a model and then at best correctly classified (Yang et al., 2023). Conversely, instances with a semantic shift should be identified as OOD and rejected from classification by the model in order to avoid incorrect predictions (Yang et al., 2023).

2.2 DINO

The name DINO originates from the work of Caron et al. (2021), which aimed to study self-supervised pretraining for the Vision Transformer (ViT) architecture. In this work, they introduced a new self-supervised training method, which they described as a kind of knowledge distillation without labels and is abbreviated as DINO. Together with their paper in which the DINO training method was explained, Caron et al. (2021) also published a series of models trained with the DINO method, which are also often named DINO. These models can be considered the predecessors of DINOv2. After the start of this thesis, a newer model version, DINOv3, was published (Siméoni et al., 2025).

First, the ViT architecture is explained. Subsequently the characteristics of each DINO model version are summarized.

2.2.1 Vision Transformer Architecture

The Vision Transformer (ViT) architecture, introduced by Dosovitskiy et al. (2021), is an adaptation of the Transformer architecture. Their purpose was to apply the Transformer architecture, which has proven successfully in the field of Natural Language Proccessing (NLP), to vision tasks. The original Transformer architecture, introduced by Vaswani et al. (2023), consists of an encoder part and a decoder. The ViT architecture employs only the encoder part, also known as the Transformer encoder (Dosovitskiy et al., 2021). To understand the role DINOv2's embedding space in the context of the general ViT architecture, it is helpful to begin with the preprocessing of an input image before the encoder is applied on the data. The Transformer encoder expects sequential data as input, which initially contradicts the two-dimensional form of images (Dosovitskiy et al., 2021). As solution, according to Dosovitskiy et al. (2021) at first the image is cut into square-size patches (step depicted at bottom on the left side in 1). Afterwards, each of these two-dimensional patches is first flattened into a one-dimensional array, then a linear transformation is applied to each flattened patch, resulting in what is called a patch embedding. An additional embedding, the cls embedding, is then appended to the series of patch embeddings (marked with the arrow in the middle on the left side in 1). Compared to the patch embeddings the cls embedding is of special kind, as it does not originate

from any part of the input image. Before the sequence of embeddings is passed to the Transformer encoder, a positional embedding is added to each embedding to convey spatial information that was lost when the image was cut into patches.

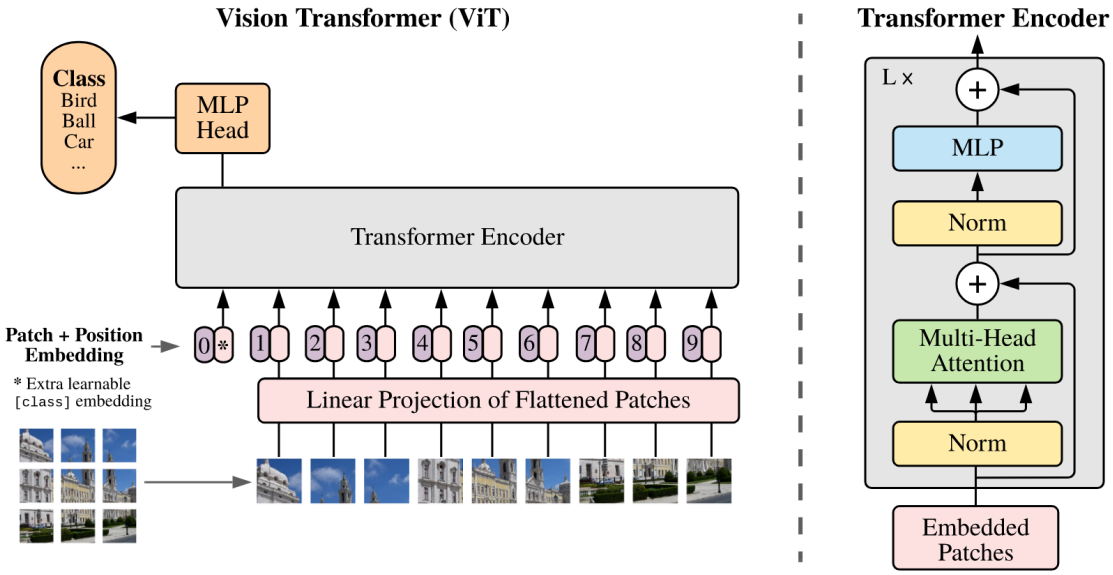

Figure 1: Complete ViT architecture on the left. Processing steps performed at each layer within the Transformer encoder on the right. Image from Dosovitskiy et al. (2021).

When the input sequence is next processed by the Transformer encoder, there on multiple layers the same sequence of operations is applied (details shown in 1 on the right). One important operation within a layer is the self-attention mechanism, which gathers information across all embeddings. Dosovitskiy et al. (2021) The Transformer encoder returns a corresponding patch token for each patch embedding given as input, as well as a CLS token for the cls embedding (the patch tokens are not explictly visible in 1)(vom Lehn, 2025). The information encoded in the CLS token and patch tokens is different. The CLS token is regarded as a representation of the complete image Dosovitskiy et al. (2021). Conversely, a patch token encodes the information about the original image patch and its relation to the other image patches (vom Lehn, 2025). As a consequence, the CLS token is used for tasks requiring information about the complete image, such as classification (Devlin et al., 2019). The patch tokens are used for dense recognition tasks, such as semantic segmentation and monocular depth estimation (Oquab et al., 2024).

2.2.2 DINO(v1)

Originally the motivation for Caron et al. (2021) was to study the effects of self-supervised pretraining on vision transfomers, what ultimately led to the DINO training method. The researches were inspired by the success of combining Transformer

architecture and self-supervised pretraining for NLP tasks. In Caron et al. (2021) five neural networks were trained using the DINO method: four ViTs and one Residual Network (ResNet). The latter was used to demonstrate the applicability of the DINO training method to convolutional neural nets. The biggest ViT model had 85 million parameters, while the ResNet model had 23 million parameters. All networks were trained on the ImageNet-1k dataset, which training split provides ca. 1.3M images. Evaluation results showed that a ViT and a ResNet backbones, both of which were trained with the DINO method and are almost equally in size, achieved similar performance on the ImageNet-1k validation set. The ViT backbones showed the unique property that a kNN classifier used together with the models' encoded features performs almost as well as a linear classifier, which isn't the case for the ResNet backbone. Another observation from their study was that features encoded by ViT backbones distinguish between class-relevant and class-irrelevant image areas. (Caron et al., 2021)

2.2.3 DINOv2

In the further development of DINO in the form of DINOv2 and DINOv3 outstanding is the increase in model and training dataset size. In contrast to DINO, with DINOv2 the explicit attempt was made to build a FM for vision tasks. Such an FM is expected to provide features for some given input image that can be applied to many vision tasks that weren't explicitly defined during model pretraining. These features, generated by a vision FM, are also referred to as general-purpose features due to their agnostic nature. The biggest model provided under DINOv2 is the ViT-g/14. (Oquab et al., 2024) The exact model notation applied here refers to the notation kind introduced by Dosovitskiy et al. (2021). The g in the model notation depicts the model size (according to hug (2025a) g stands for giant). The number in the latter part stands for the patch size that the model expects. ViT-g/14 has 1.1 billion parameters and was pretrained on the LVD-142M dataset, which contains 142 million images. Pretraining on LVD-142M is considered an essential factor to DINOv2's ability of generating general-purpose features of good quality. Besides ViT-g/14 a series of smaller DINOv2 models are provided. None of the DINOv2 models different than ViT-g/14 were trained on the LVD-142M dataset. Instead, these smaller models were derived through a process called knowledge distillation from the ViT-g/14 model (Hinton et al., 2015). When comparing the training procedures of DINOv2 and DINO then it becomes clear that these are not exactly the same. For example, in the first model versions of DINO the calculation of the training loss is solely based on the output of the CLS tokens, whereas in DINOv2 an additional loss-term is calculated on the output of the patch tokens (Caron et al., 2021). (Oquab et al., 2024)

2.2.4 DINOv3

In principle, DINOv3 is based on the work done in DINOv2. The main difference between DINOv3 and DINOv2 is scale. DINOv3's biggest model, ViT-7B/16, has 6.7 billion parameters, whereas DINOv2's biggest model, ViT-g/14, has 1.1 billion parameters (Siméoni et al., 2025; Oquab et al., 2024). ViT-7B/16 was pretrained on the LVD-1689M dataset, which contains 1.689 billion curated images drawn from a set of 17 billion images taken from Instagram. Additional pretraining was conducted with task-specific subsets from the 17 billion images, as well as with other public benchmark datasets, e.g. Imagenet-1k. Similarly to DINOv2, a series of smaller models was distilled from the largest DINOv3 model, ViT-7B/16. Some changes were made to the pretraining process to successfully increase the scale from DINOv2 to DINOv3. According to (Siméoni et al., 2025), one encountered problem was the degradation of the patch tokens' feature quality the longer the pretraining. In more detail, they showed that the greater the number of training iterations, the more frequent is the case where the output for two patch tokens is very similar, but does not align with the semantics of the input patches. In DINOv3 to solve this problem, an additional loss term based on Gram matrices with patch tokens as input was applied. As a consequence of this new loss term, there was a notable improvement in the patch tokens' feature quality compared to pretraining without this new loss term. (Siméoni et al., 2025)

2.3 Logistic Regression

In the model card of DINOv2 several options are described how the output of the CLS token and the patch tokens can be utilized for custom tasks. One of these options is to use a logistic regression model for image classification tasks (din, b). In this specific case, the logistic regression model takes a CLS token as input and returns a series of probability scores, one for each image class.

The principle task is to predict the probability of each class $k \in \{1, 2, ..., K\}$ based on the value of the CLS token, represented by x. The logistic regression model calculates a probability value for each class k in two steps. In the first step, an affine function is applied to the CLS token, followed by the application of the softmax function in the second step (div).

Affine Function The formula for an affine function, following Goodfellow et al. (2016), was adapted as follows:

$$o_k = {w_k}^T \cdot x + b_k \tag{1}$$

where:

- w_k: is the weight vector associated with class k

- x: represents the CLS token
- b_k: bias associated with class k
- o_k: affine transformation result for class k

In this formula the dot product is applied on the CLS token, represented by the x, and the weight vector w_k that is specific to class k. The bias b_k that is also specific to class k is added to the result of the dot product what gives o_k as result.

Softmax Function As the final result for each class k a probability value in the range $[0; 1]$ is required what is not the case for the result of the affine function. Furthermore, the final results for all classes should sum up to 1. Both requirements are fulfilled when the softmax function is applied on the output of the affine functions. (div) According to sta the definition of the softmax function is as follows:

$$P(y = k) = \frac{e^{o_k}}{\sum_{j=1}^{K} e^{o_j}} \tag{2}$$

3 Methodology

3.1 Comparison concept

Principle Idea When evaluating the robustness of an embedding space on covariate-shifted data, the general question is how similar an embedding of a covariate-shifted instance is to the embedding of a non-covariate-shifted instance when the corresponding input pictures of both represent the same topic. To justify this idea in practical terms, a logistic regression model trained on the ID dataset's embeddings is considered. This model does not extract additional features from the embeddings. Consequently, the more closer two embeddings are, the greater the chances that these two will be classified in the same category. Furthermore, an ML model is more robust to covariate shift the higher the proportion of correctly classified covariate-shifted data. Together, these two assumptions lead to the conclusion that, for high robustness of the embedding space to covariate shift, the embeddings of covariate-shifted images must be as similar as possible to the embeddings of ID images in the same class.

In order to measure the similarity of embeddings at a dataset scale, a general comparison concept was first developed. Initially, there is one ID dataset and one or more OOD datasets. The ID dataset contains a specific set of classes, represented by S. In this comparison setup, each of the used OOD dataset contains a subset of the classes from S. The samples in each of the OOD datasets have a covariate shift to the samples of the ID dataset.

Class-level comparisons In this work only class-level comparisons were of interest. When comparing two classes, the comparison were based on the statistics of the overall set of embeddings corresponding to each class. Considered were the mean value per dimension and standard deviation per dimension. In this work, the embedding that contains all the mean dimension values of classX is referred to as the mean embedding of classX. When $\{x_1, x_2, ..., x_N\}$ is the complete set of embeddings corresponding to some arbitrary classX, the mean embedding for that classX was defined this way:

$$\overline{x}[k] = \frac{1}{N} \sum_{i=1}^{N} x_i[k] \tag{3}$$

, and the standard deviation per dimension for that classX was defined this way:

$$\sigma(x)[k] = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i[k] - \overline{x}[k])^2} \tag{4}$$

In this work, $x_i[k]$ denotes the value at the k-th dimension of the embedding with index i from the set of embeddings $\{x_1, x_2, ..., x_N\}$.

Comparison scheme The basic structure of the comparisons made is that a classX from an OOD dataset was compared with the same classX from an ID dataset. However, a comparison between two classes results in a single value what is not enough for evaluating the robustness of the Embedding Space to covariate shift. Therefore, this value was set into relation with an additional value from the comparison of classX from the ID dataset with another classY, also from the ID dataset. ClassY represents a different topic to classX. In summary, two different comparisons were conducted: classX (OOD dataset) with classX (ID dataset), and classY (ID dataset) with classX (ID dataset).

The so far described principal idea of classwise comparisons can be formalized with following universal scheme:

$$(classA, classA', classB, classR, classMean) \tag{5}$$

- $classA$: class from the ID dataset that describes the same topic as $classA'$
- $classA'$: this class subsumes all classes from the OOD datasets which represent the same topic as $classA$
- $classB$: class from the ID dataset that is very similar to $classA$
- $classR$: a randomly chosen class from the ID dataset that is neither $classA$ nor $classB$
- $classMean$: class which subsumes all ID classes, so also $classA$, $classB$, and $classR$, and will be used to derive a mean value of the complete ID dataset for the following comparisons

With this scheme comparisons are made between classes that are from one or more OOD datasets and several classes from the ID dataset. Because the $classA$ and $classA'$ must represent the same topic, the number of times this scheme can be applied is limited by the number of classes that are part of the ID dataset as well of all OOD datasets.

The special role of $classB$ is discernible when considering the classification behaviour of a linear classifier. $ClassB$ was defined as the class which embeddings are most similar to the embeddings of $classA$ within a comparison sequence. Details about the way similarity was practically measured and manifestation of $classB$ is part of the Experiments & Results chapter (see 5). As $classB$ embeddings are most similar to $classA$ embeddings, if an embedding of $classA$ is classified wrong, chances are highest for this embedding to be classified to $classB$. The measurement results of $classB$ embeddings, together with $classA$ embeddings, define a certain value range. For the evaluation of the embedding space that means, that the greater the similarity of $classA'$ embeddings to classA embeddings compared to the $classB$ embeddings to $classA$ embeddings the better the robustness of the Embedding Space.

The two classes classR, and classMean were not discussed in more detail yet. The motivation for introducing these two classes was to generate contrast values for the measurement values of classB. For the approach described above to be meaningful, classB should be significantly more similar to classA than any other ID class. Therefore, classR and classMean were added to the comparison scheme, which are representative of all not considered ID classes. A specialty of classMean is that the mean embedding value of that class remains constant for all comparisons that were made. The measurement values of classMean from the comparison with classA are therefore considered a suitable baseline for evaluating the measurement values obtained from the comparisons of the other classes with classA.

3.2 Similarity Measures

Four different measures were applied to measure the similarity between two classes in the comparison scheme. These similarity measures are abbreviated as L1, C1, C2 and C3. The L1 Measure (also known as Manhattan Distance) is commonly used in the field of ML (Aggarwal et al., 2001; Goodfellow et al., 2016). The remaining three measures, which all have a C in their name, were self-defined. These four measures were applied to class-level statistics as defined in the equations 3 and 4.

3.2.1 L1 Measure

$$L1(\overline{\mathbf{c}}, \overline{\mathbf{e}}) = \sum_{k=1}^{N} |\overline{\mathbf{c}}[k] - \overline{\mathbf{e}}[k]| \tag{6}$$

In the formula, for each embedding dimension the absolute distance between $\overline{\mathbf{c}}$ and $\overline{\mathbf{e}}$ is first computed. The resulting dimension-wise distance values are summed up to a single value, which is in the range from zero to infinity. The closer two embeddings are to each other in the embedding space, the smaller is their corresponding L1 measure value.

When C is the corresponding class to $\overline{\mathbf{c}}$ and E the corresponding class to $\overline{\mathbf{e}}$, the inputs to $L1(\overline{\mathbf{c}}, \overline{\mathbf{e}})$ can be defined by the definition of C and E.

- $C \in \{classA', classB, classR, classMean\}$
- $E \in \{classA\}$

Consequently, the L1 distance was computed for four different combinations of general classes:

$$(classA, classA'), (classA, classB), (classA, classR), (classA, classMean) \tag{7}$$

3.2.2 C1 Measure

$$C1(\overline{\mathbf{c}}, \overline{\mathbf{e}}, \overline{\mathbf{g}}) = \sum_{k=1}^{N} \mathbb{1}\{|\overline{\mathbf{c}}[k] - \overline{\mathbf{e}}[k]| < |\overline{\mathbf{g}}[k] - \overline{\mathbf{e}}[k]|\} \tag{8}$$

The C1 measure is expected to work with three different mean embeddings given as arguments: $\overline{\mathbf{c}}$, $\overline{\mathbf{e}}$, and $\overline{\mathbf{g}}$. Each embedding dimension is indexed with k, where N represents the maximum embedding dimension. Over all embedding dimensions the same function is applied, symbolized with the $\mathbb{1}$ and the corresponding braces in which a conditional statement is defined. The $\mathbb{1}$-function returns 1 as an integer if the conditional statement evaluates to true and 0 if not. The conditional statement checks whether the absolute difference between the values of $\overline{\mathbf{c}}$ and $\overline{\mathbf{e}}$ is smaller than the absolute difference between $\overline{\mathbf{g}}$ and $\overline{\mathbf{e}}$ for the same dimension. The results of the $\mathbb{1}$-function for all embedding dimensions are finally summed up, yielding a value between 0 and the number of embedding dimensions.

Again the inputs for $\overline{\mathbf{c}}$, $\overline{\mathbf{e}}$, and $\overline{\mathbf{g}}$ can be defined by the definition of their associated classes, C, E, and G:

- $C \in \{classA', classB, classR\}$
- $E \in \{classA\}$
- $G \in \{classMean\}$

Consequently, the input to the C1 measure can be described by the following combinations of general classes:

$$(classA', classA, classMean), (classB, classA, classMean), (classR, classA, classMean) \tag{9}$$

Each combination follows the structure $(\overline{\mathbf{c}}, \overline{\mathbf{e}}, \overline{\mathbf{g}})$, as defined in 8.

3.2.3 C2 and C3 Measure

$$\begin{aligned} C2(\overline{\mathbf{e}}, \sigma(\mathbf{e}), \overline{\mathbf{c}}, \overline{\mathbf{g}}) = \sum_{k=1}^{N} \mathbb{1}\{&|\overline{\mathbf{c}}[k] - \overline{\mathbf{e}}[k]| \leq \sigma(\mathbf{e})[k] \wedge |\overline{\mathbf{g}}[k] - \overline{\mathbf{e}}[k]| > \sigma(\mathbf{e}[k]) \\ &\vee |\overline{\mathbf{c}}[k] - \overline{\mathbf{e}}[k]| \leq \sigma(\mathbf{e}[k]) \wedge |\overline{\mathbf{g}}[k] - \overline{\mathbf{e}}[k]| \leq \sigma(\mathbf{e})[k]\} \end{aligned} \tag{10}$$

$$C3(\overline{\mathbf{e}}, \sigma(\mathbf{e}), \overline{\mathbf{c}}, \overline{\mathbf{g}}) = \sum_{k=1}^{N} \mathbb{1}\{|\overline{\mathbf{c}}[k] - \overline{\mathbf{e}}[k]| \leq \sigma(\mathbf{e}[k]) \wedge |\overline{\mathbf{g}}[k] - \overline{\mathbf{e}}[k]| > \sigma(\mathbf{e})[k]\} \tag{11}$$

The C2 and C3 measures both take as input the mean embedding and the standard deviation values per dimension of one specific class, plus the mean embedding of two

other classes. Both measures check in principle if the mean embedding values of the one class fall within the standard deviation given as input in relation to the same comparison made with the second mean embedding. Similarly to the C1 measure, for the C2 and C3 measures the $\mathbb{1}$-function checks, for each dimension, whether a condition specific to each measure is fulfilled. In the C2 measure a disjunction of two conditions is checked. In the first condition of the disjunction is checked if the dimension value $\overline{\mathbf{c}}[k]$ falls within the range around $\overline{\mathbf{e}}[k]$ defined by the standard deviation $\sigma(\mathbf{e})[k]$, while the dimension value of the second mean embedding $\overline{\mathbf{g}}[k]$ does not. In the second condition of the disjunction is checked if both mean embedding values, $\overline{\mathbf{c}}[k]$ and $\overline{\mathbf{g}}[k]$, fall within the defined range. The logical disjunction could be simplified to: $|\overline{\mathbf{c}}[k] - \overline{\mathbf{e}}[k]| \leq \sigma(\mathbf{e}[k])$. However, because the calculation of the C2 measurement values were coupled to the calculation of the C3 measurement values in praxis, the definition of the C2 measure was made explicit here. In the C3 measure for all embedding dimensions it's checked if the dimension value of the one mean embedding, $\overline{\mathbf{c}}[k]$ falls within the range defined by the standard deviation, while the dimension value of the other mean embedding, $\overline{\mathbf{g}}[k]$ does not.

Again the inputs for $\overline{\mathbf{c}}$, $\sigma(\mathbf{e})$, $\overline{\mathbf{e}}$, and $\overline{\mathbf{g}}$ can be defined by the definition of their associated classes, C, E, and G:

- $C \in \{classA', classB, classR\}$
- $E \in \{classA\}$
- $G \in \{classMean\}$

Consequently, the C2 and C3 measures were applied on three combination of general classes:

$$(classA, classA, classA', classMean), (classA, classA, classB, classMean), \quad (12)$$
$$(classA, classA, classR, classMean)$$

The different conditions in the C2 and C3 measures are based on the hypothesis that class-specific properties are encoded into certain embedding dimensions. The condition used in the the C3 measure aims exactly to filter these dimensions depending on class-A. The condition in the C3 measure is only fulfilled when the mean dimension value of class-Mean is not within the standard deviation of class-A for a specific dimension. The dimension values of class-Mean represent the mean across all embeddings of all classes. In conclusion, the dimension values of class-Mean are not specific to any class. If class-Mean is close to the distribution of the embedding values of class-A of a specific dimension, then that dimension is not considered one that encodes class-specific features of class-A.

4 Datasets

4.1 ID Dataset

ImageNet-1k As the ID dataset, the validation set of the ImageNet-1k dataset was used. It covers 1,000 classes and contains 50,000 samples in total. The creation of the ImageNet-1k dataset dates back to the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) from 2012. From 2010 to 2017 the ILSVRC was held annually (ima). The task in this challenge was to develop ML models with improved performance on various vision tasks using a certain dataset. At least for the years 2012 to 2014 it is guaranteed that the training and test set (including the validation set) remained constant, which is also known as the ImageNet-1k dataset. The ImageNet-1k dataset is well established in the field of ML and widely used beyond ILSVRC. (Russakovsky et al., 2015)

The training set of the ImageNet-1k dataset is a subset of the larger ImageNet dataset, first introduced in 2009 (Deng et al., 2009), which forms the basis for all ILSVRC datasets (Russakovsky et al., 2015). The complete ImageNet dataset, also known as ImageNet-21K, covers 21,841 classes and contains 14,197,122 images in total (Ridnik et al., 2021). For the creation of ImageNet-1k 1,000 classes from the ImageNet dataset were selected, along with their corresponding examples, as the training set for the Imagenet-1k dataset (Russakovsky et al., 2015). The ImageNet dataset was part of the pre-training data used for DINOv2 Oquab et al. (2024). Consequently, DINOv2 was trained on classes that were also used in the experiments conducted in this work. However, DINOv2 was not trained on the exact same data used in the experiments, since the ImageNet-1k validation set is not a subset of the ImageNet dataset (Russakovsky et al., 2015).

There is a special semantic structure among the 21,841 classes of ImageNet, which becomes clear through a description of ImageNet's creation process. The set of relevant classes for the ImageNet dataset was derived from the WordNet hierarchy. The WordNet hierarchy describes semantic relationships among approximately 80,000 concepts, also called synsets. For the generation of dataset samples, these synsets were used to query images from search engines that matched the corresponding concepts. Afterwards, for each category, human annotators manually selected from the automatically retrieved images only those that actually represent the category in question . (Deng et al., 2009) The samples of the ImageNet-1k validation set were derived in the same way as described for the ImageNet dataset (Russakovsky et al., 2015).

According to Russakovsky et al. (2015), the ILSVRC datasets, including ImageNet-1k, have two relevant attributes for researching vision ML models. First, they emphasize, that the included 1,000 classes represent a large variety of objects. Second, despite this large variety, there are subsets within the 1,000 classes that are suitable candidates for fine-grained classification tasks. For example, the 120 different dog breeds among the 1,000 classes require fine-grained classification by an ML model. (Russakovsky et al., 2015) These qualities make ImageNet-1k not only a suitable

benchmark for evaluating model performance but also a strong basis for evaluating OOD robustness in this work. The subset of classes used for fain-grained classification tasks are well suited for forming class pairs whose corresponding samples show, on average, high similarity. In the context of the comparison scheme applied in this work, those pairs of highly similar classes are an important prerequisite for forming pairs of *classB* and *classA*. Additionally, the variety among the 1,000 classes enables the application of the presented comparison scheme across a broad range of class types. Another advantage of the ImageNet-1k dataset is that additional validation sets with covariate-shifted examples already exist and are ready to use.

4.2 OOD Datasets

Three different OOD datasets were used: ImageNet-V2, ImageNet-R, and ImageNet-C. Each OOD dataset contains samples that have a specific covariate shift relative to the samples of the ImageNet-1k dataset. ImageNet-V2 and ImageNet-C both contain samples for all 1,000 classes of the ImageNet-1k dataset. The ImageNet-R dataset only provides samples for 200 of the 1,000 ImageNet-1k classes.

ImageNet-V2 Under the name ImageNet-V2, three complete additional validation sets for ImageNet-1k are provided. Each of these validation sets contains 10 images per class and has a total size of 10,000 images. The ImageNet-V2 dataset was created as part of a study aimed at evaluating the generalization ability of models trained on commonly used datasets, e.g. ImageNet-1k. To this end, additional test sets for the ImageNet-1k dataset were designed, expected to follow the distribution of the original training and test sets. (Recht et al., 2019)

For the creation of the ImageNet-V2 dataset the same steps used for the original ImageNet-1k dataset were followed. The data for all three datasets were drawn from the same source. Some slight differences in the selection of images from the complete set of gathered images resulted in the three different datasets of ImageNet-V2. As in the creation process of the original ImageNet-1k dataset there is one step where human annotators select only those images that actually match the semantics of the class in question. In Recht et al. (2019) each image was assessed by several annotators. The relative frequency with which an image was selected by the annotators was calculated and is referred to as the image's selection frequency in the study. The three ImageNet-V2 datasets result from selecting only those annotated images whose selection frequency fulfills a specific condition.

According to Recht et al. (2019) the three datasets of ImageNet-V2 were defined this way:

- Threshold0.7: Only those images which were selected by at least 70% of the annatators
- TopImages: The most often selected images per class
- MatchedFrequency: Those images that have a comparable selection frequency to the images of the original ImageNet-1k validation set

In their study, Recht et al. (2019) reported an almost constant shift in accuracy across these three new datasets compared to the original validation set, ranging from +2.1% to -11.8%. As a reason for this accuracy shift, they concluded that a distributional shift likely exists between the three new datasets and the original validation set. Although it was not the intention of the creators to provide a new OOD dataset with ImageNet-V2 it has been used as such, for example in Taori et al. (2020) and Noda et al. (2025).

In this work the three Imagenet-V2 datasets are regarded as OOD datasets to ImageNet-1k with a relatively small change in the covariate distribution. A unique attribute of the ImageNet-V2 datasets is that they enable the evaluation of OOD robustness on relatively smaller distributional changes, what contributes to a more general picture of a ML model's OOD robustness.

ImageNet-C This dataset is intended to be used as a standardized benchmark for image corruptions that are likely to occur in the real-world. Similarly to ImageNet-V2, ImageNet-C comprises a set of several OOD datasets. Originally, 15 different specific corruption types were provided that originate from a set of four major categories of image corruptions: *noise, blur, weather, and digital*. In addition, one corruption type per corruption category is provided to enable model adaptation. These four additional corruption types, summarized under a corruption category called *extra*, were used in this work in the same way as the other 15 corruption types. (Hendrycks and Dietterich, 2018)

As in Hendrycks and Dietterich (2018), the four corruption categories and their associated corruption types are listed below:

- noise: Gaussian noise, Shot noise, Speckle noise (extra), and Impulse noise
- blur: Defocus blur, Frosted Glass blur, Motion blur, Gaussian blur (extra), and Zoom blur
- weather: Snow, Frost, Fog, Spatter (extra), and Brightness
- digital: Contrast, Elastic Transformation, Pixelate, Saturate (extra), and JPEG compression

All ImagNet-C datasets are the result of algorithmic image manipulations applied to the ImageNet-1k validation set. For each of the 19 corruption types, there is a specific corruption algorithm that can apply the corresponding corruption type at five different severity levels to a given input image. Each algorithm was applied at each severity level on the complete ImageNet-1k validation set. As a result, for each corruption type and each corruption severity there is one OOD dataset same in size as the original validation set. In total, ImageNet-C provides 95 OOD datasets. (Hendrycks and Dietterich, 2018)

In this work all 95 ImageNet-C datasets were used for OOD robustness evaluation. The results obtained for these 95 datasets were handled in two different ways depending on the comparison case. Only for comparisons within ImageNet-C the results were categorized by corruption type. When the ImageNet-C results were compared with the those of the other OOD datasets – ImageNet-V2, and ImageNet-R – the results were categorized by severity level and averaged across all corruption types.

ImageNet-R This single dataset aims to measure the generalization ability of ML models to artificially crafted samples. The key facts for ImageNet-R are that this dataset provides in total 30,000 images for 200 classes, which form a subset of ImageNet-1k classes. (Hendrycks and Dietterich, 2019) The creators describe the dataset content as consisting of "various artistic renditions" (Hendrycks and Dietterich, 2019) what is reflected in the *R* for *Rendition* in the dataset name. The motivation behind ImageNet-R was to provide data to research if ML models can mimic the human ability to recognize objects based on line drawings as well as based on colorful photographs. In the description of the dataset creation process the limitation to 200 classes is reasonable explained. There are three attributes that make a class suitable for inclusion in ImageNet-R. First, a class should have a relatively low fraction of renditions in the original ImageNet-1k validation set. Second, there should be enough renditions available on the internet for that class. And third, renditions of a class should be visually detailed enough so that correct class samples can be identified as those by human annotators. (Hendrycks and Dietterich, 2019) After the definition of relevant classes, the class examples were derived in a manner similar to ImageNet-V2 and in ImageNet-1k, except that the search engine queries were designed to return renditions instead of photographs (Russakovsky et al., 2015; Recht et al., 2019; Hendrycks and Dietterich, 2019)

5 Experiments & Results

Before any measurements were applied, the embeddings which form the basis for the measurements had first to be computed. The computation of the embeddings was done in a completely decoupled way from the later analysis.

The work station of the chair was utilized for the computation of the embeddings and the subsequent analyses on them. The work station was equipped with a NVIDIA RTX A5000 GPU, an Intel Xeon W-2265 CPU, and a 125GB RAM. Table 2 lists the Python packages used, which are considered to have the greatest influence on the results, along with the versions of these packages.

Name	Version
python	3.12.9
pytorch	2.5.1
torchvision	0.20.1
numpy	2.0.1
timm	1.0.20

Table 2: Name and version of important packages that were used.

5.1 Data preprocessing

Set of required embeddings With respect to the ImageNet-1k dataset, it is sufficient to compute embeddings only for the validation set. For the other three datasets — ImageNet-R, ImageNet-C, and ImageNet-V2 — the complete dataset embeddings are required for subsequent analysis. The data for the ImageNet-1k dataset was provided by the chair. The data for ImageNet-R, ImageNet-V2, and ImageNet-C was downloaded from the internet. Information for downloading these can be found on the respective GitHub-sites [1].

Used DINOv2 model In this work, solely embeddings computed by the smallest distilled DINOv2 model, ViT-S/14, which has an embedding dimension of 384, were analyzed (Oquab et al., 2024). The ViT-S/14 model was retrieved via the torch module the way described in the DINOv2 repository on GitHub (din, a). For each image one CLS token and 1,369 patch tokens (the exact number of patch tokens depends on the input image size) were retrieved from the last model layer of the used backbone. The CLS token was stored without modification. From the set of the 1,369 patch tokens, only one embedding – computed as the per-dimension mean across all patch tokens - was stored.

[1] https://github.com/hendrycks/imagenet-r, https://github.com/modestyachts/ImageNetV2, https://github.com/hendrycks/robustness

Transformations The values of an encoded embedding are strongly influenced by the transformations applied to the corresponding input image. The transformation step usually comprises a sequence of image transformations applied to the input. The result of these transformations is then converted into a PyTorch tensor, which is the expected input format for the model. Of particular relevance for the DINOv2 model is the image rescaling transformation. All DINOv2 models create patches of size 14x14, which requires that the shape of an input images is a multiple of that patch size (din, b). In this work, the transformation settings provided with DINOv2 were requested via the *timm* module (Pytorch Image Models), which also provides functionality to put the sequence of transformation into code (Wightman, 2019). Only one change was made in the retrieved transformation settings. The input image size was set to 518x518, which was 224x224 in the standard settings before. One reason for that change is that the biggest DINOv2 model, ViT-g/14, was trained for a short duration on images of size 518x518 in order to enhance the model's performance on pixel-level tasks (Oquab et al., 2024). Furthermore, the input image size of 518x518 is explicitly defined in the model card on *Hugging Face* and is applied in practice (hug, 2025b; Gur-Arie, 2025).

Following transformations were defined in the *transforms.Compose* object that was passed to each PyTorch *DataLoader*:

```
Compose(
    Resize(size=592, interpolation=bicubic, max_size=None,
        antialias=True)
    CenterCrop(size=(518, 518))
    MaybeToTensor()
    Normalize(mean=tensor([0.4850, 0.4560, 0.4060]), std=
        tensor([0.2290, 0.2240, 0.2250]))
)
```

Listing 1: Printed transforms.Compose object which defines the image transformations that were applied on each processed input image.

Storing the computed embeddings The computed embeddings were stored in dataset-specific dictionaries. The keys of such a dictionary were defined by the datasets' class names in form of WNIDs (a shorthand used to reference ImageNet-1k class unambigiously). Each key stores the embedding results of samples belonging to the class identified by the corresponding WNID. The dictionary with the results of the ImageNet-C benchmark had a different structure. For the subsequent analysis it was of advantage to create a nested dictionary for the results of the ImageNet-C benchmark. At the basic dictionary level for each of the 19 corruption types a dictionary was created. Each of those 19 dictionaries contained five different dictionaries, one for each severity level. Each of those five dictionaries contained the embeddings for one specific corruption type at one specific severity level. Python's pickle module was utilized for storing and loading the dictionaries containing the

computed embeddings (pyt). The storage size required per dataset and the set of embeddings coressponding to each dataset is listed in Table 3.

Datasetname	Storage size dataset	Storage size embeddings
ImageNet-1k (only val.-set)	6.7 GB	150 MB
ImageNet-R	2.2 GB	90 MB
ImageNet-V2-MF	1.3 GB	30 MB
ImageNet-V2-70	1.3 GB	30 MB
ImageNet-V2-TOP	1.3 GB	30 MB
ImageNet-C	83 GB	14 GB

Table 3: Storage size per used dataset and embeddings computed per dataset.

5.2 Application of the Comparison Scheme

In the next step of the experiments, the described comparison scheme was applied on the set of computed embeddings. The task here was to reference each general class of the comparison scheme to a certain class from a certain dataset. The embeddings corresponded to the referenced class were then used in the comparisons in-place of the general class.

The process of referencing the general classes of the comparison scheme to dataset classes is termed here as the initialization of the comparison scheme. A concrete initialization of the comparison scheme is termed as a comparison sequence. In total 200 different comparison sequences were defined in this work. A small selection of the derived comparison sequences is shown in Table 5. Next, follows a step-by-step explanation on how the 200 different comparison sequences were derived.

Initialization Procedure For each of the 200 comparison sequences, at first class-A and class-A' were initialized. Based on the initializations of class-A and class-A', class-B, and class-R were initialized subsequently. Class-Mean wasn't initialized for each comparison sequence as its value remained fixed across all comparison sequences.

Initialization of Class-A and Class-A' In each comparison sequence class-A was initialized with a specific class from ImageNet-1k. Class-A' subsumed all classes of ImageNet-R, ImageNet-V2 and ImageNet-C with the same WNID as the initialization of class-A. For each ImageNet-1k class that was used as initialization for class-A there was one class in ImageNet-R, three classes from the ImageNet-V2 datasets, and 95 classes from the ImageNet-C datasets with the same WNID. Consequently, 99 classes were referenced by each instantiation of class-A'. For example, in one comparison sequence class-A was initialized with the cabbage class

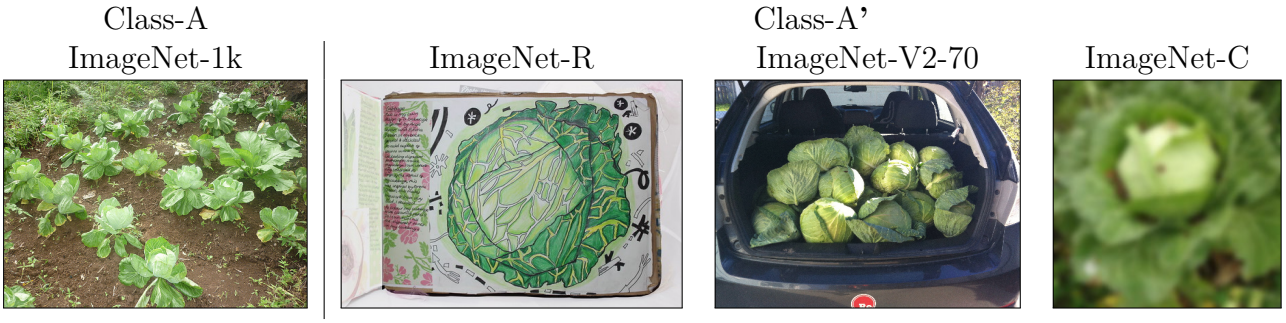

Figure 2: Samples of classes which were instantiated as class-A and class-A' in the example comparison sequence, respectively.

from ImageNet-1k. In the same comparison sequence, class-A' referenced all cabbage classes from ImageNet-R, ImageNet-V2, and ImageNet-C. A small fraction of the samples that correspond to this specific initializations of class-A and class-A' is visualised in Figure 5.2.

The initialization of class-A and class-A' requires that each of the four datasets - ImageNet-1k, ImageNet-R, ImageNet-V2, and ImageNet-C – contains at least one class corresponding to the same WNID. ImageNet-R contains only 200 classes, in contrast to the other two OOD datasets, which provide samples for all 1,000 ImageNet-1k classes. Consequently, the set of WNIDs corresponding to the 200 classes of ImageNet-R was used to define the initializations of class-A and class-A' in all comparison sequences.

Figure 3: Samples of selected class-B per similarity measure for the example comparison sequence

Initialization of Class-B In the general comparison scheme class-B is regarded as the most similar ID class to class-A. All four discussed similarity measures were used to find the most similar ID class to a specific class-A instantiation. In practice, for each class-A instantiation the similarity to the remaining 999 ImageNet-1k classes was measured using each of the four similarity measures separately. Consequently, for each class-A instantiation four class-B instantiations were determined, one for each measure used, which are generally termed as: class-B-L1, class-B-C1, class-B-C2, and class-B-C3. In the later described comparisons the measurement results of

class-A' were related to the measurement results of class-B. The class-B instantiation used depended on the measurement that was used. For example the results for the L1 distance between class-A' and class-A, were only set into relation to the L1 distances between class-B-L1 and class-A. In the comparison sequence in which class-A was instantiated with the cabbage class following instantiations for class-B were derived: spotlight (class-B-L1), plant pot (class-B-C1), bee (class-B-C2), cauliflower (class-B-C3). Visual examples for these four different class-B instantiations can be found in Figure 3.

	L1 Measure	C1 Measure	C2 Measure	C3 Measure
L1 Measure	-	23	33	69
C1 Measure	-	-	39	77
C2 Measure	-	-	-	72

Table 4: Number of times a different ImageNet-1k class was determined as class-B instance when comparing results for two specific measures.

When comparing the four measure-specific class-B instantiations across all comparison sequences, two different similarity measures often identified the same ImageNet-1k class as most similar ID class to a class-A instantiation. Table 4 shows how many times two different similarity measures derived different class-B instantiations for the same class-A instantiation.

Initialization of Class-R By definition class-R is an arbitrary ID class that is neither the same as class-A nor class-B. Through the determination of a class-B instantiation per similarity measure there are now up to four different instantiations for class-B in a specific comparison sequence. These different instantiations for class-B were also considered when defining class-R. A concrete class-R was drawn randomly from a set of potential ID classes. This set was defined as the set of all ImageNet-1k classes excluding the instantiation of class-A and all class-B instantiations. For the comparison sequence with the cabbage class as class-A instantiation class-R was drawn randomly from the set of all ImageNet-1k classes excluding: cabbage (class-A), spotlight (class-B-L1), plant pot (class-B-C1), bee (class-B-C2), cauliflower (class-B-C3). For this comparison sequence the Irish setter class was randomly selected as the instantiation of class-R (example samples shown in Figure 5.2). Before the first class-R was instantiated, the random seed was fixed in order to make the class-R assignments repeatable in subsequent second runs.

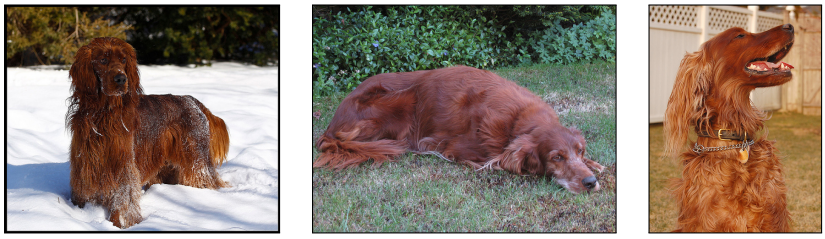

Figure 4: Samples of the Irish setter class from ImageNet-1k, which was selected as class-R in the example comparison sequence.

Initialization of Class-Mean In contrast to the other general classes, class-Mean does not reference different classes across comparison sequences. Instead, class-Mean represents the complete ImageNet-1k validation set through a single mean embedding. In practice, this mean embedding was computed once over the set of embeddings corresponding to the ImageNet-1k validation samples and then reused in all comparison sequences.

Class-A & Class-A'	Class-B-L1	Class-B-C1	Class-B-C2	Class-B-C3	Class-R
Border Collie	collie	collie	collie	collie	southern black widow
cockroach	cricket insect	cricket insect	bee	ground beetle	computer mouse
lab coat	stethoscope	stethoscope	syringe	Windsor tie	crate
cabbage	spotlight	plant pot	bee	cauliflower	Irish Setter
goldfish	gar fish	gar fish	gar fish	axolotl	lawn mower
starfish	sea cucumber	sea cucumber	sea cucumber	brain coral	prayer rug
canoe	paddle	paddle	paddle	paddle	dhole
great egret	crane bird	crane bird	crane bird	spoonbill	paintbrush

Table 5: Selection of 8 comparison sequences from the set of 200 in total comparison sequences.

5.3 Comparison Results

In the methodology section was already defined which combination of classes of the comparison scheme are compared per similarity measure. This general definition defines in consequence also the comparisons made in each comparison sequence. Consequently, the general structure of made comparisons is the same for all comparison sequences. By hand of one of the 200 comparison sequences, the comparisons made per comparison sequence are demonstrated in an explicit way. Subsequently, the aggregated results from all comparison sequences are presented.

5.3.1 Results of a single Comparison Sequence

The previously described comparison sequence with the cabbage class as class-A instantiation is used for illustrating the general structure of made comparisons per comparison sequence. The presentation of made comparisons is structured according used similarity measures. At first the comparison results when using the L1 measure are presented. Subsequently, then the results of the comparisons that were based on the C1 measure. Finally, the comparison results based on the C2 and C3 measures are presented together.

L1 Measure Results In 7 was already defined on which pairs of general classes the L1 measure is to be applied to. For the example comparison sequence the L1 measure was in consequence then applied on following pairs of classes:

$(cabbage, \textbf{cabbage}), (cabbage, spotlight),$
$(cabbage, IrishSetter), (cabbage, ImageNet - 1k - Mean)$

The bold **cabbage** denotes a cabbage class originating from one of the OOD datasets. The number of cabbage classes with distribution shift determines the number of times the L1 measure was applied to the combination $(cabbage, \textbf{cabbage})$. There was one cabbage class in the ImageNet-R dataset, one in each of the three ImageNet-V2 datasets, and one in each of the 95 ImageNet-C datasets. Each of the 95 ImageNet-C datasets differs either in the corruption type or the corruption severity of the samples. In total, there were 99 cabbage classes with a distribution shift. Consequently, the L1 measure was applied to 99 distinct combinations of the type $(cabbage, \textbf{cabbage})$. Generally, for each of the 200 comparison sequences there were 99 different combinations of the type $(classA, classA')$.

In 5 the L1 measure results for the example comparison sequence are visualised. The achieved L1 measure values are scaled along the y-axis. Each bar represents the result of the L1 measure between the cabbage class and another class as defined in the described combinations. One exception is the measurement result of $(cabbage, ImageNet - 1k - Mean)$, which is visualised as a red horizontal line. The correspondence between each bar in the diagram and the previously described combinations of classes is the following. In all combinations, there is the case that the similarity between the cabbage class and another class was measured. Therefore, in the diagram, only the class that changed with each combination is denoted. A result can be uniquely related to a specific class by combining the colour of a bar with the information on the x-axis. The colour of a bar generally indicates its reference to the general class in the scheme. The spotlight class is the initialization of class-B for the L1 measure in this comparison sequence, which is why the corresponding bar is blue. The bar corresponding to the Irish Setter class is orange because it represents class-R in this comparison sequence. All green bars belong to class-A'. The dataset denoted at the bottom of a bar indicates the origin of the class whose results are represented by the bar. The datasets denoted along the x-axis are especially relevant

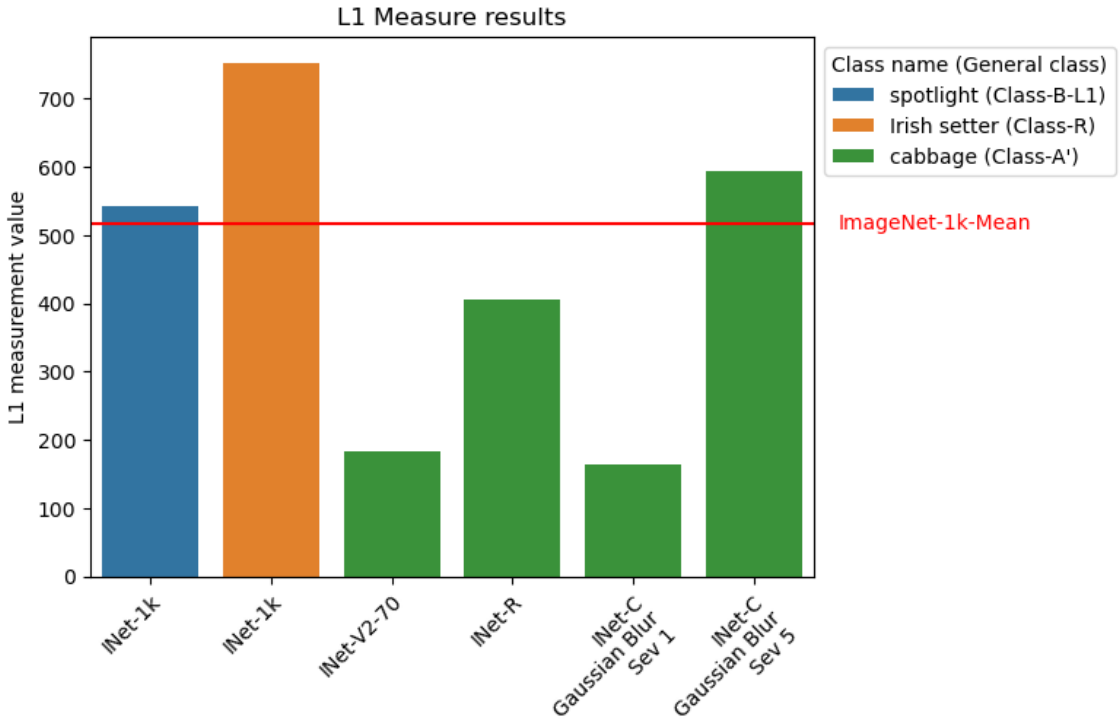

Figure 5: Diagram that visualizes the L1 measure results of the example comparison sequence.

to identify the results of the 99 cabbage classes with distribution shift. In this diagram, a small selection of four out of the 99 cabbage classes with distribution shift is presented. The first green bar from the left corresponds to the cabbage class from the ImageNet-V2 dataset, which samples have a selction frequency of 70% or higher. This ImageNet-V2 dataset is termed as ImageNet-V2-70 from here. The next green bar corresponds to the cabbage class in the ImageNet-R dataset. The last two green bars correspond to the result of the cabbage classes that originated from two of the 95 ImageNet-C benchmark datasets. These two classes result from corrupting the original cabbage class (class-A) with Gaussian blur. The second-to-last green bar shows the result of the cabbage class generated by applying the corruption at the mildest level, while the final green bar shows the result when the corruption was applied at the most severe level.

C1 Measure Results In 9 is defined on which combinations of general classes the C1 measure was applied to.
For the example comparison sequence these were:

(cabbage, **cabbage**, INet-1k-Mean), (cabbage, plant pot, INet-1k-Mean),
(cabbage, Irish Setter, INet-1k-Mean)

The meaning of the cabbage class written in bold is the same as described for the L1 measure. Consequently, also the C1 measure was used to measure the similarity between all 99 cabbage classes with distribution shift and the cabbage class from the ImageNet-1k validation set.

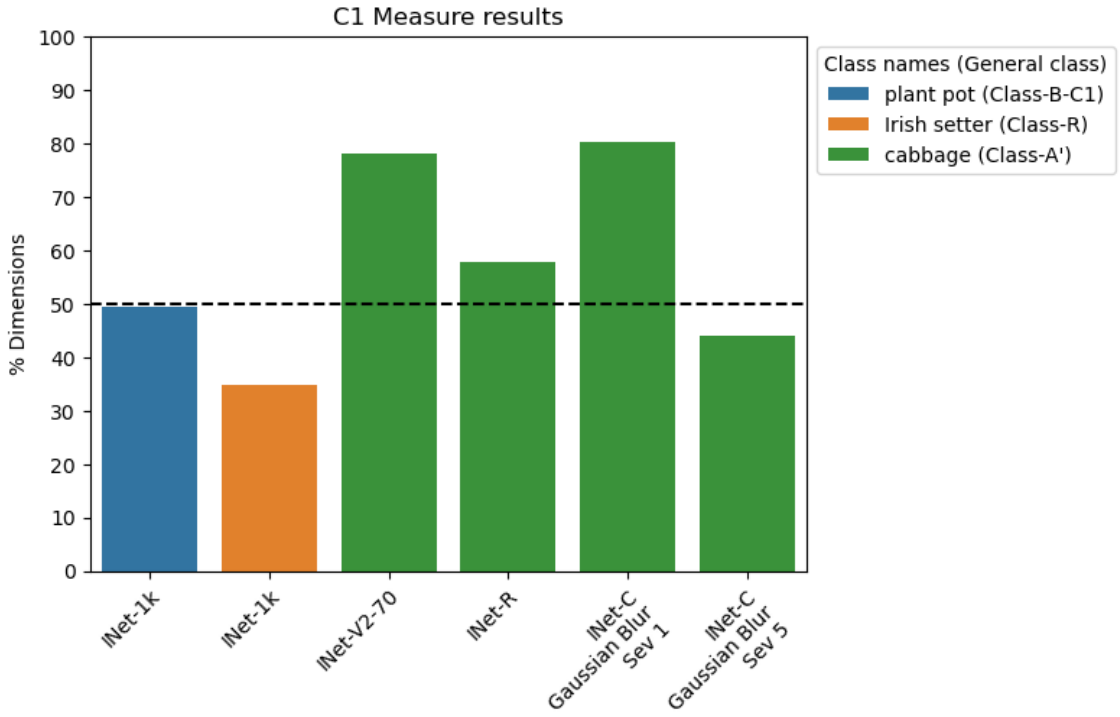

Figure 6: Diagram that visualizes the C1 measure results of the example comparison sequence.

Figure 6 visualises the C1 measure results for this example comparison sequence. The y-axis represents normalized C1 measure values. For normalization, the C1 measure values were divided by the embedding size and then multiplied by 100, resulting in percentages of the embedding size. Similar to the L1 measure diagram, the datasets denoted along the x-axis indicate the origin of the classes whose results are represented by the corresponding bars. The dotted line marks the 50% line, dividing the values on the y-axis into two halves. For all compared classes that achieve a value below the dotted line ImageNet-1k-Mean is closer to the mean embedding of the cabbage class for more than half of the embedding dimensions. In the converse case, the compared class is closer to the mean embedding of the cabbage class for more than the half of embedding dimensions. The colours of the bars have the same meaning as in the L1 measure diagram, which is denoted in the upper right legend of the diagram. In contrast to the L1 measure diagram, in this diagram the result of the plant pot class and not of the spotlight class is visualised as the class-B result. Except for this change in class-B, all other classes whose results are visualised in this diagram, are the same as in the L1 measure diagram 5. The

orange bar represents the C1 measure value of the Irish Setter class. The green bars show the C1 measure values of the same cabbage classes with distribution shift as in the L1 measure diagram.

C2 and C3 Measure Results In practice the C2 and C3 measure values were computed together. When the C2 measure value for some sequence of inputs was computed, the C3 measure value for this input sequence was computed as well. In 12 is defined on which combination of general classes the C2 and C3 measures were applied to.

In practice, for the example comparison sequence the instantiation of following input sequences was the same for the C2 and for the C3 measure:

$(classA, classA, classA', classMean), (classA, classA, classR, classMean)$
, which were instantiated for both measures with:
(cabbage, cabbage, **cabbage**, INet-1k-Mean), and
(cabbage, cabbage, Irish Setter, INet-1k-Mean)

The meaning of the **cabbage** in bold is the same as described in the application of the L1 measure and the C1 measure on the example comparison sequence.

Different between C2 and C3 measure was the instantiation of the input sequence $(classA, classA, classB, classMean)$, because the C2 and C3 measures identified two different class-B instantiations for this comparison sequence.

For the C2 measure the corresponding instantiation was:
(cabbage, cabbage, bee, INet-1k-Mean)

, and for the C2 measure the corresponding instantiation for was:
(cabbage, cabbage, cauliflower, INet-1k-Mean)

The results of the C2 and C3 measure are visualised in Figure 7. Each bar represents the C2 and C3 measurement values of a class. Along the x-axis, the denoted datasets indicate the origin of the class whose result is represented by the corresponding bar. The value scaled along the y-axis will become clear after explanation of the colours within a bar. In essence, the colours within a bar represent the proportion of dimensions within the compared embedding that fulfill a certain condition. The condition represented by each colour is specified in the right lower legend. The three conditions are mutually exclusive and together cover all possible cases. Consequently, each bar that is comprised of three different coloured parts is comparable to a complete embedding. The 0-value on the y-axis stands for not one dimension, whereas a value of 100 stands for all dimensions of the embedding. The C3 measure applies the exact same condition, which is denoted by the dark red colour in the legend. The C2 measure applies the disjunction of the conditions, which are denoted by the dark red colour and the orange red colour in the legend. The height of the dark red part within a bar represents the C3 measure value. The C2 measure value is represented by the stacked bar composed of the orange red and dark red part of a

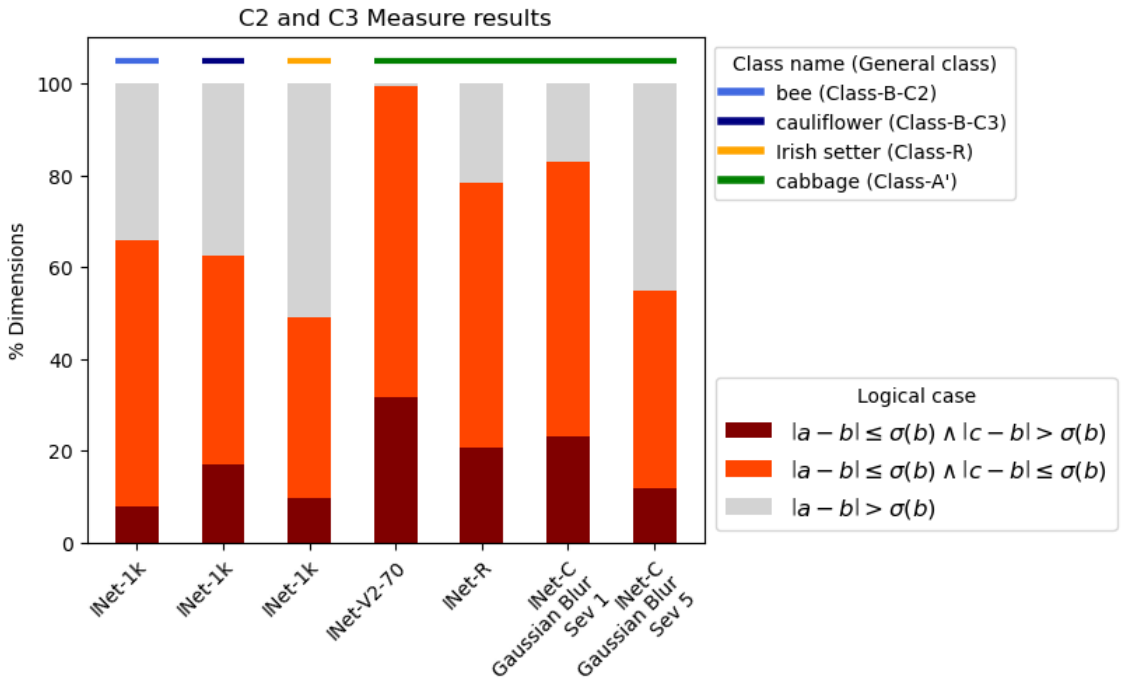

Figure 7: Diagram that visualizes the C2 and C3 measure results of the example comparison sequence.

bar. In other words the C2 measure value of a class corresponds to the top of the orange red part of a bar in the diagram. The remaining grey part of a bar has no meaning for the C2 and C3 measure values. The fat coloured horizontal lines above the stacked bars relate each bar to general classes of the comparison scheme. The first two bars correspond to the instantiations of class-B specific to the C2 and C3 measure, respectively. The first bar represents the results of the bee class, which is the instantiation of class-B derived through the C2 measure. The second bar represents the results of the cauliflower class, which is the instantiation of class-B derived through the C3 measure. The third bar represents the results of the Irish Setter class, which is the instantiation of class-R in this comparison sequence. From the bar to the right of the Irish Setter class to the last bar in the diagram, the result of the classes that correspond to class-A' are visualised. The classes that correspond to the these last four bars in the diagram align exactly with those whose results were already presented in the L1 measure diagram and in the C1 measure diagram.

5.3.2 Results across all Comparison Sequences

In the previous section the comparisons conducted on a specific comparison sequence from the set of in total 200 comparison sequences was described in detail. For the remaining 199 comparison sequences the general procedure of conducted compar-

isons was the same. The only difference from comparison sequence to comparison sequence was the initialization of the comparison scheme with concrete classes.

The comparison results of the remaining 199 comparison sequences are not discussed in the same detail as in the previous section. Instead, the average results across all 200 comparison sequences are presented.

Averaging of the results For each general class the average result for the L1, C1, C2, and C3 measure was computed across all comparison sequences. For instance, the value shown for class-R in the L1 measure diagram (see Figure 8) represents the average L1-measure result across all class-R instantiations. The same principle is true for the values of class-R in the C1 measure diagram and in the C2 and C3 measure diagram. In the same way, the average results per measure result for class-A', class-B, and ImageNet-1k-Mean were computed. In case of class-A', the mean result was computed per OOD dataset separately. Consequently, for each measure there was one average value for ImageNet-R, one average value for each of the three ImageNet-V2 datasets, and one for each of the 95 for ImageNet-C datasets. In Figures 8, 10, and 12 the 95 results of the ImageNet-C datasets were grouped per severity level. For ImageNet-R and the three Imagenet-V2 datasets the averaged results are presented as is. In case of class-B, the averaged results of class-B-L1, class-B-C1, class-B-C2, and class-B-C3 are shown in the respective diagrams.

L1 Measure Results In Figure 8 the average results for the L1 measure are shown. The general structure of this diagram is similar to the corresponding diagram that visualises the L1 measure results of the example comparison sequence (shown in Figure 5).

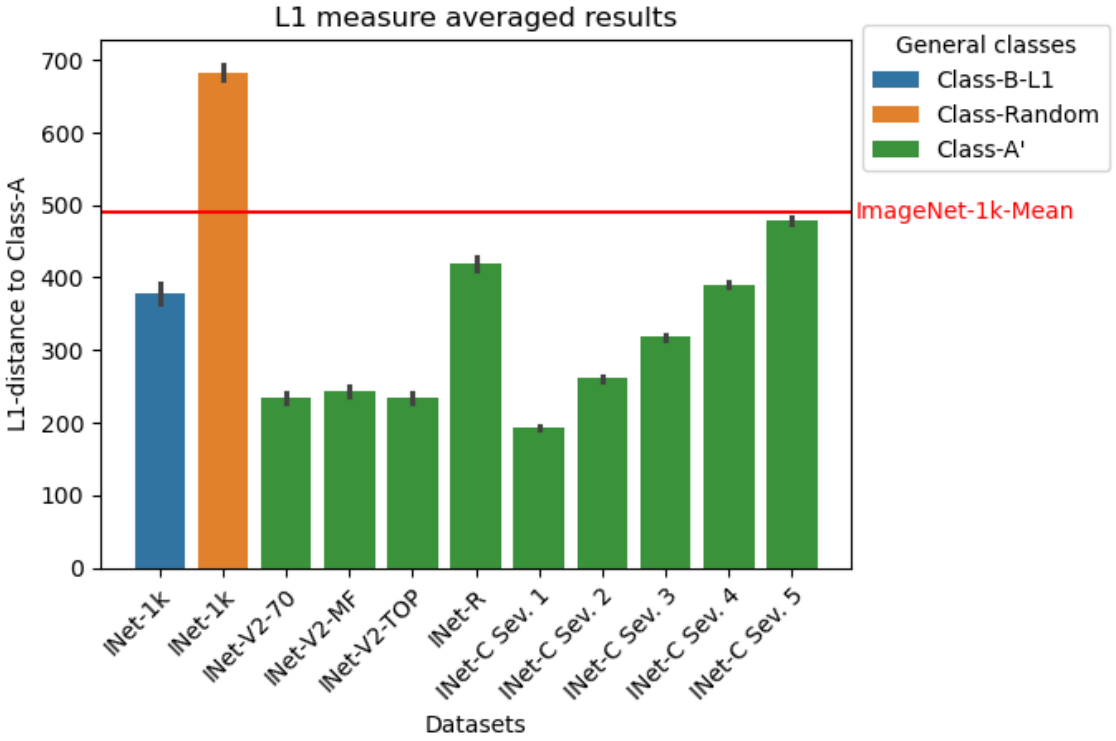

Figure 8: Averaged L1 measure results

In short, the explanation for this diagram is as follows: The y-axis represents the L1 measure values. On the x-axis the dataset is denoted indicating the origin of the samples whose averaged result is represented by the corresponding bar. The colour of a bar indicates the relationship between the represented result to the general class of the comparison scheme. This diagram deviates in the representation of the results of class-A' from the corresponding diagram with the L1 measure results of the example comparison sequence. Between the bar corresponding to class-R and the bar corresponding to the results of ImageNet-R samples there are now two additional bars. Both of these two additional bars correspond to the two ImageNet-V2 datasets, which results were not shown in the previous diagrams. One additional bar was annotated with Inet-V2-MF. This annotation refers to the ImageNet-V2 dataset which samples have the same selection frequency as the samples of the original ImageNet-1k validation set. The other additional bar was annotated with Inet-V2-TOP. This annotation refers to the ImageNet-V2 dataset which samples have the highest selection frequency. The results of the 95 ImageNet-C datasets are represented by the five bars at the right end of the diagram. Each of these five bars represents the average result across all corruption types at the same severity level. The severity level the represented result corresponds to is indicated by the last part of label on the x-axis.

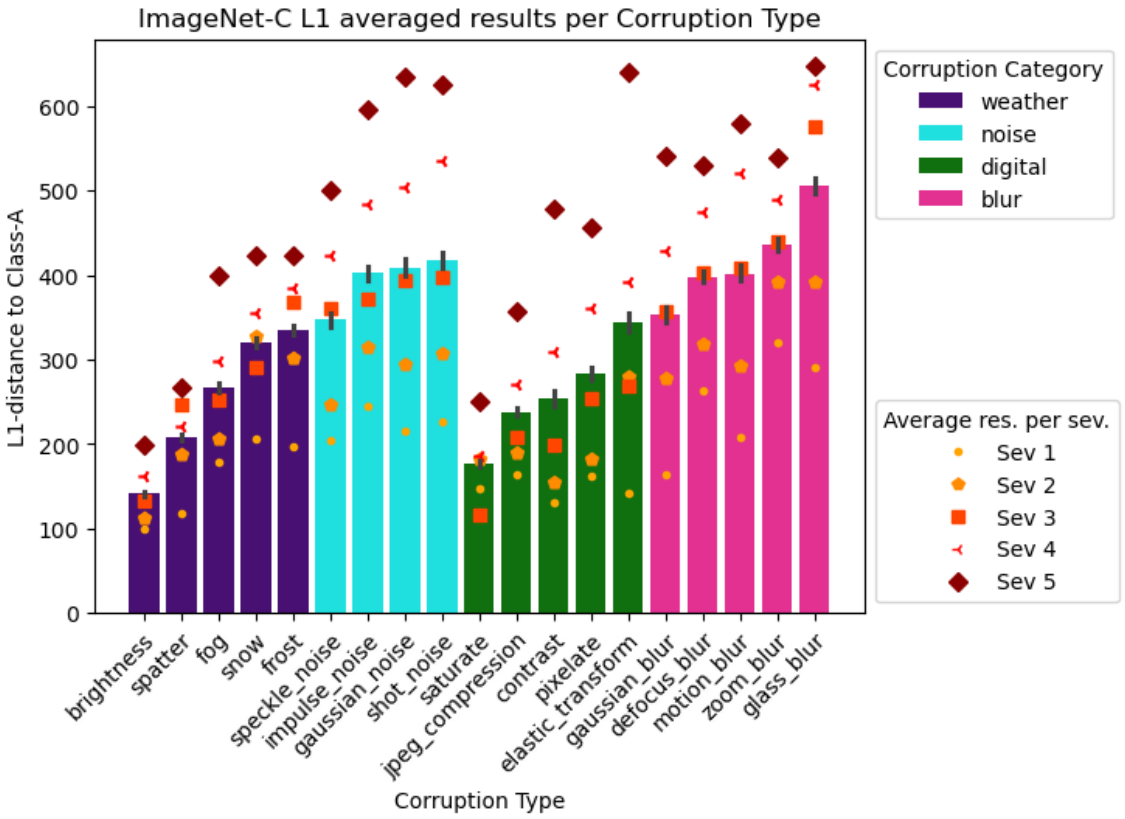

Figure 9: Averaged L1 measure results of the ImageNet-C benchmark categorized by corruption type

Figure 9 visualises the L1 measure results of the 95 ImageNet-C datasets categorized by corruption type. Each of the 19 bars in the diagram represents the L1 measure result of a specific corruption type averaged across all 200 comparison sequences and averaged across all five severity levels. The corruption type corresponding to each represented result is denoted at the bottom of the bar. The y-axis represents the L1 measure value. A bar's colour indicates the corruption category a bar's corruption type belongs to. The exact meaning of each colour is denoted in the legend. There is no special meaning behind the order of corruption categories. However, the order of corruption types per corruption category follows a certain order. Within a group of corruption types of the same corruption category, the corruption type with the lowest L1 measure result is placed at the leftmost position. The corruption types of the same corruption category are arranged from left to right according to their results. Vertically aligned to each bar there are five marks which deviate in form and colour. Each of those five marks per bar visualises the average result of the corresponding corruption type at a specific severity level. The severity level represented by a specific marker is denoted in the legend in the lower right.

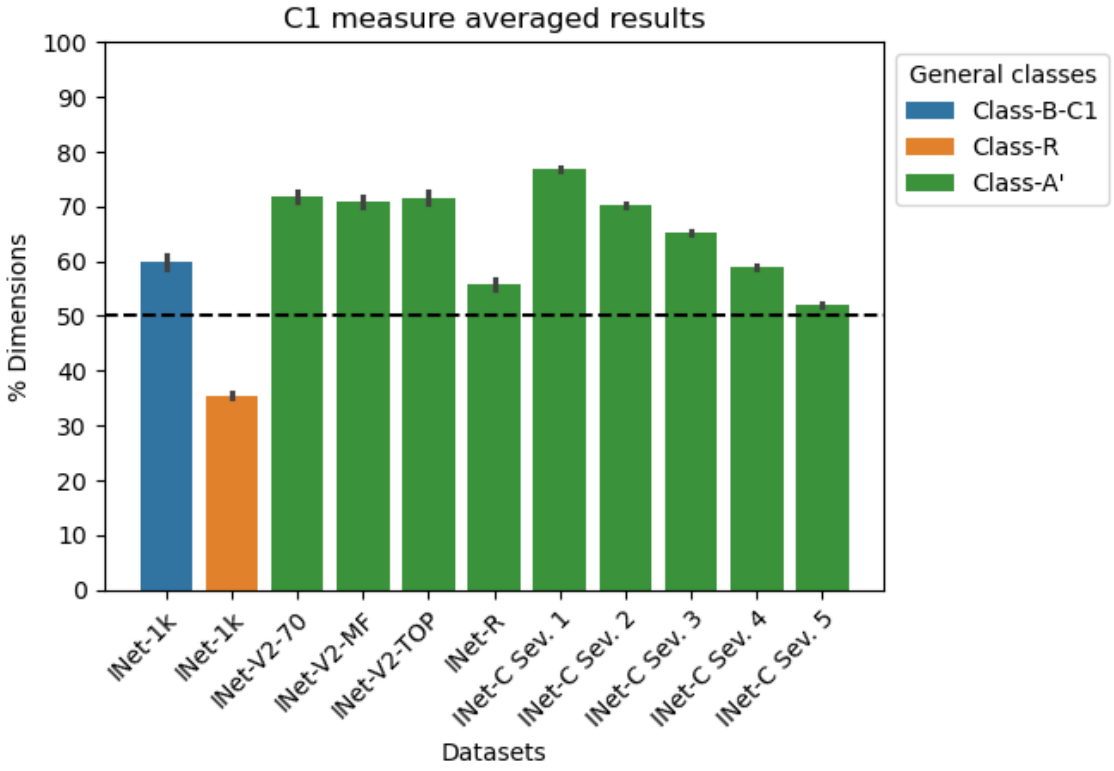

Figure 10: Averaged C1 measure results

C1 Measure Results In Figure 10 the averages of the C1 measure values are visualised. The y-axis in this diagram represents the normalized C1 measure values as it is also the case in the diagram with the C1 measure values of the example comparison sequence (Figure 6). The x-axis labels and the meaning of each colour correspond with those in the diagram representing the L1 measure values shown in Figure 8. The meaning of the dotted line, which is also shown in Figure 6, corresponds to the description provided there.

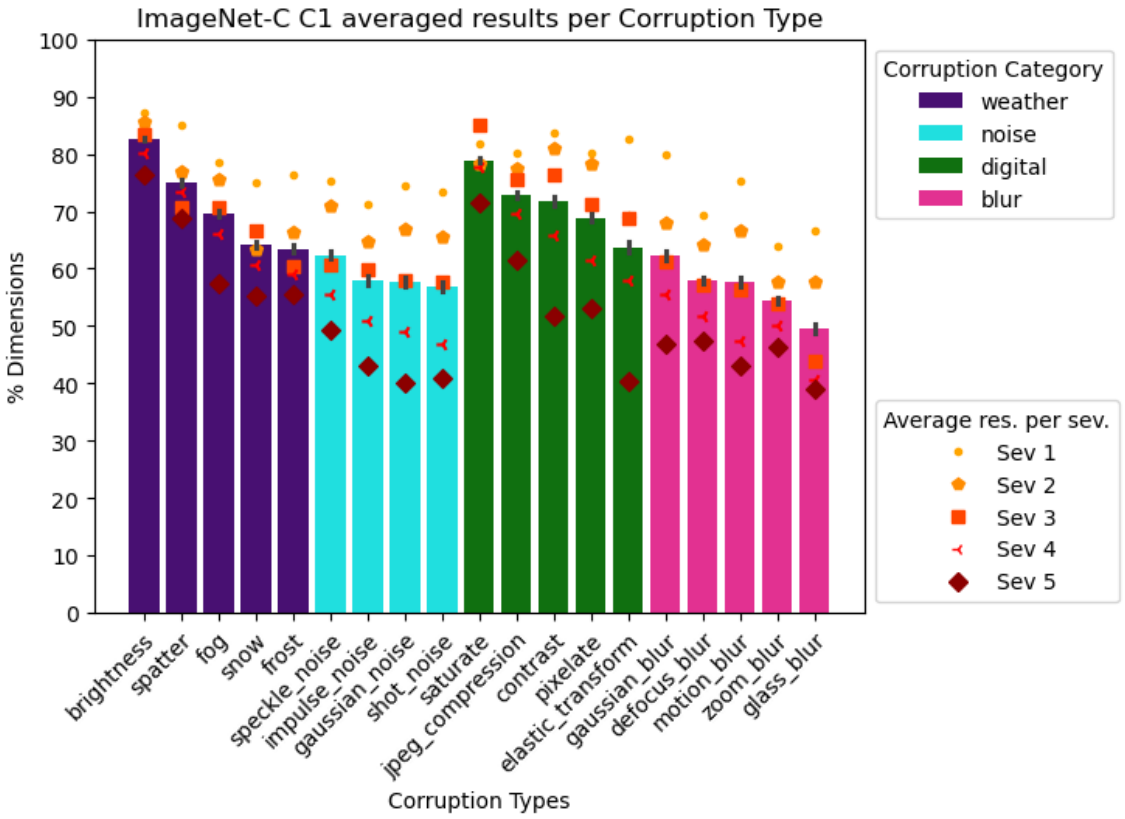

Figure 11: Aggregated C1 measure results of the ImageNet-C benchmark categorized by corruption type.

Figure 11 represents the averaged C1 measure values of the ImageNet-C datasets categorized by corruption type. This figure is almost identical to Figure 9. The only difference is that Figure 9 scales the L1-measure values along the y-axis, whereas this figure scales the normalized C1-measure values along the y-axis.

C2 and C3 Measure Results Figure 12 visualises the average C2 and C3 measure results over all comparison sequences. The axes in this diagram represent the same information as the corresponding diagram visualizing C2 and C3 measure values of the example comparison sequence (see Figure 7). The y-axis represents the normalized C2 and C3 measure values. The labels on the x-axis denote the datasets that are associated with the results represented by the corresponding bar. Similarly to the diagrams representing the averaged results of the L1 measure and C1 measure, this diagram also includes the results of the two additional ImageNet-V2 datasets and the complete ImageNet-C benchmark results. Each colour within a bar is connected with a conditional statement that is denoted in the lower-right legend. The horizontal coulourful line on top of a bar, connects this bar with a certain general class of the comparison scheme. The interpretation of the colour of each horizontal line is explained in the upper-right legend. The first two bars visualise

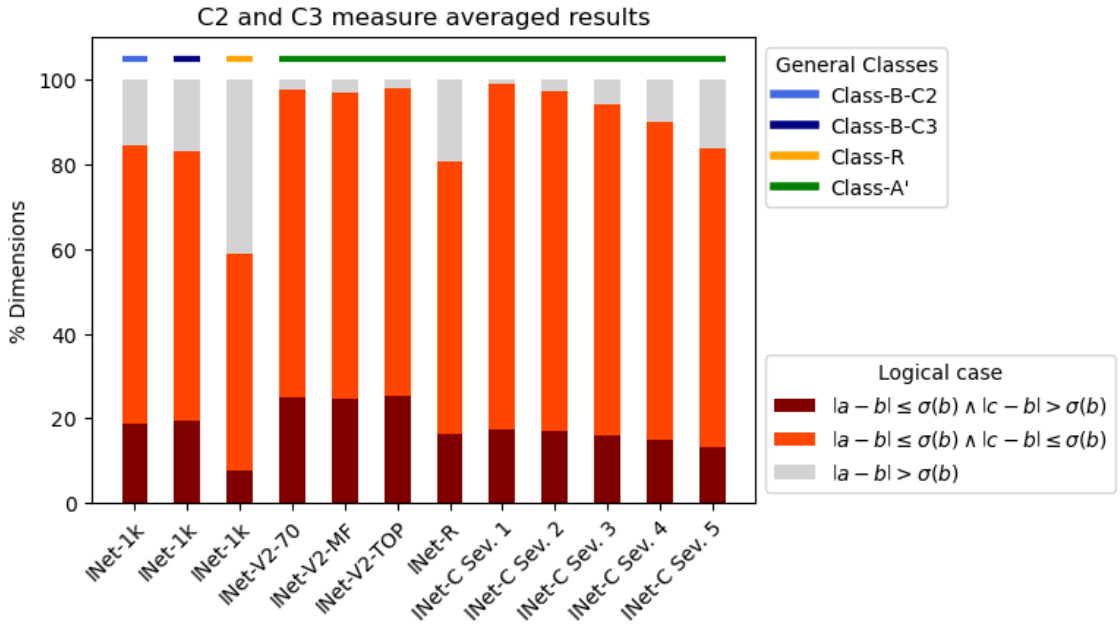

Figure 12: Averaged C2 and C3 measure results

the average results of class-B instantiations derived through two different measures. The first bar corresponds to those class-B instantiations determined through the C2 measure. The second bar corresponds to those class-B instantiations determined through the C3 measure. The order of the remaining results is the same as in the diagrams showing the average L1 and average C1 measure results. For the C2 and C3 measure values of the ImageNet-C benchmark there is no additional diagram included showing the measure results categorized by corruption.

5.4 Linear Classifier Results

In order to test the relation between classification results and the reported measurement results from the previous section, logistic regression models were utilized in a conclusive experiment. Two logistic regression models were used, both predicting the ImageNet-1k classes from ViT-S/14 embeddings. The two models used in the experiments were initialized with the parameters of a classification head for ViT-S/14, obtained from the official DINOv2 GitHub repository. The classification head from the repository was trained on ImageNet-1k.

Retrieved were a single weight matrix of dimension (1000 x 786) and a bias vector of dimension (1000, 1). These two elements are generally used for the initialization of a logistic regression model that is composed of a linear layer with 786 input dimensions and 1000 output dimensions. This logistic regression model takes as input the concatenation of the CLS token and the mean of the patch-token embeddings from the final layer. The model returns a 1,000-dimensional vector containing probabilities

that correspond to ImageNet-1k classes. This provided classifier makes prediction partly based on the patch-tokens what contradicts with the purpose of this work, which is to evaluate solely the CLS token output. Moreover predictions that were made on the patch-tokens cannot be set directly into relation to the measurement results, which were solely made on the CLS token output. Therefore, in addition to the classifier in the intended form, a second classifier was created that uses only the CLS token as input for prediction. This second classifier was initialized with the portion of the weight matrix (dimensions (1000 x [0:384])) that connects the output of the CLS token representation to the output logits. No modifications were made on the bias parameters.

From here on the classifier utilizing the CLS token output and the output of the patch tokens is abbreviated as classifier-CLS-PA. The other classifier, which uses only the CLS token output for prediction, is hereafter abbreviated as classifier-CLS.

The two classifiers were used to compute the accuracy for each OOD dataset which data was used for class-A' in the measurements. In practice, the precomputed embeddings of ImageNet-V2, ImageNet-R, and ImageNet-C were used as input to the classifiers. The results of the 95 different ImageNet-C datasets were again aggregated per severity level.

Datasets	Measures				Accuracy	
	C2 measure	C3 Measure	C1 Measure	L1 Measure	CLS	CLS+Patch
INet-C-Sev-1	0.990543	0.174890	0.768167	193.948395	0.677464	0.739889
INet-C-Sev-2	0.975055	0.169819	0.702090	261.079498	0.615355	0.673617
INet-C-Sev-3	0.943668	0.160773	0.651534	318.191406	0.548516	0.599708
INet-C-Sev-4	0.902001	0.147889	0.589387	390.724548	0.456148	0.497449
INet-C-Sev-5	0.839912	0.131853	0.520074	478.026642	0.338309	0.367285
INet-R	0.807253	0.163854	0.559063	418.955475	0.424333	0.406200
INet-V2-70	0.977057	0.250560	0.716953	233.969543	0.729400	0.796900
INet-V2-MF	0.970885	0.247018	0.709674	243.640472	0.668600	0.727700
INet-V2-TOP	0.978945	0.252669	0.715417	234.731995	0.763500	0.832600

Table 6: Mean results for the C1, C2, C3 and L1 measure per OOD dataset combined with accuracy results per OOD dataset.

Table 6 presents the accuracy results of both classifiers together with the mean results of the L1, C1, C2, and C3 measure per dataset. The results in each row correspond to the results of one specific OOD dataset. The first four cells contain the mean results of the four discussed measures (the results for C1, C2, and C3 are normalized by the embedding size). The last two cells in a row contain the accuracies achieved by the two classifiers. The background colour of a cell indicates the ranking of the results within a column. The cells with the purest yellow as background colour are considered as the ones with the best values per column. On the contrary, cells with dark blue as background colour are considered as the one with the worst values per column. Only for the L1 results column, lower values indicate a better performance. For the remaining columns a higher value is considered better.

6 Discussion

The purpose of this work was to analyse the impact of covariate shifts in the data on the embedding space of the DINOv2 ViT. In the following, the results obtained from the experiments are evaluated with respect to the research goal. At first, the relationship between the averaged results of class-B, class-R, and ImageNet-1k-Mean is analysed, as visualised in Figures 8 , 10 and 12. The results of these three form the basis for the subsequent evaluation of the class-A' results. The ImageNet-C results, which are represented by corruption type in Figures 9 and 11, are analysed separately. Finally, the class-A' results for all measures are compared to the accuracies achieved by the logistic regression models, as presented in Table 6.

Results for class-B, class-R, and ImageNet-1k-Mean When comparing the results of class-B with those of class-R, it is evident that class-B achieves significantly higher similarity values than class-R. In general, the clear difference between the results of class-B and class-R confirms the approach of evaluating the results of class-A' in relation to those of class-B. For the L1 and C1 measures, ImageNet-1k-Mean achieves higher similarity values than class-R but lower values than class-B. For the C2 and C3 measures, it is not possible to compare the results of class-B and class-R with ImageNet-1k-Mean, as no explicit results for ImageNet-1k-Mean are shown. These results indicate that ImageNet-1k-Mean is not highly similar to the mean embedding of some arbitrary ImageNet-1k class, but is still significantly more similar than a randomly selected class.

Class-A' patterns Several general patterns can be observed when analyzing the results of class-A-prime in more detail. One observation is that the results for the three ImageNet-V2 datasets are very similar across all four measures. A more detailed comparison among these three datasets reveals that ImageNet-V2-MF achieves regularly lower results than the other two ImageNet-V2 datasets. This aligns with reported accuracy drops on these three datasets. Recht et al. (2019) for example reports the biggest change in TOP-1 accuracy for ImageNet-V2-MF (-11.8%) compared with ImageNet-V2-70 (-3.2%) and ImageNet-V2-TOP (+1.8%). Another common pattern is the consistent deterioration of the ImageNet-C results with increasing severity level of the data. This concludes that the severity of a distribution shift is reflected by all measures.

Class-A' results relative to class-B results When comparing the class-A' results with the class-B results than the analysis for ImageNet-V2 and ImageNet-R is clearer than for ImageNet-C. For all four measures, all three ImageNet-V2 datasets achieve better results than class-B. Conversely, the ImageNet-R results fall below the results of class-B for all four measures. These results are reflected in the results of the classifiers. On ImageNet-R classifier-CLS achieved an accuracy of 42% and

classifier-CLS-PA an accuracy of 41%. For the ImageNet-V2 datasets the accuracies of both classifiers range between 67% and 83%.

The relation between the class-B result and the results of the ImageNet-C datasets deviates more by measure. For the L1 and C1 measures the result of class-B is between the results of ImageNet-C-Sev-3 and ImageNet-C-Sev-4. In contrast, for the C2 measure the result of ImageNet-C-Sev-5 seems almost equal to the result of the corresponding class-B. The biggest change for the ImageNet-C results is observable for the C3 measure. For this measure all ImageNet-C results are below the value of class-B-C3. These differences per measure can be interpreted as indicating that each measure evaluates the impact of the image corruptions on the embedding space differently. The C2 measure reflects only small changes in the embedding space that are induced by image corruptions. Thereafter equally follow the L1 and C1 measures. The largest changes in the embedding space induced by image corruptions are measured by the C3 measure.

ImageNet-C specific results Both diagrams visualising the ImageNet-C results per corruption type, one visualising the L1 measure results and the other the C1 measure results, show similar patterns. Therefore the results of both diagrams are discussed jointly.

From the analysis of the mean results per corruption type, two properties were discerned. The first is that the corruption types in the noise and blur category generally achieve higher similarity results than the corruption types of the weather and digital category. Even the best performing corruption types of the noise and blur category - namely, *speckle_noise* and *gaussian_blur*, respectively - achieve higher similarity results than the worst performing corruption types in the weather and digital category, namely *frost* and *elastic_transform*, respectively. In conclusion, the model is likely more robust against weather and digital corruptions than against noise and blur corruptions. Second, there is a large spread among mean corruption type results of the same corruption category. Especially for the three categories: weather, digital and blur, the deviation between highest and lowest mean corruption type result is significant. For the noise category the deviation among the mean results is more stable. The relatively big deviations across the mean corruption type results contradicts the expectations on a robust model behaviour. For a robust model, the measured values on the embeddings would be expected to be quite similar for all types of corruption.

When considering the average results per severity level, these show a significant dispersion around the overall mean result per corruption type. As could be expected the lowest severity level normally achieves the best result of a corruption type and the highest severity level the worst result. There are subtle differences observable for the dispersion among the five severity level results per corruption type. First, the dispersion of the severity levels for weather corruption types seems to be less severe than for corruption types of the other three categories. Furthermore, in some cases specific severity levels deviate significantly from the mean result of the corruption type. For example, it is clearly observable that the result of the fifth

severity level for the *elastic_transform* corruption does not align with the mean result for this corruption relative to the other corruption types. This concludes that generally there is no linear relationship between corruption severity and impact on the embedding space. Furthermore, the relationship between corruption severity and impact on the embedding space depends on the corruption type.

Class-A' measure results relative to classification results When considering the order of the classification results for each classifier (classifier-CLS and classifier-CLS-PA), it is notable that they are identical. For almost all datasets the accuracy values of classifier-CLS-PA are up to 7 percentage points higher than the respective accuracies achieved by classifier-CLS on the same dataset. Only for ImageNet-R, classifier-CLS achieved an accuracy that is ca. 2 percentage points higher, compared to the accuracy of classifier-CLS-PA. The higher accuracy of classifier-CLS for ImageNet-R may be explained by the hypothesis that the information stored in the CLS token is less affected by the distribution shift present in the ImageNet-R data than the information stored in the patch tokens.

From the table, it is evident that the C2 measure evaluates the impact of image corruptions on the embedding space less strongly than the other measures. This observation was already made when comparing the results of class-A' with the results of class-B. By hand of the table, this becomes clear by comparing the rank of ImageNet-R with the ranks of the ImageNet-C results for all measures and all classifiers. In the column containing the C2 measure values, ImageNet-R is ranked last, whereas the classifiers and the other measures place ImageNet-C-Sev-5 in last position. Consequently, the C2 measure assigns the embeddings of the Imagenet-C-Sev-5 samples a higher similarity to the embeddings of class-A samples than the embeddings of ImageNet-R samples.

The L1 and C1 measures maybe also underestimate the shift induced by image corruptions on the embedding space. This assumption is supported by the fact that ImageNet-C-Sev-1 achieves better results than ImageNet-V2-TOP for the L1 and C1 measures, wheras ImageNet-V2-TOP achieves better classification results than ImageNet-C-Sev-1.

Furthermore, not all measures show the same order of ImageNet-V2 results as the classifiers. Only for the C2 and C3 measure the ImageNet-V2 results have the same order as in the classification results. The L1 and C1 measures evaluate ImageNet-V2-70 better than ImageNet-V2-TOP although Imagenet-V2-TOP achieves higher accuracies than ImageNet-V2-70. This indicates that the C2 and C3 measures are better suited for evaluating the impact of smaller covariate shifts in the data such as present in ImageNet-V2.

7 Conclusion

This thesis aimed to measure the influence of covariate shifts on the embedding space of the DINOv2 ViT. More specifically, the approach was to quantify the change in the embeddings of covariate-shifted samples relative to the embeddings of ID classes with different semantic content.

In practice, a general scheme was designed which enabled systematic class-level comparisons of embeddings across several datasets. On focus of the comparisons was the evaluation of the similarity between the embeddings of covariate-shifted samples and the embeddings of the corresponding ID samples. For evaluating the similarity of two embeddings, four different measures were used, three of which were proposed in this work. For robustness evaluation, the results achieved by covariate-shifted samples were set in relation to results of ID samples.

The comparison scheme was applied on ImageNet-1k, which was used as the ID dataset, and on ImageNet-R, ImageNet-V2, and ImageNet-C, which were used as the OOD datasets. In total 200 comparisons were conducted between the embeddings of one ID class and the embeddings of 99 corresponding OOD classes. In each comparison, the 99 OOD classes differed either in the severity or the type of the covariate shift present in the samples.

In summary, the following are considered as the most important findings from the conducted experiments. Across all measures, an overall pattern in the results was observed. As part of this pattern, embeddings corresponding to ImageNet-V2 samples showed a significantly higher similarity to their corresponding ID embeddings than those of ImageNet-R. For the ImageNet-C benchmark, a consistent decrease in similarity between the embeddings of corrupted samples and the embeddings of the corresponding ID samples was observed with increasing corruption severity. Furthermore, the results of the ImageNet-C dataset revealed that the effect on the embedding space varies according to the type of image corruption. The methodological approach used to relate embeddings of covariate-shifted samples to the embeddings of certain ID classes, that represent not the same topic as the covarate-shifted sample, was not successful across all measures. For two of the measures, the similarity scores of the reference ID embeddings were roughly between those of samples with smaller covariate shifts and those with larger covariate shifts. In contrast, in the results of the other two measures this behaviour was not observed. Overall, each of the four measures showed a different evaluation behaviour for each type of covariate shift corresponding to its associated OOD dataset. A potential explanation for this behaviour is that different types of covariate shifts induce different changes in the embedding space. Depending on which properties of the embedding space are more reflected by a given measure, the impact of certain covariate shift types may be evaluated as more or less severe.

This works contributed by providing potential approaches for interpreting DINOv2's OOD robustness through the model's embedding space. However, the observations reported in this thesis cannot be generalized to all DINOv2 models. In subsequent studies, experiments on larger DINOv2 models, like ViT-g/14, are considered

important for obtaining a more comprehensive understanding of DINOv2's OOD robustness behaviour.

Bibliography

facebookresearch/dinov2: PyTorch code and models for the DINOv2 self-supervised learning method., a. URL `https://github.com/facebookresearch/dinov2/tree/main`.

dinov2 model card, b. URL `https://github.com/facebookresearch/dinov2/blob/main/MODEL_CARD.md`.

4.1. Softmax Regression. URL `https://d2l.ai/chapter_linear-classification/softmax-regression.html`.

ImageNet Large Scale Visual Recognition Challenge. URL `https://www.image-net.org/challenges/LSVRC/`.

pickle — Python object serialization. URL `https://docs.python.org/3/library/pickle.html`.

Unsupervised Feature Learning and Deep Learning Tutorial. URL `http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression/`.

facebook/dinov2-giant, August 2025a. URL `https://huggingface.co/facebook/dinov2-giant`.

timm/vit_small_patch14_dinov2.lvd142m, September 2025b. URL `https://huggingface.co/timm/vit_small_patch14_dinov2.lvd142m`.

Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space. In Gerhard Goos, Juris Hartmanis, Jan Van Leeuwen, Jan Van Den Bussche, and Victor Vianu, editors, *Database Theory — ICDT 2001*, volume 1973, pages 420–434. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. ISBN 978-3-540-41456-8 978-3-540-44503-6. doi: 10.1007/3-540-44503-X_27. URL `http://link.springer.com/10.1007/3-540-44503-X_27`. Series Title: Lecture Notes in Computer Science.

Emma Beede, Elizabeth Baylor, Fred Hersch, Anna Iurchenko, Lauren Wilcox, Paisan Ruamviboonsuk, and Laura M. Vardoulakis. A Human-Centered Evaluation of a Deep Learning System Deployed in Clinics for the Detection of Diabetic Retinopathy. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–12, New York, NY, USA, April 2020. Association for Computing Machinery. ISBN 978-1-4503-6708-0. doi: 10.1145/3313831.3376718. URL `https://dl.acm.org/doi/10.1145/3313831.3376718`.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John

Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the Opportunities and Risks of Foundation Models, July 2022. URL `http://arxiv.org/abs/2108.07258`. arXiv:2108.07258 [cs].

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. pages 9650–9660, 2021. URL `https://openaccess.thecvf.com/content/ICCV2021/html/Caron_Emerging_Properties_in_Self-Supervised_Vision_Transformers_ICCV_2021_paper`.

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, June 2009. doi: 10.1109/CVPR.2009.5206848. URL `https://ieeexplore.ieee.org/document/5206848`. ISSN: 1063-6919.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423/`.

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, June 2021. URL `http://arxiv.org/abs/2010.11929`. arXiv:2010.11929 [cs].

Timo Freiesleben and Thomas Grote. Beyond generalization: a theory of robustness in machine learning. *Synthese*, 202(4):109, September 2023. ISSN 1573-0964. doi: 10.1007/s11229-023-04334-9. URL https://doi.org/10.1007/s11229-023-04334-9.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Ishaan Gulrajani and David Lopez-Paz. In Search of Lost Domain Generalization, July 2020. URL http://arxiv.org/abs/2007.01434. arXiv:2007.01434 [cs].

Lihi Gur-Arie. Harness DINOv2 Embeddings for Accurate Image Classification, July 2025. URL https://pub.towardsai.net/harness-dinov2-embeddings-for-accurate-image-classification-f102dfd35c51.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. pages 1026–1034, 2015. URL https://openaccess.thecvf.com/content_iccv_2015/html/He_Delving_Deep_into_ICCV_2015_paper.html.

Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. September 2018. URL https://openreview.net/forum?id=HJz6tiCqYm.

Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations, March 2019. URL http://arxiv.org/abs/1903.12261. arXiv:1903.12261 [cs].

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network, March 2015. URL http://arxiv.org/abs/1503.02531. arXiv:1503.02531 [stat].

Terran Lane and Carla E. Brodley. Approaches to Online Learning and Concept Drift for User Identification in Computer Security. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD'98, pages 259–263, New York, NY, August 1998. AAAI Press.

Yifei Ming, Hang Yin, and Yixuan Li. On the Impact of Spurious Correlation for Out-of-Distribution Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(9):10051–10059, June 2022. ISSN 2374-3468. doi: 10.1609/aaai.v36i9.21244. URL https://ojs.aaai.org/index.php/AAAI/article/view/21244.

Jose G. Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, January 2012. ISSN 0031-3203. doi: 10.1016/j.patcog.2011.06.019. URL https://www.sciencedirect.com/science/article/pii/S0031320311002901.

Shiho Noda, Atsuyuki Miyai, Qing Yu, Go Irie, and Kiyoharu Aizawa. A Benchmark and Evaluation for Real-World Out-of-Distribution Detection Using Vision-Language Models. In *2025 IEEE International Conference on Image Processing (ICIP)*, pages 181–186, September 2025. doi: 10.1109/ICIP55913.2025.11084642. URL https://ieeexplore.ieee.org/document/11084642. ISSN: 2381-8549.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, February 2024. URL http://arxiv.org/abs/2304.07193. arXiv:2304.07193 [cs].

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, editors. *Dataset Shift in Machine Learning*. The MIT Press, December 2008. ISBN 978-0-262-25510-3. doi: 10.7551/mitpress/9780262170055.001.0001. URL https://doi.org/10.7551/mitpress/9780262170055.001.0001.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, July 2021. URL https://proceedings.mlr.press/v139/radford21a.html. ISSN: 2640-3498.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet Classifiers Generalize to ImageNet?, June 2019. URL http://arxiv.org/abs/1902.10811. arXiv:1902.10811 [cs].

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. ImageNet-21K Pretraining for the Masses, August 2021. URL http://arxiv.org/abs/2104.10972. arXiv:2104.10972 [cs].

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, January 2015. URL http://arxiv.org/abs/1409.0575. arXiv:1409.0575 [cs].

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, August 2025. URL http://arxiv.org/abs/2508.10104. arXiv:2508.10104 [cs].

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 18583–18599, Red Hook, NY, USA, December 2020. Curran Associates Inc. ISBN 978-1-71382-954-6.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, August 2023. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762 [cs].

Frederik vom Lehn. Understanding Vision Transformers (ViT), October 2025. URL https://medium.com/advanced-deep-learning/understanding-vision-transformers-vit-70ca8d817ff3.

Ross Wightman. PyTorch Image Models, 2019. URL https://github.com/huggingface/pytorch-image-models. Publication Title: GitHub repository.

Jingkang Yang, Kaiyang Zhou, and Ziwei Liu. Full-Spectrum Out-of-Distribution Detection. *International Journal of Computer Vision*, 131(10):2607–2622, October 2023. ISSN 1573-1405. doi: 10.1007/s11263-023-01811-z. URL https://doi.org/10.1007/s11263-023-01811-z.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized Out-of-Distribution Detection: A Survey. *International Journal of Computer Vision*, 132(12):5635–5662, December 2024. ISSN 1573-1405. doi: 10.1007/s11263-024-02117-4. URL https://doi.org/10.1007/s11263-024-02117-4.

Wenqian Ye, Luyang Jiang, Eric Xie, Guangtao Zheng, Yunsheng Ma, Xu Cao, Dongliang Guo, Daiqing Qi, Zeyu He, Yijun Tian, Megan Coffee, Zhe Zeng, Sheng Li, Ting-hao Huang, Ziran Wang, James M. Rehg, Henry Kautz, and Aidong Zhang. The Clever Hans Mirage: A Comprehensive Survey on Spurious Correlations in Machine Learning, October 2025. URL http://arxiv.org/abs/2402.12715. arXiv:2402.12715 [cs].

Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *British Machine Vision Conference 2016*, York, France, January 2016. British Machine Vision Association. doi: 10.48550/arXiv.1605.07146. URL https://enpc.hal.science/hal-01832503.

Xu-Yao Zhang, Cheng-Lin Liu, and Ching Y. Suen. Towards Robust Pattern Recognition: A Review. *Proceedings of the IEEE*, 108(6):894–922, June 2020. ISSN 1558-2256. doi: 10.1109/JPROC.2020.2989782. URL https://ieeexplore.ieee.org/document/9103349.

Declaration of Authorship

Ich erkläre hiermit gemäß §9 Abs. 12 APO, dass ich die vorstehende Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Des Weiteren erkläre ich, dass die digitale Fassung der gedruckten Ausfertigung der Abschlussarbeit ausnahmslos in Inhalt und Wortlaut entspricht und zur Kenntnis genommen wurde, dass diese digitale Fassung einer durch Software unterstützten, anonymisierten Prüfung auf Plagiate unterzogen werden kann.

Place, Date Stegaurach, 16.01.2026 Signature Frühling A.