Neural Relation Graph for Identifying Problematic Data

Jang-Hyun Kim [1] **Sangdoo Yun** [2] **Hyun Oh Song** [1]

Abstract

Diagnosing and cleaning datasets are crucial for building robust machine learning systems. However, identifying problems within large-scale datasets with real-world distributions is difficult due to the presence of complex issues, such as label errors or under-representation of certain types. In this paper, we propose a novel approach for identifying problematic data by utilizing a largely ignored source of information: a relational structure of data in the feature-embedded space. We develop an efficient algorithm for detecting label errors and outlier data points based on the relational graph structure of the dataset. We further introduce a visualization tool for contextualizing data points, which can serve as an effective tool for interactively diagnosing datasets. We evaluate label error and out-of-distribution detection performances on large-scale image and language domain tasks, including ImageNet and GLUE benchmarks, and demonstrate the effectiveness of our approach for debugging datasets and building robust machine learning systems.

1. Introduction

Identifying problems within a dataset is crucial for improving the quality of the machine learning system and analyzing the model's predictions. For instance, identifying mislabeled or uninformative data help construct concise and effective training datasets (Northcutt et al., 2021a), while identifying whether test data is out-of-distribution (OOD) or corrupted allows for more accurate model evaluation and analysis (Vasudevan et al., 2022).

In recent years, efforts have been made to identify problematic data by utilizing unary scores on individual data from trained models, such as estimating data influence, monitoring prediction variability throughout training, and calculating prediction error margins (Koh & Liang, 2017; Toneva et al., 2019; Northcutt et al., 2021b). However, identifying such data can be challenging, particularly when dealing with large-scale datasets from real-world distributions. In real-world settings, datasets may have multiple problems, including label issues or under-representation of certain types, each of which can lead to the model error and prediction sensitivity (Koh et al., 2021). For example, Figure 2 shows that a neural network exhibits low negative prediction margin scores for a sample with label error and outlier data, indicating that previous unary scoring methods may have limitations in identifying the sources of problems.

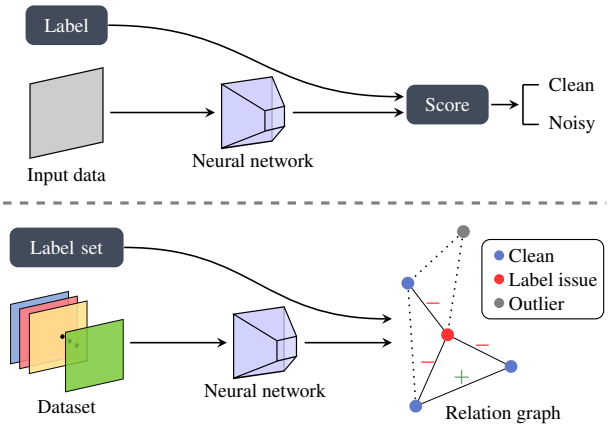


Figure 1. The conceptual illustration of the conventional approaches (top) and our proposed approach (bottom). While the previous approaches measure the prediction error or sensitivity on each data point, our method identifies problematic data by leveraging the relational structure of data. In the relation graph, positive edges signify complementary relation, negative edges denote conflicting relation, and dashed lines indicate negligible relation between data.

In this work, we propose a novel approach for identifying problematic data by leveraging the feature-embedded structure of a dataset that provides richer information than individual data alone (Song et al., 2016; Park et al., 2019). We measure the relationship among data in the feature embedding space while comparing the assigned labels independently. By comparing input data and labels separately, we are able to isolate the factors contributing to model errors, resulting in improved detection of problematic data. Based on this relational information, we construct a graph structure on the dataset and identify whether the data itself or the

[1]Department of Computer Science and Engineering, Seoul National University [2]NAVER AI Lab. Correspondence to: Hyun Oh Song <hyunoh@snu.ac.kr>.

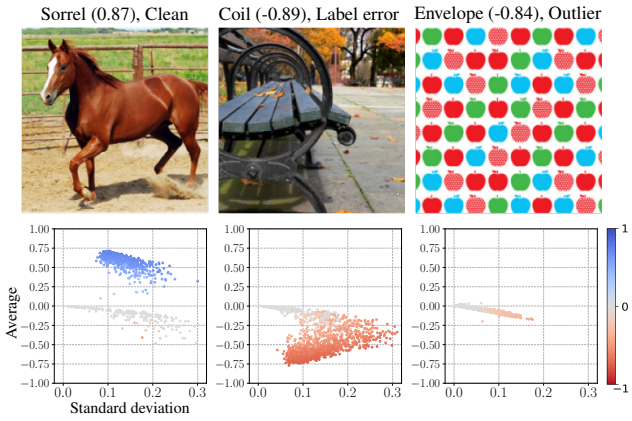


Figure 2. Data samples and their labels from ImageNet (top row) and the corresponding relation maps by an MAE-Large model (bottom row). We report the prediction margin score ($\in [-1, 1]$) in the parenthesis next to the label. The color represents the relation value at the last converged checkpoint. We select representative samples for clean data, data with a label error, and outlier data.

label is problematic (Figure 1). To this end, we develop a novel graph algorithm that efficiently identifies label errors and OOD data points.

Inspired by dataset cartography (Swayamdipta et al., 2020), we further introduce a tool, named data relation map, for contextualizing data points within the feature-embedded structure (Section 3.4). This tool visualizes the relational structure of a data sample and can serve as an interactive tool for diagnosing datasets. Specifically, a data relation map measures the variance and mean value of relations between a data pair throughout training. From Figure 2, we observe that the second and the third samples exhibit different relation map patterns, although they have similar margin scores. This highlights that the relational structure provides new information not captured by the unary scoring methods.

Our approach only requires the model's feature embedding and prediction score on data, making it more scalable compared to methods that require calculating the network gradient on each data point or retraining models multiple times to estimate data influence (Pruthi et al., 2020; Ilyas et al., 2022). Furthermore, our method is domain- and model-agnostic, and thus is applicable to various tasks. We evaluate our approach on label error and OOD detection tasks with large-scale image and language datasets, including ImageNet and GLUE benchmarks (Russakovsky et al., 2015; Wang et al., 2019). Our experiments show state-of-the-art performance on both tasks, demonstrating its effectiveness for debugging datasets and developing robust machine learning systems.

2. Related Works

Label error detection Label errors in datasets can negatively impact model generalization and destabilize machine learning evaluations (Hu et al., 2021; Northcutt et al., 2021b). Previous works attempt to address this issue by detecting and correcting label errors using trained models, such as bagging and bootstrapping (Sluban et al., 2014; Reed et al., 2014). Subsequent approaches propose training neural networks to learn clean data sampling schemes or data Shapley values (Jiang et al., 2018; Ghorbani & Zou, 2019). To mitigate overfitting on label errors, some works suggest tracking the training process to measure prediction variability and the area under the margin curve (Toneva et al., 2019; Pleiss et al., 2020). Recently, Northcutt et al. (2021a) verify the effectiveness of the prediction margin scores for the iterative dataset cleaning. Chong et al. (2022) observe that simple fine-tuned losses by large pre-trained models are competitive with more complex previous approaches. In this work, we leverage the previously under-explored relational structure of data, resulting in improved label error detection.

OOD detection Detecting OOD data is important for building robust machine learning systems in real-world environments (Koh et al., 2021). However, the over-confidence of neural networks makes it difficult to identify OOD data (Nguyen et al., 2015). To address this issue, previous works propose scoring methods such as the Maximum Softmax Probability, Energy score, and Max Logit score (Hendrycks & Gimpel, 2017; Liu et al., 2020; Hendrycks et al., 2022). Other approaches suggest adding perturbations to the input or rectifying the activation to identify OOD data (Liang et al., 2018; Sun et al., 2021). Lee et al. (2018) propose fitting a Gaussian probabilistic model to estimate the data distribution. Recently, Sun et al. (2022) propose a non-parametric approach measuring the k-nearest feature distance. In this work, we explore the use of the relational structure on the feature-embedded space for OOD data identification. Our approach is applicable to a wide range of domains without additional training while outperforming existing scoring methods on large-scale OOD detection benchmarks.

Data influence Another line of research for identifying problematic data involves measuring the influence of a training data point on model predictions. Koh & Liang (2017) originally propose the second-order approximation of the loss function to measure the influence. Subsequent works attempt to reduce the computational cost by applying the representer theorem, tracing network gradients on data, or speeding up the inverse hessian calculation (Yeh et al., 2018; Pruthi et al., 2020; Schioppa et al., 2022). Other works attempt to fit a data model that estimates model predictions given a subset of data (Ilyas et al., 2022). These influence-based approaches identify problematic data by measuring self-influence, *i.e.*, the influence of a training data point on its own loss (Koh & Liang, 2017; Pruthi et al., 2020). While these methods offer new perspectives, they are computationally expensive due to the need to calculate network gradients

on each data point or train multiple neural networks, making them difficult to apply to large-scale settings. In addition, some recent works observe that these methods can be sensitive to outliers and fragile to training schemes (Barshan et al., 2020; Basu et al., 2021). Our data relation graph can be a scalable alternative for estimating the relationship between training data and identifying problematic data.

3. Methods

The learned neural networks' feature space has shown to capture meaningful semantics of data (Ramesh et al., 2022). In this section, we describe our approach leveraging the neural feature-embedded structure of a dataset for identifying problematic data. To this end, we first define data relation to construct a data relational graph on the feature space. Using this graph structure, we propose novel graph algorithms for the detection of label errors and OOD data points. We further introduce a data relation map as an effective tool for diagnosing and contextualizing data.

3.1. Data Relation

We describe our approach in the context of a classification task, while also noting that the ideas are generalizable to other types of tasks as well. Let us assume we have a trained neural network on a training dataset $\mathcal{T} = \{(x_i, y_j) \mid i = 1, \dots, n\}$. For $x_i \in \mathcal{X}$, we extract the feature representation $\mathbf{f}_i$ and the prediction probability vector $\mathbf{p}_i$ from the model.

We propose a class of bounded kernel $k : \mathcal{X} \times \mathcal{X} \to [0, M]$ that measures the semantic similarity between data points on feature space:

$$k(x_i, x_j) = |s(\mathbf{f}_i, \mathbf{f}_j) \cdot c(\mathbf{p}_i, \mathbf{p}_j)|^t. \tag{1}$$

Here, t is a positive scalar value that controls the shape of the kernel value distribution. A larger value of t makes a small kernel value smaller, which is effective in handling small noisy kernel values. The scalar value $s(\mathbf{f}_i, \mathbf{f}_j)$ denotes a similarity measurement between features. We adopt the truncated cosine-similarity that has been widely used in representation learning (Schroff et al., 2015; Reimers & Gurevych, 2019). We use the hinge function at zero, resulting in the following positive feature-similarity function:

$$s(\mathbf{f}_i, \mathbf{f}_j) = \max(0, \cos(\mathbf{f}_i, \mathbf{f}_j)).$$

While the feature similarity captures the meaningful semantic relationship between data points, we observe that considering the prediction scores $\mathbf{p}_i$ can further improve the quality of data identification. To incorporate prediction scores into our approach, we introduce a term $c(\mathbf{p}_i, \mathbf{p}_j)$ that measures the compatibility between the predictions on data points. Any positive and bounded compatibility function is suitable for the kernel class defined in Equation (1). In section 4.3, we examine the effects of different design choices for the compatibility term through empirical studies.

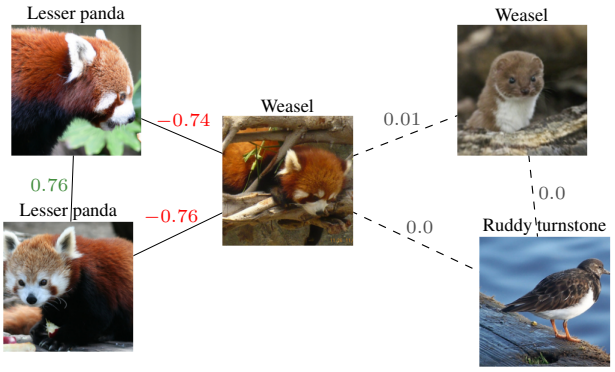


Figure 3. Illustration of a relation graph with samples from ImageNet and the MAE-Large model. We denote the assigned label above each sample. Here, the center image has a label error.

In our main experiments, we use the predicted probability of belonging to the same class as the compatibility term $c(\mathbf{p}_i, \mathbf{p}_j)$. Specifically, given the predicted label random variables $\hat{y}_i$ and $\hat{y}_j$, the proposed compatibility term is

$$c(\mathbf{p}_i, \mathbf{p}_j) = P(\hat{y}_i = \hat{y}_j) = \mathbf{p}_i^\intercal \mathbf{p}_j. \tag{2}$$

From a different perspective, we interpret this compatibility term as a measure of confidence for feature similarity.

By incorporating the assigned label information with the similarity kernel in Equation (1), we define the relation function $r : \mathcal{X} \times \mathcal{Y} \times \mathcal{X} \times \mathcal{Y} \to [-M, M]$:

$$r((x_i, y_i), (x_j, y_j)) = 1(y_i = y_j) \cdot k(x_i, x_j), \tag{3}$$

where $1(y_i = y_j) \in \{-1, 1\}$ is a signed indicator value. The relation function reflects the degree to which data samples are complementary or conflicting with each other. From Figure 3, the center image with a label error has a negative relation to the left samples that belong to the same ground-truth class. In contrast, the two left samples with correct labels have a positive relation. We also observe that samples with dissimilar semantics exhibit near-zero relation values.

Our relation function in Equation (3) only requires forward computation of neural networks which is embarrassingly parallelizable and scalable to large-scale settings. In the following sections, we describe our efficient graph algorithms for detecting label errors and OOD data.

3.2. Label Error Detection

We consider a fully-connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where the set of nodes $\mathcal{V}$ corresponds to $\mathcal{T}$ and the weights $\mathcal{W}$ on edges $\mathcal{E}$ are the relation values defined in Equation (3). For clarity, we denote a data point by an index,

i.e., $\mathcal{T} = \{1, \dots, n\}$ and $r((x_i, y_i), (x_j, y_j)) = r(i, j)$. Additionally, we set $r(i, i) = 0$ for $i \in \mathcal{T}$. In this section, we present a method utilizing the relation graph for measuring the label noisiness score on each data point as in prior studies (Northcutt et al., 2021a).

As illustrated in Figure 3, the data with label errors exhibit negative relations to other samples, implying that the data samples are similar in the feature-embedding space, yet have dissimilarly assigned labels. By aggregating the edge information of each node, we can measure the label noisiness score for each data sample. However, simply summing all edge weights of a node can lead to suboptimal results as a negative edge weight can also contribute to the score for clean data as shown in Figure 3. To rectify this, we identify a subset of data likely to have correct labels and calculate scores based on them.

Specifically, we partition the nodes into two groups such that the sum of edges between the groups has the lowest negative value, meaning that the label information of the two groups is the most different. Since the label error ratio is typically lower than the half of the training set, we estimate the smaller set to be the noisy set $\mathcal{N}$ and formulate this as the following min-cut problem:

$$\mathcal{N}^* = \underset{\mathcal{N} \subset \mathcal{T}}{\operatorname{argmin}} \ \text{cut}(\mathcal{N}, \mathcal{T} \setminus \mathcal{N}) \ \Big(:= \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{T} \setminus \mathcal{N}} r(i, j) \Big), \quad (4)$$
$$\text{subject to } |\mathcal{N}| < n/2.$$

The min-cut problem with the signed edges is NP-complete (Hartmanis, 1982). To solve this problem, we adopt the Kerninghan-Lin algorithm, which finds a local minimum solution through greedy optimization (Kernighan & Lin, 1970). However, the original algorithm that swaps data one by one is not suitable for large-scale settings. To this end, we propose an efficient batch-level algorithm in Algorithm 1.

Algorithm 1 iteratively updates the noisy set $\mathcal{N}$ and label noisiness score vector $s \in \mathbb{R}^n$, where a lower score indicates a higher likelihood of label errors. Given the current estimation of $\mathcal{N}$, the cut value excluding edges of node $i \in \mathcal{T}$ is $\text{cut}(\mathcal{N} \setminus \{i\}, \mathcal{T} \setminus \mathcal{N} \setminus \{i\})$. We measure the label noisiness score of node i by comparing the objective values when including i in $\mathcal{N}$ and when including i in $\mathcal{T} \setminus \mathcal{N}$:

$$\begin{aligned} s[i] &= \text{cut}(\mathcal{N} \cup \{i\}, \mathcal{T} \setminus \mathcal{N} \setminus \{i\}) - \text{cut}(\mathcal{N} \setminus \{i\}, \mathcal{T} \setminus \mathcal{N} \cup \{i\}) \\ &= \sum_{j \in \mathcal{T} \setminus \mathcal{N}} r(i, j) - \sum_{j \in \mathcal{N}} r(i, j). \end{aligned}$$

Here we use the assumption $r(i, i) = 0$. In practice, the number of elements in $\mathcal{N}$ is small, so we can efficiently update the score vector s as in Algorithm 1. After calculating the score vector s, we update the noisy set $\mathcal{N}$ by selecting nodes with score values below the threshold ϵ. We can obtain the solution to Equation (4) with $\epsilon = 0$, while negative values of ϵ result in smaller $\mathcal{N}$ consisting of data samples that are more likely to have label noise.

Algorithm 1 Noisy label detection

Input: Relation function r, threshold value ϵ
Notation: The number of data n
for $i = 1$ **to** n **do**
  # initialize label noisiness scores
  $\bar{s}[i] = \sum_{j=1}^{n} r(i, j)$
end for
$s = \bar{s}$
repeat
  # estimate noisy data subset
  $\mathcal{N} = \{i \mid s[i] < \epsilon, i \in [1, \dots, n]\}$
  for $i = 1$ **to** n **do**
    $s[i] \leftarrow \bar{s}[i] - 2 \sum_{j \in \mathcal{N}} r(i, j)$
  end for
until convergence
Output: $s, \mathcal{N}$

Algorithm 1 satisfies the following convergence property. In Appendix A, we provide proof and present an empirical convergence analysis on large-scale datasets.

Proposition 1. *Algorithm 1 with a single node update at each iteration converges to local minima.*

3.3. OOD Detection

The utilization of the relational structure of data enables the isolation of problems arising from label issues or outlier data, resulting in more accurate identification of label errors and outlier data compared to existing methods. To verify this, we further propose a scoring method for OOD detection based on the feature embedding structure. Specifically, we measure the distance of a data point from the data distribution using the similarity kernel in Equation (1). This is possible by aggregating the similarity kernel values of a data point. For $\mathcal{S} \subseteq \mathcal{T}$ and data x, we measure the OOD score as

$$\text{ood}(x) = \sum_{i \in \mathcal{S}} k(x, x_i).$$

Lower values in the OOD score indicate that the data are more distributionally outliers. By using a subset of the data $\mathcal{S}$, we adjust the computational cost and memory requirements for the OOD score to suit the inference environment. We experimentally demonstrate that our approach maintains the OOD detection performance even when using a 1% random subset of the data on ImageNet (Section 4.2). In comparison to previous approaches such as the k-nearest neighbor or Mahalanobis distance (Sun et al., 2022; Lee et al., 2018), our approach is less sensitive to hyperparameters and requires no additional training (Section 4.3).

3.4. Data Relation Map

Swayamdipta et al. (2020) proposed a method called dataset cartography, which aids the analysis of a dataset by projecting them onto a 2D plot. Specifically, they draw a scatter plot of the mean and standard deviation of the model's prediction probabilities for each data sample during training. Analogous to the dataset cartography, we propose a novel data contextualization tool called data relation map, that visualizes the relationship between data along the training process. To this end, we uniformly store checkpoints during training. We denote a set of these checkpoints as $\mathcal{K}$, where r_k refers to the relation function for checkpoint $k \in \mathcal{K}$. For each data sample i, we draw a scatter plot of the mean and standard deviation of relation values $\{r_k(i,j) \mid k \in \mathcal{K}\}$ for $j \in \mathcal{T} \setminus \{i\}$.

In Figure 2, we provide relation maps of three samples from ImageNet, using 10 checkpoints of MAE-Large (He et al., 2022). The three samples each represent clean data, data with a label error, and outlier data. From the figure, samples show different relation map patterns. Specifically, the relation map of a clean data sample exhibits a majority of positive relations with relatively small variability. We note that there are gray-colored relations in high variability regions (0.2<std), indicating that the model resolves conflicting relations at convergence. On the other hand, the relation map of the sample with a label error demonstrates a majority of negative relations. Notably, high variance relations result in largely negative relations at convergence, suggesting that conflicts intensify. Lastly, the relation map of the outlier data sample reveals that relations are close to 0 during training.

The data relation map can serve as a model-based fingerprint of the data, and our algorithm exploits these patterns to effectively identify problematic data. We provide additional data relational maps for various models in Appendix D. In the following experimental section, we quantitatively evaluate our method, providing further evidence for its effectiveness.

4. Experimental Results

In this section, we experimentally verify the effectiveness of our approach in detecting label errors (Section 4.1) and OOD data (Section 4.2). We further conduct ablation study of our algorithm in Section 4.3. We provide implementation details including hyperparameter settings in Appendix B.1.

4.1. Label Error Detection

4.1.1. Setting

Datasets We conduct label error detection experiments on large-scale image and language datasets: ImageNet, SST2, and MNLI (Russakovsky et al., 2015; Wang et al., 2019).

Table 1. Validation top-1 accuracy of MAE-Large trained on ImageNet with noisy labels.

Label Noise Ratio	0.	0.04	0.08	0.12	0.15
Top-1 Accuracy	85.89	84.96	84.15	82.88	81.50

ImageNet consists of about 1.2M image data from 1,000 classes. SST2 is a binary sentiment classification dataset with about 67K movie review sentences. MNLI consists of about 393K sentence pairs with textual entailment annotations. The task is to classify whether the premise sentence entails or contradicts the hypothesis sentence or is neutral.

Following Pruthi et al. (2020), we construct a noisy training set by flipping labels of certain percentages of training data with the top-2 prediction of the trained model on correctly classified data. We use different neural network architectures for constructing a noisy training set and detecting label errors to avoid possible correlation. We leave a more detailed procedure for constructing the noisy training set in Appendix B.2. Table 1 shows the validation accuracy of a model trained on ImageNet with noisy labels, demonstrating the potential benefits of label error detection and cleaning.

Baselines We compare our method, denoted as *Relation*, to three baselines that are suitable for large-scale datasets. We consider fine-tuned loss on data from pre-trained models, denoted as *Loss* (Chong et al., 2022). Cleanlab is another competitive baseline that uses a prediction probability margin score, referred to as *Margin* (Northcutt et al., 2021a). We also consider the influence-based approach called *TracIn* that measures the network gradient norm on each data throughout the training (Pruthi et al., 2020). TracIn proposes to use a temporal ensemble of neural networks. However, note that other methods including ours can also use multiple neural networks by averaging the scores. For a fair comparison, we compare methods using the same number of neural networks. Specifically, we use a single converged neural network in our main experiments, while also providing results with a temporal ensemble in Table 5.

Metric We evaluate the detection performance based on label noisiness scores by each method. In general, the label error ratio is small, resulting in an imbalanced detection problem (Northcutt et al., 2021b). The AUROC measure can be optimistic and misleading for imbalanced classification (Davis & Goadrich, 2006). In this respect, we mainly report the AP (average precision) and TNR95 (TNR at 0.95 TPR) scores. We provide AUROC results in Appendix C.1.

4.1.2. Results and Analysis

ImageNet We measure the label error detection performance on ImageNet with the synthetic label noise by train-

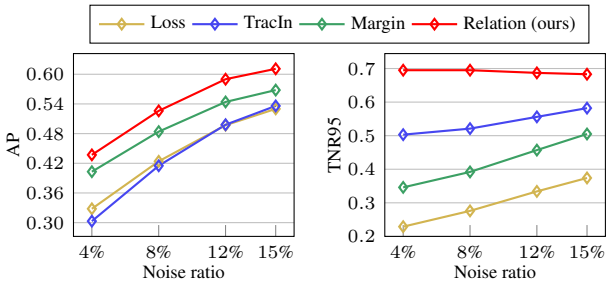


Figure 4. Label error detection performance on a range of label error ratios (MAE-Large, ImageNet).

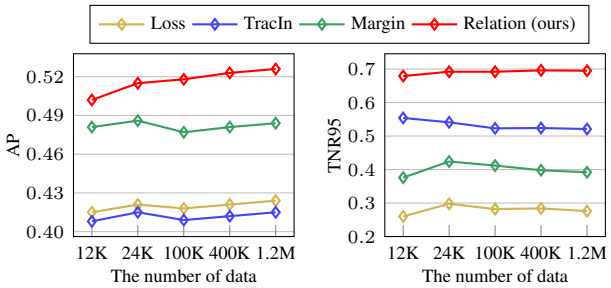


Figure 5. Label error detection performance according to the number of data (MAE-Large, ImageNet, 8% label noise). The x-axis is on a log-scale.

ing an MAE-Large model (He et al., 2022). Note that the model does not have access to information about the changed clean labels during the entire training process. Figure 4 shows the detection performance over a wide range of label noise ratios from 4% to 15%. As shown in the figure, our approach achieves the best AP and TNR95 performance compared to the baselines. Especially, our method maintains a high TNR95 over a wide range of noise ratios, indicating that the number of data that need to be reviewed by human annotators is significantly smaller when cleaning the dataset.

It is worth noting that our method relies on the number of data for constructing a relation graph. To measure the sensitivity of our algorithm to the number of data, we evaluate the detection performance using a reduced number of data. Figure 5 shows the detection performance on 8% label noise with MAE-Large. From the figure, we find that our algorithm maintains the best detection performance even with 1% of the data (12K). This demonstrates that our algorithm is effective even when only a small portion of the training data is available, such as continual learning or federated learning (Parisi et al., 2019; McMahan et al., 2017).

In Table 2, we provide detection performance for different scales of neural networks on 8% label noise. The table shows our approach achieves the best AP with MAE-Base, verifying the robustness of our approach to the network scales. From the table, we note that larger models are more robust to label noise and show better detection performance.

Table 2. Label error detection AP on various architecture scales.

Model	Loss	TracIn	Margin	Relation
MAE-Base	0.412	0.393	0.477	**0.514**
MAE-Large	0.424	0.415	0.484	**0.526**

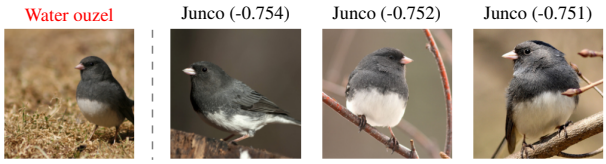
Table 3. Label error detection performance on language datasets with RoBERTa-Base.

Dataset	Metric	Loss	TracIn	Margin	Relation
MNLI	AP	0.764	0.724	0.754	**0.766**
	TNR95	0.497	0.510	0.514	**0.638**
SST2	AP	0.861	0.854	0.861	**0.881**
	TNR95	0.850	0.837	0.850	**0.870**

Natural language domain We apply our method to language domain datasets: MNLI and SST2. We design the label noise settings identical to the previous ImageNet section. Specifically, we train the RoBERTa-Base model under the 10% label noise setting (Liu et al., 2019). Table 3 shows our approach achieves the best AP and TNR95 on the language datasets, demonstrating the generality of our approach across different data types. Note that SST2 is a binary classification task, so Loss and Margin have the same order of scores and therefore the same performance. Figure 6 illustrates detected samples with label errors. We present additional qualitative results in Appendix D.2.

Realistic label noise scenario Beyer et al. (2020) observe that there exist many label errors in the ImageNet validation set. To address this issue, they carry out a label cleaning process with human experts and correct approximately 29% of the validation data labels through multi-labeling. Using this re-labeled validation set, we conduct experiments under the realistic label noise, with the task of detecting the data samples with changed labels. Note that Beyer et al. (2020) utilize neural networks trained on ImageNet to assist human expert annotators in the label-cleaning process. To avoid potential correlation, we conduct experiments using the latest models that Beyer et al. (2020) did not use. Specifically, we measure the detection performance with MAE, BEIT, and ConvNeXt (He et al., 2022; Bao et al., 2022; Liu et al., 2022). Additionally, we consider ConvNeXt pre-trained on ImageNet-22k, denoted as ConvNeXt-22k, to analyze the effect of pre-training on external data.

We evaluate our relation graph in two settings: using only the validation set and considering the training set in addition to the validation set. Table 4 shows that our approach outperforms the baselines when using only the validation set and performs even better when considering the training dataset. The results on ConvNeXt-22k demonstrate that pre-training on external data improves detection performance.



"entertaining and informative documentary"	Negative
"entertaining movie"	Positive (-0.850)
"fascinating and timely content"	Positive (-0.846)

Figure 6. Detected data samples with label errors (marked in red) from ImageNet (top) and SST2 (bottom). We present samples with conflicting relations next to the detected samples and denote the corresponding relation value in parenthesis.

Table 4. Label error detection AP on ImageNet validation set. All the model scales are Large. *Rel. (w/ train)* means the relation graph on the union of validation and training sets.

Model	Loss	TracIn	Margin	Rel.	Rel. (w/ train)
MAE	0.703	0.695	0.708	0.733	**0.735**
BEIT	0.719	0.718	0.718	0.737	**0.740**
ConvNeXt	0.709	0.700	0.713	0.735	**0.737**
ConvNeXt-22k	0.722	0.719	0.724	0.744	**0.746**

Table 5. Label error detection AP by the temporal model ensemble (MAE-Large, ImageNet, 8% label noise). In parenthesis, we denote the performance gain compared to the detection by a single converged model.

Loss	TracIn	Margin	Relation
0.465 (0.041)	0.449 (0.034)	0.544 (0.06)	**0.562** (0.036)

Memorization issue We further examine the findings of Zhang et al. (2021), who observe that large-scale neural networks have a large capacity to memorize label errors, which can negatively impact detection performance. As shown in the left figure of Figure 7, we find that the AP score decreases as the training progresses after 30 epochs, with the MAE-Large model that converges at 50 epochs. To further inspect this phenomenon, we plot the precision-recall curve in the right figure of Figure 7. Interestingly, we observe that precision increases at low recall but decreases at mid-level recall as the training progresses. This suggests that learning has both positive and negative effects on detecting label noise, and we speculate that memorization is one cause.

One approach for utilizing the aforementioned observations is to use the temporal ensemble of models (Laine & Aila, 2017). Following Pruthi et al. (2020), we average the estimated label noisiness scores of uniformly sampled models throughout training. Table 5 shows that this technique improves the performance of all methods, with our approach still exhibiting the best performance. These results confirm the effectiveness of temporal ensembles when more computation and storage are available.

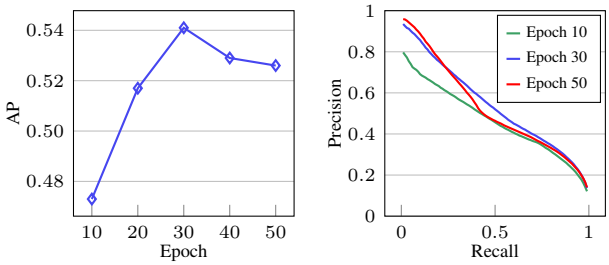


Figure 7. Label error detection performance of relation graph throughout the training (MAE-Large, ImageNet, 8% label noise).

4.2. OOD Detection

4.2.1. Setting

Datasets Following Sun et al. (2022), we evaluate methods on ImageNet using four OOD datasets: *Places*, *SUN*, *iNaturalist*, and *Textures* (Zhou et al., 2017; Xiao et al., 2010; Van Horn et al., 2018; Cimpoi et al., 2014). Each of these OOD datasets consists of 10,000 data samples except for Textures which has 5,640 data samples. We also combine these four datasets, denoted as *ALL*, and measure the overall OOD detection performance on this dataset.

Baselines We consider the following representative OOD scoring approaches: Maximum Softmax Probability (*MSP*), *Max Logit*, and *Energy* scores (Hendrycks & Gimpel, 2017; Hendrycks et al., 2022; Liu et al., 2020). We also consider the *KNN* method that computes feature distance to the k-nearest neighbor (Sun et al., 2022). We tune the hyperparameter k following the guideline provided in the paper.

4.2.2. Results and Analysis

Detection performance Table 6 reports the OOD detection AP and TNR95 on ImageNet with MAE-Large and ResNet-50. The table shows that our approach achieves the best detection performance, demonstrating the effectiveness of the relation graph for OOD detection. According to Sun et al. (2022), the KNN approach exhibits diminishing results on neural networks trained from scratch without pre-training, such as contrastive learning. Table 6 shows the OOD detection performance on ResNet-50 with the KNN method underperforming other baselines, which is consistent with the findings of Sun et al. (2022). However, our approach shows the best performance, demonstrating its robustness to the training technique and model architecture. We report the individual performance on each OOD dataset including the AUROC results in Appendix C.2, where our approach achieves the best score as well.

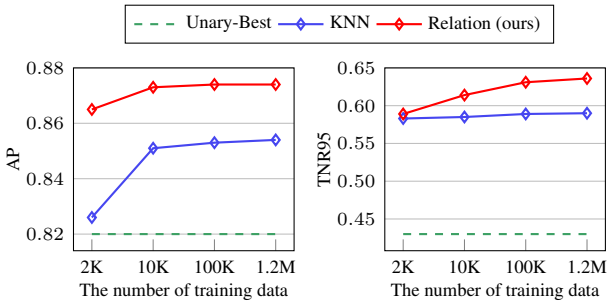


Figure 8. OOD detection performance according to the number of data (MAE-Large, ALL). *Unary-Best* denotes the best performance among the other methods that do not rely on the number of training data. The x-axis is on a log-scale.

Table 6. OOD detection performance on ImageNet (ALL).

Model	Metric	MSP	Max Logit	Energy	KNN	Relation
MAE -Large	AP	0.818	0.808	0.757	0.854	**0.874**
	TNR95	0.428	0.138	0.074	0.590	**0.636**
ResNet-50	AP	0.782	0.767	0.721	0.764	**0.818**
	TNR95	0.496	0.482	0.481	0.380	**0.515**

Effect of the training set size Both our approach and the KNN method rely on the number of training data samples. In Figure 8, we measure the OOD detection performance with a reduced number of training data to examine the effect of the training set size. From the figure, we find that both approaches outperform other baselines while maintaining their performance even with 1% of the training dataset (10K). It is worth noting that KNN requires tuning its hyperparameter according to the size of the training set, whereas our approach uses the identical hyperparameter ($t = 1$) regardless of the training set size.

4.3. Ablation Study

In this section, we conduct an ablation study of our method. We begin by analyzing the sensitivity to the temperature t in Equation (1), and then further study the effect of the probability compatibility term $c(\mathbf{p}_i, \mathbf{p}_j)$. For analysis, we use MAE-Large on ImageNet with 8% label noise ratio.

Temperature t In Equation (1), we introduce a temperature t that controls the shape of the kernel value distribution. A large value of t increases the influence of large similarity values in our algorithm. Figure 9 shows the effect of the temperature value on our detection algorithm's performance. From the figure, we observe that the label error detection performance increases as the t value increases, saturating at $t = 6$. In the case of OOD detection, we achieve the best performance at around $t = 1$. Our algorithm outperforms the best baseline over a wide range of hyperparameters, demonstrating the robustness of our algorithm to the hyperparameter.

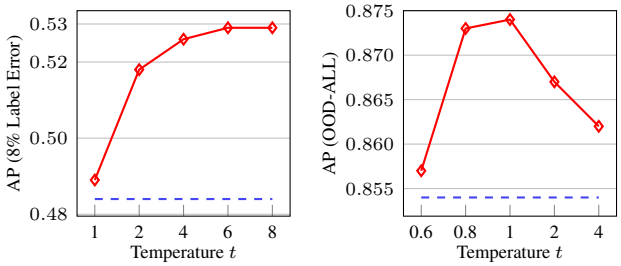


Figure 9. Detection AP Sensitivity over the kernel temperature t. The dashed blue line means the AP of the best baseline.

Table 7. Comparison of the similarity kernel designs.

Task	Metric	Baseline-Best	None	Distance	Dot
Label Error (8%)	AP	0.484	0.471	0.506	0.526
	TNR95	0.521	0.671	0.708	0.695
OOD (ALL)	AP	0.854	0.855	0.857	0.874
	TNR95	0.590	0.630	0.637	0.636

Similarity kernel design In Equation (2), we propose a compatibility term that measures a dot-product between prediction probability vectors, which we denote as *Dot*. In Table 7, we design other types of similarity kernels and evaluate the detection performance. Specifically, we consider a similarity kernel without the compatibility term, denoted as *None*. We also consider a distance-based compatibility term $c(\mathbf{p}_i, \mathbf{p}_j) = \|\mathbf{p}_i - \mathbf{p}_j\|_2$, referred to as *Distance*.

Table 7 shows that our relation graph outperforms the best baseline regardless of the kernel design, especially on the TNR95 metric by a large margin. The results show that the dot-product compatibility term performs comparably to the distance-based compatibility term, slightly outperforming it on the AP metric. By comparing *None* to others, we verify the effectiveness of the compatibility term.

5. Conclusion

In this paper, we exploit the relational structure of data in the feature embedding space for identifying problematic data. To this end, we propose a novel data relation function and develop efficient graph algorithms for detecting label errors and OOD data, which are applicable to large-scale tasks across various domains. Through extensive experiments on ImageNet and GLUE benchmarks, we demonstrate the effectiveness of our approach, achieving state-of-the-art performance in both label error and OOD detection tasks. Additionally, we introduce a data contextualization tool based on our data relation, which can assist in the diagnosis of datasets. The proposed algorithms and tools can facilitate the analysis of large-scale datasets and contribute to the development of robust machine learning systems.

References

Bao, H., Dong, L., and Wei, F. Beit: Bert pre-training of image transformers. In *ICLR*, 2022.

Barshan, E., Brunet, M.-E., and Dziugaite, G. K. Relatif: Identifying explanatory training samples via relative influence. In *AISTATS*, 2020.

Basu, S., Pope, P., and Feizi, S. Influence functions in deep learning are fragile. In *ICLR*, 2021.

Beyer, L., Hénaff, O. J., Kolesnikov, A., Zhai, X., and Oord, A. v. d. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

Chong, D., Hong, J., and Manning, C. D. Detecting label errors using pre-trained language models. *arXiv preprint arXiv:2205.12702*, 2022.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *CVPR*, 2014.

Davis, J. and Goadrich, M. The relationship between precision-recall and roc curves. In *ICML*, 2006.

Ghorbani, A. and Zou, J. Data shapley: Equitable valuation of data for machine learning. In *ICML*, 2019.

Hartmanis, J. Computers and intractability: a guide to the theory of np-completeness (michael r. garey and david s. johnson). *Siam Review*, 24(1), 1982.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

Hendrycks, D., Basart, S., Mazeika, M., Mostajabi, M., Steinhardt, J., and Song, D. Scaling out-of-distribution detection for real-world settings. In *ICML*, 2022.

Hu, W., Li, Z., and Yu, D. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *ICLR*, 2021.

Ilyas, A., Park, S. M., Engstrom, L., Leclerc, G., and Madry, A. Datamodels: Predicting predictions from training data. In *ICML*, 2022.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018.

Kernighan, B. W. and Lin, S. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(2), 1970.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *ICML*, 2017.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. In *NeurIPS*, 2020.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *CVPR*, 2022.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.

Northcutt, C., Jiang, L., and Chuang, I. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70, 2021a.

Northcutt, C. G., Athalye, A., and Mueller, J. Pervasive label errors in test sets destabilize machine learning benchmarks. In *NeurIPS Datasets and Benchmarks Track*, 2021b.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113, 2019.

Park, W., Kim, D., Lu, Y., and Cho, M. Relational knowledge distillation. In *CVPR*, 2019.

Pleiss, G., Zhang, T., Elenberg, E., and Weinberger, K. Q. Identifying mislabeled data using the area under the margin ranking. In *NeurIPS*, 2020.

Pruthi, G., Liu, F., Kale, S., and Sundararajan, M. Estimating training data influence by tracing gradient descent. In *NeurIPS*, 2020.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*, 2019.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3), 2015.

Schioppa, A., Zablotskaia, P., Vilar, D., and Sokolov, A. Scaling up influence functions. In *AAAI*, 2022.

Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

Sluban, B., Gamberger, D., and Lavrač, N. Ensemble-based noise detection: noise ranking and visual performance evaluation. *Data mining and knowledge discovery*, 28(2), 2014.

Song, H. O., Xiang, Y., Jegelka, S., and Savarese, S. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016.

Sun, Y., Guo, C., and Li, Y. React: Out-of-distribution detection with rectified activations. In *NeurIPS*, 2021.

Sun, Y., Ming, Y., Zhu, X., and Li, Y. Out-of-distribution detection with deep nearest neighbors. In *ICML*, 2022.

Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N. A., and Choi, Y. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *EMNLP*, 2020.

Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. In *ICLR*, 2019.

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The inaturalist species classification and detection dataset. In *CVPR*, 2018.

Vasudevan, V., Caine, B., Gontijo-Lopes, R., Fridovich-Keil, S., and Roelofs, R. When does dough become a bagel? analyzing the remaining mistakes on imagenet. In *NeurIPS*, 2022.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2019.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

Yeh, C.-K., Kim, J., Yen, I. E.-H., and Ravikumar, P. K. Representer point selection for explaining deep neural networks. In *NeurIPS*, 2018.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3), 2021.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6), 2017.

A. Algorithm Analysis

A.1. Proof

In this section, we provide proof for Proposition 1. Recall that Algorithm 1 updates an estimated noisy set $\mathcal{N}$ at a batch-level as $\mathcal{N} = \{i \mid s[i] < \epsilon\}$. We can also conduct Algorithm 1 with a single node update by moving one sample at each iteration, referred to as a sample-level version of Algorithm 1. Specifically, let $v \in \mathbb{R}^n$ such that $v[i] = -1$ for $i \in \mathcal{N}$ and $v[i] = 1$ for else. Then we move a sample k to another partition at each iteration, where $k = \text{argmin}_{i \in \mathcal{T}} v[i]s[i]$. The algorithm stops when $0 \leq v[k]s[k]$.

Proposition 1. *Algorithm 1 with a single node update at each iteration converges to local minima.*

Proof. The change in the objective value of Equation (4) by moving data i from $\mathcal{T} \setminus \mathcal{N}$ to $\mathcal{N}$ is

$$\sum_{j \in \mathcal{T} \setminus \mathcal{N}} r(i,j) - \sum_{j \in \mathcal{N}} r(i,j), \tag{5}$$

where the change by moving data i from $\mathcal{N}$ to $\mathcal{T} \setminus \mathcal{N}$ is

$$\sum_{j \in \mathcal{N}} r(i,j) - \sum_{j \in \mathcal{T} \setminus \mathcal{N}} r(i,j).$$

Algorithm 1 updates the score s by Equation (5) as

$$\begin{aligned} s[i] &= \sum_{j \in \mathcal{T}} r(i,j) - 2 \sum_{j \in \mathcal{N}} r(i,j) \\ &= \sum_{j \in \mathcal{T} \setminus \mathcal{N}} r(i,j) - \sum_{j \in \mathcal{N}} r(i,j). \end{aligned}$$

The change in the objective value by moving data i to another partition becomes $s[i]$ for $i \in \mathcal{T} \setminus \mathcal{N}$ and $-s[i]$ for $i \in \mathcal{N}$, which is $v[i]s[i]$. Therefore, moving a sample with a negative value of $v[i]s[i]$ to another partition guarantees a decrease in the objective function value. Because a cut value of a graph is bounded, the algorithm converges to local minima by the monotone convergence theorem. □

A.2. Empirical Convergence Analysis

We conduct an empirical study on the convergence of Algorithm 1. Specifically, we randomly sample 100,000 data from ImageNet and construct a relation graph. We compare the batch-level Algorithm 1 and its sample-level version in Figure 10. The figure indicates that both algorithms converge to local minima, while our batch-level algorithm converges faster. Additionally, we observe that the batch-level algorithm achieves a lower objective value, verifying its effectiveness in large-scale settings.

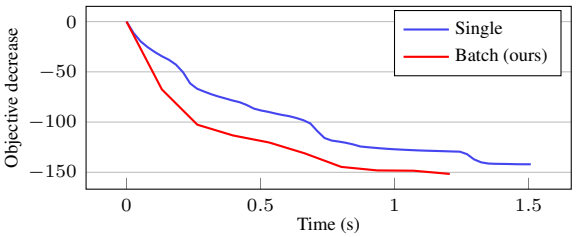


Figure 10. Empirical convergence analysis of min-cut algorithms. *Batch* denotes Algorithm 1 and *Single* denotes a sample-level version of the algorithm.

Table 8. Time spent (s) for label error detection (ImageNet). *Feature* indicates the total computing time for calculating feature embeddings of a dataset, and *Gradient* means the total computing time for calculating network gradient on data. *Algorithm 1* indicates the time spent by our algorithm excluding feature calculation.

Model	Feature	Algorithm 1	Gradient
MAE-Base	2300	400	6000
MAE-Large	6900	420	21000

Table 9. Time spent per sample (ms) for OOD detection (ImageNet). *Unary* refers to methods using logit or probability scores.

Model	Unary	Relation
MAE-Large	12.2	12.3
ResNet-50	8.1	8.2

A.3. Computation Time Comparison

In this section, we measure the time spent on detection algorithms. We use an RTX3090-Ti GPU and conduct experiments on the full ImageNet training set. Table 8 compares computation time for Algorithm 1 and feature calculation. Note that all existing methods based on neural networks, including ours, require the calculation of features. The table shows that Algorithm 1 (excluding feature calculation) requires significantly less computation time than forward computation. We also observe that our algorithm efficiently scales up to larger neural networks which have a larger number of feature embedding dimensions. It is also worth noting that computing gradient takes a much longer time and also requires a large memory budget, demonstrating the efficiency of our algorithm in large-scale label error detection.

In Table 9, we measure the time spent for OOD detection on the full ImageNet training set. Computing our kernel similarity is embarrassingly parallelizable on GPUs. As shown in the table, the overhead time for computing our OOD scores is negligible compared to the time spent for the neural network forward pass on a single data point. Note that we can further reduce the time cost and memory requirements by measuring the OOD score on a subset of the training set as shown in Figure 8.

B. Experiment Settings

B.1. Implementation Details

Models For label error detection, we train models on datasets with label noise. In the case of ImageNet, we fine-tune the pre-trained MAE models following the official training codes[1], which train MAE-Large for 50 epochs and MAE-Base for 100 epochs. It is worth noting that the masked auto-encoding pre-training process of MAE does not utilize label information. In the case of language domain tasks, we fine-tune RoBERTa-Base following the official training codes[2], where we train models for 5 epochs. For models used in OOD detection and label error detection on the validation set (Table 4), we use the trained models provided by the Timm library[3]. For all experiments, we use the inputs of the classification layers as feature embeddings. In the case of RoBERTa, this corresponds to the encoder output of the [CLS] token.

Hyperparameter As shown in Figure 9, we observe that a large value of temperature t benefits label error detection, while a moderate temperature value around 1 shows the best performance on OOD detection. We use $t = 4$ for all experiments regarding label error detection, which shows robust performances over various settings, and use $t = 1$ for all experiments related to OOD detection regardless of neural network architectures. We use $\epsilon = -0.05$ in Algorithm 1 for all experiments after scaling the label noisiness score to have a maximum absolute value of 1. We observe that the conservative estimation of noisy set $\mathcal{N}$ by using small negative ϵ values leads to slightly better results than $\epsilon = 0$ in label error detection (about 1% improvement in TNR95).

Other tricks We find that small noisy kernel values accumulate errors as we consider large numbers of data. To resolve this issue, we clamp small similarity kernel values that fall below an absolute value of 0.03 as zero in Equation (1).

B.2. Synthetic Label Noise

In Section 4.1, we conduct controlled experiments by generating synthetic label noise on the ImageNet and GLUE benchmarks. Specifically, we flip labels of a certain percentage of training data with the top-2 prediction of trained models on correctly classified data. For the label flip, we use MAE-Huge for ImageNet and RoBERTa-Large for language datasets. Note that we did not use these models for detecting label errors to prevent possible correlations. We note that the original ImageNet training set may contain label issues which can lead to misleading experimental results (Northcutt et al., 2021b). Therefore, we remove about 4% of data that are likely to have label issues by following Northcutt et al. (2021a) with MAE-Huge, resulting in a total of 1,242,890 data samples. We conduct synthetic label error experiments on this pre-cleaned training set.

[1] https://github.com/facebookresearch/mae

[2] https://github.com/facebookresearch/fairseq/tree/main/examples/roberta

[3] https://github.com/rwightman/pytorch-image-models

C. Additional Experimental Results

C.1. Label Error Detection

In this section, we provide the exact performance values for Figures 4 and 5, including AUROC results. We report label error detection performances under various noise levels in Table 10, and provide performances according to the number of data in Table 11. The tables confirm that our relation graph approach achieves the best label error detection performances in all three metrics, regardless of the noise ratio and the number of data.

C.2. OOD Detection

In Tables 12 and 13, we provide OOD detection results on individual datasets mentioned in Section 4.2: Places, SUN, iNaturalist, and Textures. We report the OOD detection performance of MAE-Large in Table 12 and the performance of ResNet-50 in Table 13. The tables demonstrate that our approach achieves the best OOD detection performance on three out of four datasets considered, while achieving the best overall performance. Furthermore, our method shows the best performance on all three metrics with both models, which highlights its effectiveness in detecting OOD data.

D. Additional Qualitative Results

D.1. Relation Map

In Figure 11, we present additional relation maps on ImageNet. We draw the relation maps with MAE-Large and ResNet-50. Note, MAE-Large utilizes masked auto-encoding pre-training whereas ResNet-50 is trained from scratch. From the figure, we observe that the models exhibit similar distributions of positive and negative relations for each data point. However, the ResNet model tends to have an overall larger variance of relation, indicating that the pre-training process reduces the relation variance and is helpful in forming relationships between data.

D.2. Detection Results

We present problematic data samples detected by our algorithm. Specifically, Figure 12 shows ImageNet samples with label errors and their most conflicting data samples with negative relation values. Figure 13 shows the OOD data detected by our algorithm on the ImageNet validation set, indicating the existence of inappropriate data for evalu-

ation. We also present SST2 text samples with label errors and outlier texts in Tables 14 and 15.

Table 10. Label error detection performance on a range of label error ratios (MAE-Large, ImageNet).

Noise ratio	AUROC				AP				TNR95			
	Loss	TracIn	Margin	Relation	Loss	TracIn	Margin	Relation	Loss	TracIn	Margin	Relation
4%	0.864	0.889	0.876	**0.917**	0.328	0.303	0.403	**0.437**	0.229	0.503	0.346	**0.695**
8%	0.864	0.888	0.875	**0.914**	0.424	0.415	0.484	**0.526**	0.276	0.521	0.392	**0.695**
10%	0.865	0.887	0.876	**0.913**	0.466	0.462	0.520	**0.564**	0.303	0.537	0.425	**0.693**
12%	0.865	0.887	0.876	**0.910**	0.497	0.498	0.544	**0.590**	0.334	0.556	0.457	**0.687**
15%	0.865	0.885	0.876	**0.904**	0.530	0.536	0.568	**0.611**	0.374	0.582	0.505	**0.683**

Table 11. Label error detection performance according to the number of data (MAE-Large, ImageNet, 8% label noise).

# data	AUROC				AP				TNR95			
	Loss	TracIn	Margin	Relation	Loss	TracIn	Margin	Relation	Loss	TracIn	Margin	Relation
12k	0.869	0.891	0.878	**0.910**	0.415	0.408	0.481	**0.502**	0.260	0.554	0.376	**0.679**
25k	0.868	0.890	0.878	**0.912**	0.421	0.415	0.486	**0.515**	0.298	0.541	0.424	**0.692**
100k	0.863	0.886	0.874	**0.912**	0.418	0.409	0.477	**0.518**	0.282	0.523	0.412	**0.692**
400k	0.864	0.887	0.875	**0.914**	0.421	0.412	0.481	**0.523**	0.284	0.524	0.398	**0.696**
1.2M	0.864	0.888	0.875	**0.914**	0.424	0.415	0.484	**0.526**	0.276	0.521	0.392	**0.695**

Table 12. OOD detection performance on ImageNet with MAE-Large. *Rel.* denotes Relation.

Dataset	AUROC					AP					TNR95				
	MSP	Logit	Energy	KNN	Rel.	MSP	Logit	Energy	KNN	Rel.	MSP	Logit	Energy	KNN	Rel.
Places	0.835	0.787	0.728	0.861	**0.883**	0.543	0.531	0.446	0.550	**0.618**	0.386	0.106	0.060	0.487	**0.547**
SUN	0.833	0.790	0.738	0.884	**0.894**	0.567	0.557	0.467	0.593	**0.653**	0.315	0.088	0.050	0.560	**0.587**
iNaturalist	0.907	0.881	0.829	0.946	**0.951**	0.699	0.700	0.592	0.736	**0.782**	0.651	0.320	0.127	0.808	**0.810**
Textures	0.853	0.850	0.837	**0.922**	0.921	0.514	0.570	0.553	**0.648**	0.642	0.376	0.216	0.154	0.626	**0.641**
ALL	0.857	0.824	0.776	0.901	**0.911**	0.818	0.808	0.757	0.854	**0.874**	0.428	0.138	0.074	0.590	**0.636**

Table 13. OOD detection performance on ImageNet with ResNet-50. *Rel.* denotes Relation.

Dataset	AUROC					AP					TNR95				
	MSP	Logit	Energy	KNN	Rel.	MSP	Logit	Energy	KNN	Rel.	MSP	Logit	Energy	KNN	Rel.
Places	0.829	0.827	0.821	0.743	**0.830**	0.469	0.445	0.398	0.335	**0.493**	**0.471**	0.463	0.463	0.295	0.429
SUN	0.836	0.833	0.825	0.783	**0.853**	0.501	0.468	0.407	0.384	**0.543**	0.461	0.448	0.448	0.372	**0.498**
iNaturalist	0.896	0.894	0.883	0.884	**0.922**	0.641	0.605	0.503	0.546	**0.708**	0.634	0.622	0.622	0.587	**0.691**
Textures	0.814	0.807	0.798	**0.922**	0.879	0.369	0.331	0.269	**0.774**	0.513	0.352	0.340	0.340	**0.496**	0.465
ALL	0.847	0.844	0.836	0.822	**0.870**	0.782	0.767	0.721	0.764	**0.818**	0.496	0.482	0.481	0.380	**0.515**

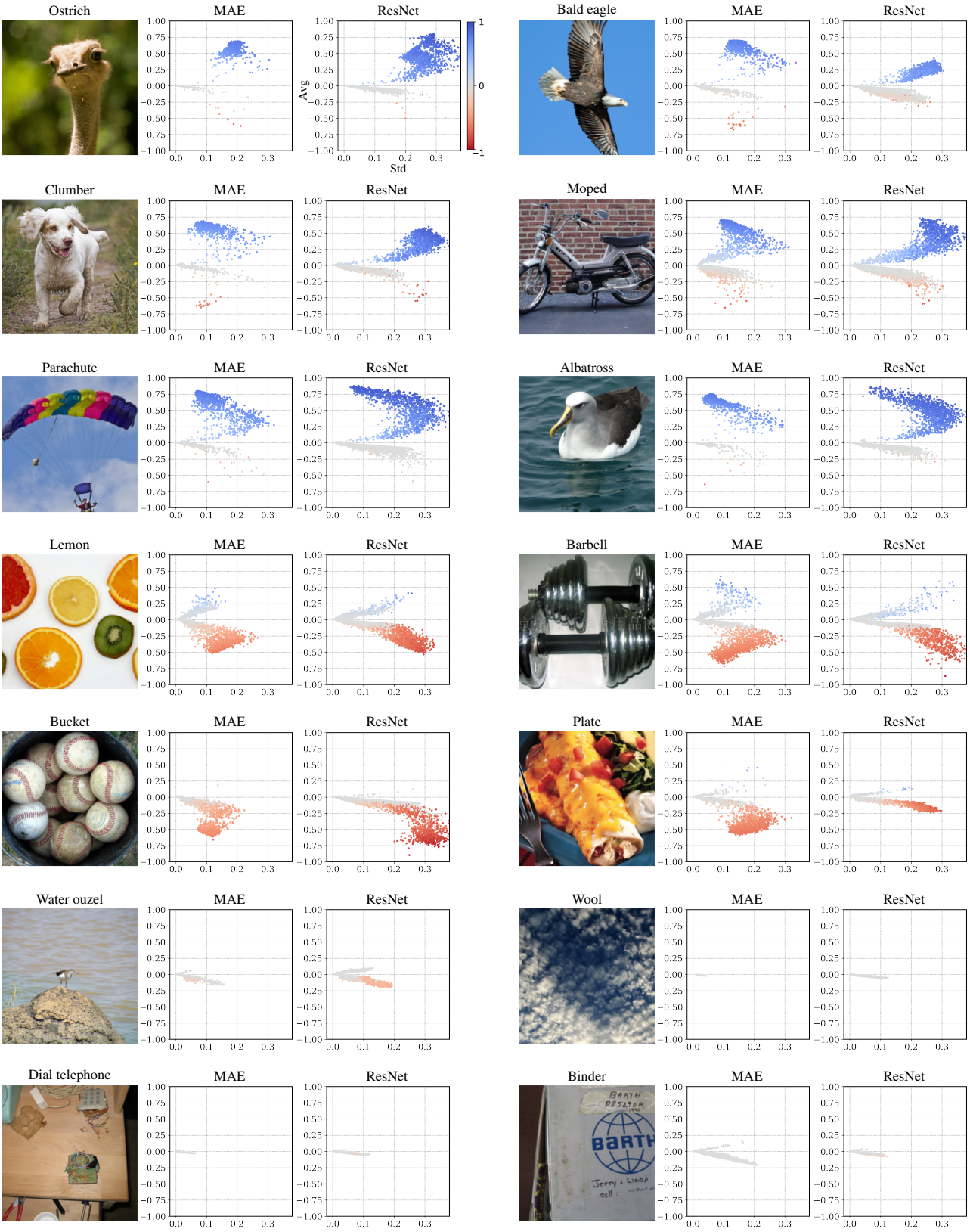


Figure 11. Data relation maps on ImageNet with MAE-Large and ResNet-50. We denote the assigned label above each image. The color represents the relation value at the last checkpoint. The x-axis is the standard deviation and the y-axis is the mean value of the relation values throughout training.

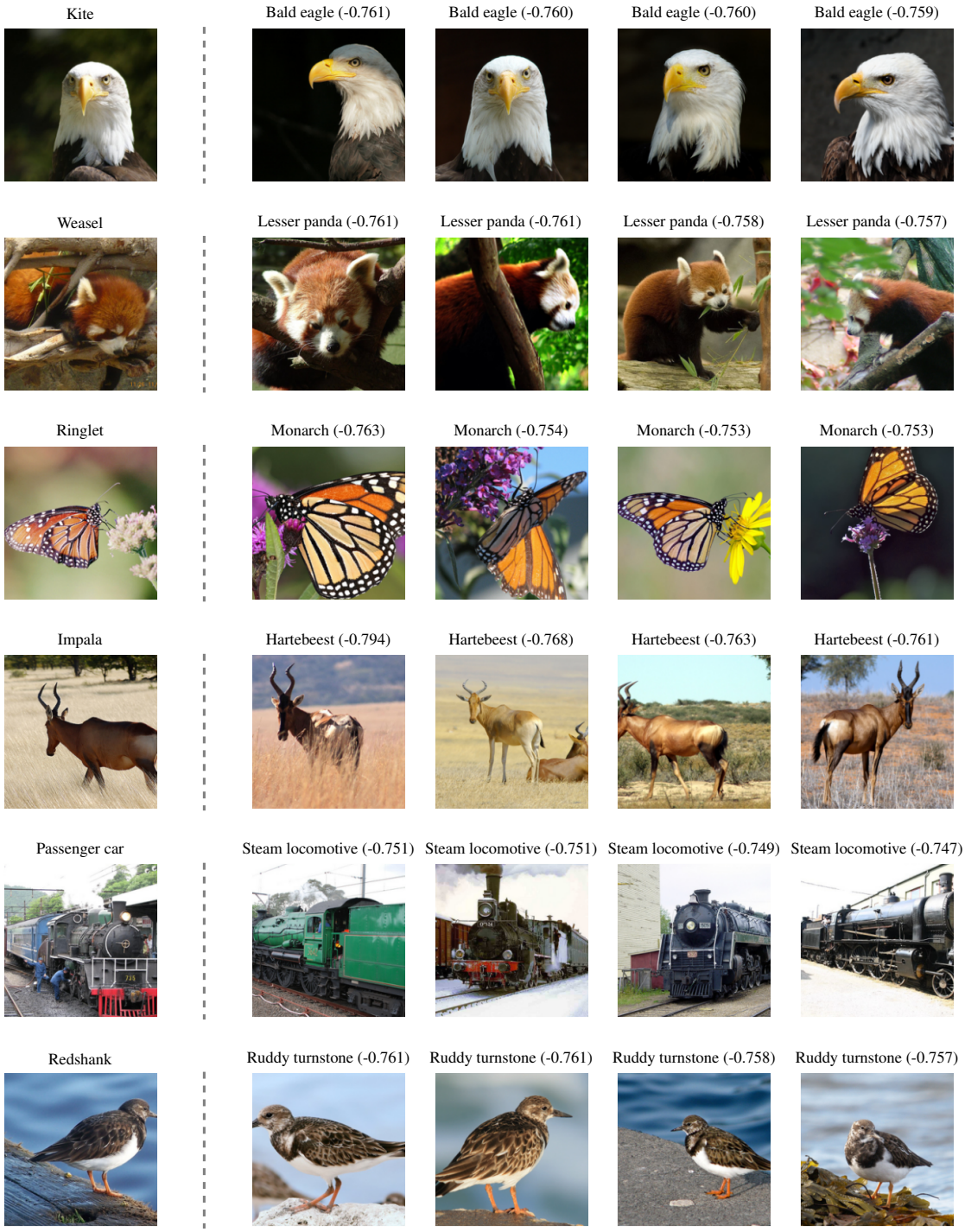


Figure 12. The first column shows the data samples detected by our label error detection algorithm using MAE-Large on ImageNet. We present the samples with the most conflicting relation next to the detected samples. We denote the assigned label and the corresponding relation value above the image.

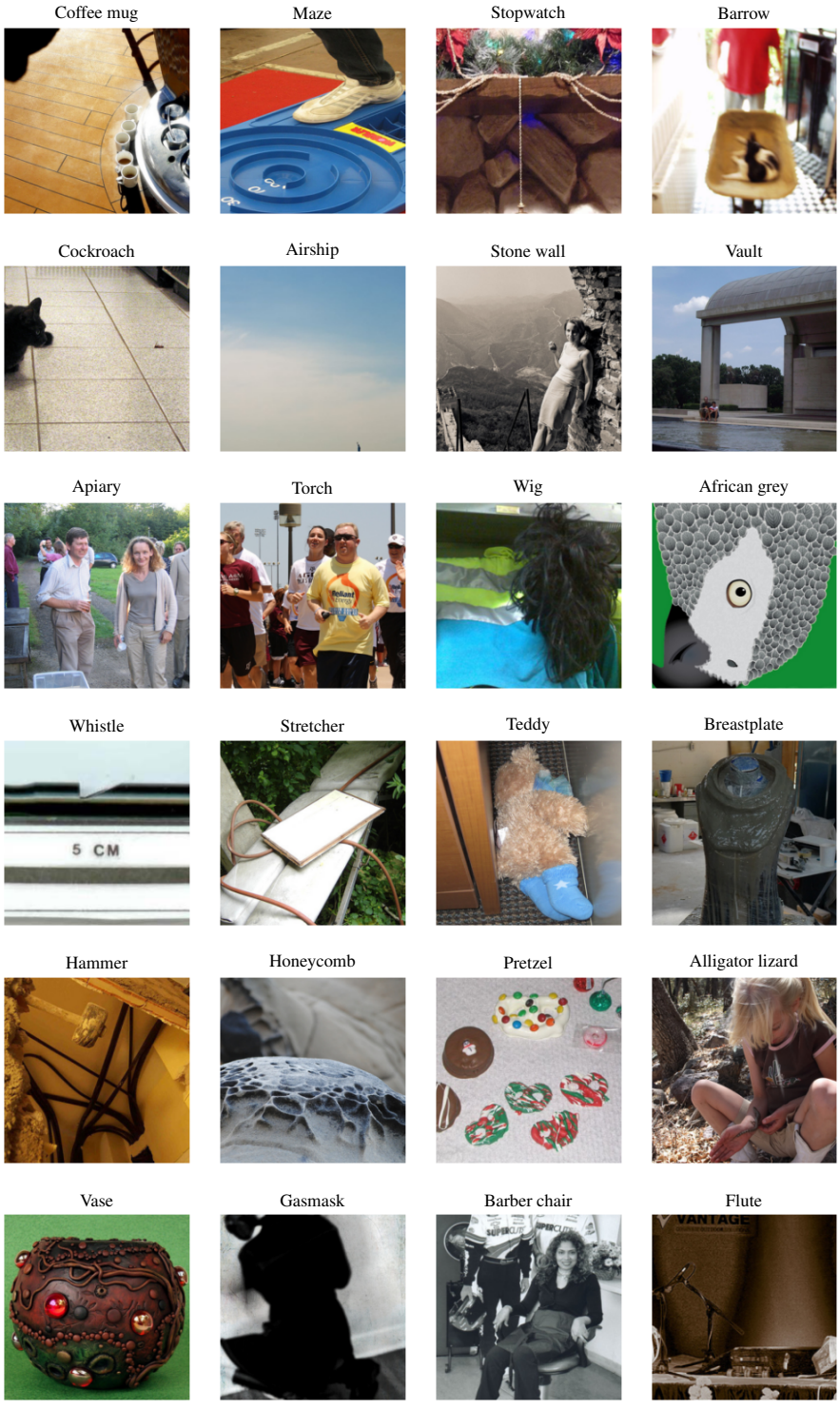


Figure 13. Data samples with the lowest OOD score by our method on the ImageNet validation set. We denote the assigned label above the image.

Table 14. Text samples with label errors detected by our algorithm on the SST2 dataset. Below the text with label error, we present two text samples with conflicting relations and denote the corresponding relation value in parenthesis.

Text	Label
a damn fine and a truly distinctive and a deeply pertinent film	Negative
- a breathtakingly assured and stylish work	Positive (-0.980)
- a winning and wildly fascinating work	Positive (-0.978)
fails to have a heart, mind or humor of its own	Positive
- failing to find a spark of its own	Negative (-0.970)
- this movie's lack of ideas	Negative (-0.958)
a ploddingly melodramatic structure	Positive
- plodding action sequences	Negative (-0.959)
- plodding picture	Negative (-0.957)
is somewhat problematic	Positive
- the more problematic aspects	Negative (-0.945)
- the problematic script	Negative (-0.930)
a bittersweet contemporary comedy	Negative
- bittersweet film	Positive (-0.940)
- of bittersweet camaraderie and history	Positive (-0.921)

Table 15. Outlier text samples detected by our algorithm on the SST2 dataset.

Text	Label
the battle	Negative
give a backbone to the company	Positive
leather pants	Positive
the israeli/palestinian conflict as	Negative
the story relevant in the first place	Positive
from a television monitor	Positive
loud, bang-the-drum	Positive
a movie instead of an endless trailer	Negative
a doctor's office, emergency room, hospital bed or insurance company office	Positive
this is more appetizing than a side dish of asparagus	Negative