EVALUATION OF SIMILARITY-BASED EXPLANATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Explaining predictions made by complex machine learning models helps users understand and accept the predicted outputs with confidence. One way to achieve this goal is to use similarity-based explanation that provides similar instances as evidence to support a model's prediction. Several *relevance metrics* are used for this purpose. In this study, we investigate which relevance metric can provide a reasonable explanation to users. Specifically, we adopted three tests to evaluate whether the relevance metrics satisfy the minimal requirements for similarity-based explanation. Our experiments reveal that the cosine similarity of the gradients of the loss performs best, which would be a recommended choice in practice. In addition, we show that some metrics performs poorly in our tests, and analyze the reasons of their failure. We expect our insights to help practitioners to select appropriate relevance metrics, and also to help further researches for designing better relevance metrics for explanations.

1 INTRODUCTION

Explaining predictions made by complex machine learning models helps users understand and accept the predicted outputs with confidence (Ribeiro et al., 2016; Lundberg & Lee, 2017; Guidotti et al., 2018; Adadi & Berrada, 2018; Molnar, 2020). Instance-based explanation attains this goal by presenting one or several training instances that support the model's prediction. Several types of instance-based explanation have been proposed: explaining with instances similar to the instance of interest (i.e. the test instance in question) (Charpiat et al., 2019; Barshan et al., 2020); harmful instances that degrade the performance of the model (Koh & Liang, 2017; Khanna et al., 2019); counter-examples that contrast how the prediction can be changed (Wachter et al., 2018); and irregular instances (Kim et al., 2016).

Among these types, we focus on the type of explanation that gives one or several training instances similar to a test instance in question and the model's prediction for it. We refer to this type of instance-based explanation as *similarity-based explanation*. A similarity-based explanation is of the form "I (the model) think this image is cat because similar images I saw in the past were also cat." This type of explanation is analogous to the way humans make decisions by referring to their prior experiences (Klein & Calderwood, 1988; Klein, 1989; Read & Cesa, 1991). Hence, this type of explanation tends to be easy to understand even to users with little expertise about machine learning. There is also a report that with this type of explanation, users tend to have higher confidence in model predictions compared to explanations that presents contributing features (Cunningham et al., 2003).

In the instance-based explanation paradigm, including similarity-based explanation, a relevance metric $R(\boldsymbol{z}, \boldsymbol{z}') \in \mathbb{R}$ is typically used to quantify the relationship between two instance $\boldsymbol{z} = (\boldsymbol{x}, y)$ and $\boldsymbol{z}' = (\boldsymbol{x}', y')$.

Definition 1 (Instance-based Explanation Using Relevance Metric)**.** Let $\mathcal{D} = \{\boldsymbol{z}^{(i)}_{\mathrm{train}} = (\boldsymbol{x}^{(i)}_{\mathrm{train}}, y^{(i)}_{\mathrm{train}})\}_{i=1}^{N}$ be a set of training instances, and $\boldsymbol{x}_{\mathrm{test}}$ be a test input of interest whose predicted output is given by $\widehat{y}_{\mathrm{test}} = f(\boldsymbol{x}_{\mathrm{test}})$ with a model f. An instance-based explanation method gives the most relevant training instance $\bar{\boldsymbol{z}} \in \mathcal{D}$ to the test instance $\boldsymbol{z}_{\mathrm{test}} = (\boldsymbol{x}_{\mathrm{test}}, \widehat{y}_{\mathrm{test}})$ by $\bar{\boldsymbol{z}} = \arg\max_{\boldsymbol{z}_{\mathrm{train}} \in \mathcal{D}} R(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}_{\mathrm{train}})$ using a relevance metric $R(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}_{\mathrm{train}})$.

Previously proposed relevance metrics include *similarity* (Caruana et al., 1999), *kernel functions* (Kim et al., 2016; Khanna et al., 2019), and *influence function* (Koh & Liang, 2017).

Table 1: The relevance metrics and their evaluation results. For the model randomization test, the results that passed the test are colored. For the identical class test and identical subclass test, the results with the five highest average evaluation scores are colored. The details of the relevance metrics, the evaluation criteria, and the evaluation procedures can be found in Sections 1.2, 3, and 4, respectively.

Relevance Metrics		Abbrv.	Evaluation Criteria		
			Model Randomization Test	Identical Class Test	Identical Subclass Test
ℓ_2	$\phi(\boldsymbol{z}) = \boldsymbol{x}$	ℓ_2^x	Failed	0.557 ± 0.243	0.587 ± 0.249
	$\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{last}}$	ℓ_2^{last}	Passed	0.865 ± 0.108	0.578 ± 0.215
	$\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{all}}$	ℓ_2^{all}	Failed	0.828 ± 0.128	0.645 ± 0.194
Cosine	$\phi(\boldsymbol{z}) = \boldsymbol{x}$	$\cos^x$	Failed	0.586 ± 0.230	0.621 ± 0.242
	$\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{last}}$	$\cos^{\text{last}}$	Passed	0.874 ± 0.100	0.586 ± 0.216
	$\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{all}}$	$\cos^{\text{all}}$	Failed	0.855 ± 0.111	0.704 ± 0.156
Dot	$\phi(\boldsymbol{z}) = \boldsymbol{x}$	dot^x	Failed	0.329 ± 0.203	0.338 ± 0.218
	$\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{last}}$	dot^{last}	Failed	0.649 ± 0.323	0.307 ± 0.116
	$\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{all}}$	dot^{all}	Failed	0.716 ± 0.310	0.548 ± 0.242
Gradient	Influence Function	IF	Passed	0.379 ± 0.283	0.306 ± 0.174
	Relative IF	RIF	Passed	0.758 ± 0.332	0.643 ± 0.281
	Fisher Kernel	FK	Passed	0.238 ± 0.102	0.190 ± 0.074
	Grad-Dot	GD	Passed	0.752 ± 0.273	0.413 ± 0.136
	Grad-Cos	GC	Passed	0.998 ± 0.003	0.729 ± 0.198

An immediate, critical question is which relevance metric is appropriate for which type of instance-based explanations. There is no doubt that different types of explanations require different metrics. Despite its potential importance, however, little has been explored on this question. Given this background, in this paper, focusing on similarity-based explanation, we investigate which relevance metrics are appropriate through comprehensive experiments.

Contributions We provide the first answer to the question about which relevance metrics have desirable properties for similarity-based explanation. For this purpose, we propose to use three "minimal requirement" tests to evaluate a broad variety of relevance metrics in terms of its appropriateness. The first test is the model randomization test originally proposed by Adebayo et al. (2018) for evaluating saliency-based methods, and the other two tests, the identical class test and identical subclass test, are newly designed in this study. As summarized in Table 1, our experiments reveal that (i) the cosine similarity of gradients performs best, which is probably a recommended choice for similarity-based explanation in practice, and (ii) some relevance metrics, such as influence function, have surprisingly poor performances on the identical class test and identical subclass test, indicating that their use should be deprecated for similarity-based explanation. We also analyze why some metrics are successful and why some are not. We expect these insights to help practitioners to select appropriate relevance metrics.

1.1 Preliminaries

Notations For vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^p$, we denote the dot product by $\langle \boldsymbol{a}, \boldsymbol{b} \rangle := \sum_{i=1}^p a_i b_i$, the ℓ_2 norm by $\|\boldsymbol{a}\| := \sqrt{\langle \boldsymbol{a}, \boldsymbol{a} \rangle}$, and the cosine similarity by $\cos(\boldsymbol{a}, \boldsymbol{b}) := {}^{\langle \boldsymbol{a}, \boldsymbol{b} \rangle}/_{\|\boldsymbol{a}\| \|\boldsymbol{b}\|}$.

Classification Problem We consider a standard classification problem as the evaluation benchmark. The model is the conditional probability $p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$ with parameter $\boldsymbol{\theta}$. Let $\widehat{\boldsymbol{\theta}}$ be a trained parameter $\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{train}} := \frac{1}{N} \sum_{i=1}^N \ell(\boldsymbol{z}_{\text{train}}^{(i)}; \boldsymbol{\theta})$, where the loss function ℓ is the cross entropy $\ell(\boldsymbol{z}; \boldsymbol{\theta}) = -\log p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$ for an input-output pair $\boldsymbol{z} = (\boldsymbol{x}, y)$. The model classifies a test input $\boldsymbol{x}_{\text{test}}$ by assigning the class with the highest probability $\widehat{y}_{\text{test}} = \arg\max_y p(y \mid \boldsymbol{x}_{\text{test}}; \widehat{\boldsymbol{\theta}})$.

1.2 Relevance Metrics

We present an overview of the two types of relevance metrics considered in this study, *similarity metrics* and *gradient-based metrics*. See Table 1 for the list of metrics and their abbreviations.

Similarity Metrics We consider the following popular similarity metrics with a feature map $\phi(\boldsymbol{z})$.

- **ℓ_2 Metric**: $R_{\ell_2}(\boldsymbol{z}, \boldsymbol{z}') := -\|\phi(\boldsymbol{z}) - \phi(\boldsymbol{z}')\|^2$, which is a typical choice for nearest neighbor methods (Hastie et al., 2009; Abu Alfeilat et al., 2019).
- **Cosine Metric**: $R_{\cos}(\boldsymbol{z}, \boldsymbol{z}') := \cos(\phi(\boldsymbol{z}), \phi(\boldsymbol{z}'))$, which is commonly used in natural language processing tasks (Mikolov et al., 2013; Arora et al., 2017; Conneau et al., 2017).
- **Dot Metric**: $R_{\mathrm{dot}}(\boldsymbol{z}, \boldsymbol{z}') := \langle \phi(\boldsymbol{z}), \phi(\boldsymbol{z}') \rangle$, which is a kernel function used in kernel models such as SVM (Schölkopf et al., 2002; Fan et al., 2005; Bien & Tibshirani, 2011).

As the feature map $\phi(\boldsymbol{z})$, we consider (i) an identity map $\phi(\boldsymbol{z}) = \boldsymbol{x}$; (ii) the last hidden layer $\phi(\boldsymbol{z}) = \boldsymbol{h}^{\mathrm{last}}$, which is the latent representation of the input $\boldsymbol{x}$, one layer before the output in a deep neural network; and, (iii) all hidden layers $\phi(\boldsymbol{z}) = \boldsymbol{h}^{\mathrm{all}}$, where $\boldsymbol{h}^{\mathrm{all}} = [\boldsymbol{h}^1, \boldsymbol{h}^2, \ldots, \boldsymbol{h}^{\mathrm{last}}]$ is the concatenation of all latent representations in the network. Note that the metrics with the identity map merely measure the similarity of the inputs without model information. We adopt these metrics as naive baselines to contrast with other advanced metrics that utilize the information of the models.

Gradient-based Metrics Gradient-based metrics use the gradient $\boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}} := \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{z}; \widehat{\boldsymbol{\theta}})$ to measure the relevance. We consider the five metrics: Influence Function (IF) (Koh & Liang, 2017), Relative IF (RIF) (Barshan et al., 2020), Fisher Kernel (FK) (Khanna et al., 2019), Grad-Dot (GD) (Yeh et al., 2018; Charpiat et al., 2019), and Grad-Cos (GC) (Perronnin et al., 2010; Charpiat et al., 2019). See Appendix A for the detail.

- **IF**: $R_{\mathrm{IF}}(\boldsymbol{z}, \boldsymbol{z}') := \langle \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}}, \boldsymbol{H}^{-1} \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}'} \rangle$
- **RIF**: $R_{\mathrm{RIF}}(\boldsymbol{z}, \boldsymbol{z}') := \cos(\boldsymbol{H}^{-\frac{1}{2}} \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}}, \boldsymbol{H}^{-\frac{1}{2}} \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}'})$
- **FK**: $R_{\mathrm{FK}}(\boldsymbol{z}, \boldsymbol{z}') := \langle \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}}, \boldsymbol{I}^{-1} \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}'} \rangle$,
- **GD**: $R_{\mathrm{GD}}(\boldsymbol{z}, \boldsymbol{z}') := \langle \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}}, \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}'} \rangle$
- **GC**: $R_{\mathrm{GC}}(\boldsymbol{z}, \boldsymbol{z}') := \cos(\boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}}, \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}'})$

where $\boldsymbol{H}$ and $\boldsymbol{I}$ are the Hessian and Fisher information matrices of the loss $\mathcal{L}_{\mathrm{train}}$, respectively.

2 Related Work

Model-specific Explanation Aside of the relevance metrics, there is another approach for similarity-based explanation that uses specific models that can provide explanations by its design (Kim et al., 2014; Plötz & Roth, 2018; Chen et al., 2019). We set aside these specific models and focus on generic relevance metrics because of their applicability to a wide range of problems.

Evaluation of Metrics for Improving Classification Accuracy In many machine learning problems, metrics between instances play an essential role. For example, the distance between instances is essential for the distance-based methods such as nearest neighbor methods (Hastie et al., 2009). Another example is kernel models where the kernel function represents the relationship between two instances (Schölkopf et al., 2002). There are several studies evaluating which metrics are desirable for specific tasks (Hussain et al., 2011; Hu et al., 2016; Li & Li, 2018; Abu Alfeilat et al., 2019). The goal of these studies is to find metrics that improve classification accuracy. Different from these evaluations based on accuracy, our goal in this study is to evaluate the validity of relevance metrics for similarity-based explanation; thus, the findings in these previous studies are not directly applicable to our goal.

Evaluation of Explanations There are a variety of desiderata argued as requirements for explanations, such as faithfulness (Adebayo et al., 2018; Lakkaraju et al., 2019; Jacovi & Goldberg, 2020), plausibility (Lei et al., 2016; Lage et al., 2019; Strout et al., 2019), robustness (Alvarez-Melis & Jaakkola, 2018), and readability (Wang & Rudin, 2015; Yang et al., 2017; Angelino et al., 2017). Evaluating the existing explanation methods with respect to these requirements is of particular importance. However, there is no standard test established for evaluating these requirements, and the designing of such tests still remains as an open problem (Doshi-Velez & Kim, 2017; Jacovi & Goldberg, 2020). In this study, as the first empirical study for evaluating the existing relevance metrics for similarity-based explanation, we take an alternative approach by designing "minimal requirements" tests for the two primary requirements, faithfulness and plausibility, instead of jumping into the direct evaluation of these requirements.

Figure 1: A valid (✔) and invalid (✘) examples for the identical class test.

3 Evaluation Criteria for Similarity-based Explanation

The goal of this study is to investigate which relevance metrics have desirable properties for similarity-based explanation. In this section, we propose three tests to evaluate whether relevance metrics satisfy the minimal requirements for similarity-based explanation. If a relevance metric fails one of the tests, we conclude that the metric do not meet the minimal requirements; thus, its use would be deprecated. The first test assesses whether each relevance metric satisfies the minimal requirements for the faithfulness of explanation, which requires that an explanation to a model prediction must reflect the underlying inference process (Adebayo et al., 2018; Lakkaraju et al., 2019; Jacovi & Goldberg, 2020). The latter two tests are designed to assess relevance metrics in terms of the plausibility of the explanations they produce (Lei et al., 2016; Lage et al., 2019; Strout et al., 2019), which requires explanations to be sufficiently convincing to the users.

3.1 Model Randomization Test

Any explanations that are irrelevant to the model should be avoided because such fake explanations can mislead the users. Thus, any valid relevance metric should be model-dependent, which constitutes the the first requirement.

We use the model randomization test of Adebayo et al. (2018) to assess whether the a given relevance metrics satisfies a minimal requirement for the faithfulness. If a relevance metric produces almost same explanations for the same inputs on two models with different reasoning processes, that metric is likely to ignore the underlying model, i.e., the metric is independent to the model. Thus, we can evaluate whether the metric is model-dependent or not by comparing explanations from two different models. A typical choice of the models is a well-trained model that can predict the output well and a randomly initialized model that can make only poor prediction. These two models have different reasoning processes, and hence the explanations should be different.

Definition 2 (Model Randomization Test)**.** Let f and f_{rand} be a well-trained model and a randomly initialized model, respectively. For the test instance $\boldsymbol{z}_{\mathrm{test}} = (\boldsymbol{x}_{\mathrm{test}}, \widehat{y}_{\mathrm{test}})$ and for the model f, let π_f be a permutation of the training instances sorted based on the relevance metric R, i.e., $R(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}_{\mathrm{train}}^{(\pi_f(1))}) \geq R(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}_{\mathrm{train}}^{(\pi_f(2))}) \geq \ldots \geq R(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}_{\mathrm{train}}^{(\pi_f(N))})$. We also define $\pi_{f_{\mathrm{rand}}}$ accordingly. Then, we require π_f and $\pi_{f_{\mathrm{rand}}}$ to have a small rank correlation.

If the relevance metric R is independent of the model, it produces the same permutation both for f and f_{rand} and their rank correlation becomes one. If the rank correlation is significantly smaller than one and is close to zero, we can confirm that the relevance metric is model-dependent.

3.2 Identical Class Test

The second minimal requirement is that the raised similar instance should belong to the same class as the test instance, as shown in Figure 1. The violation of this requirement leads to nonsense explanations such as "I think this image is cat because a similar image I saw in the past was dog." in Figure 1, which do not make sense to the users at all. When faced with such an explanation, the users will find the validity of the model's prediction questionable, and will ignore the prediction even if the underlying model is valid. This observation leads to the identical class test below.

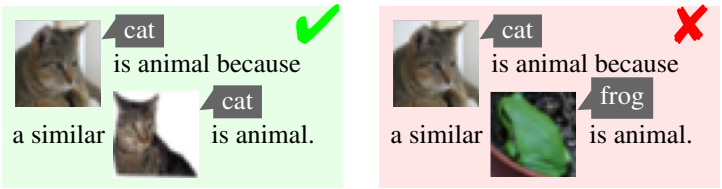


Figure 2: A valid (✔) and invalid (✘) examples for the identical subclass test.

Definition 3 (Identical Class Test)**.** The most similar instance for the prediction of a test instance $\boldsymbol{z}_{\text{test}} = (\boldsymbol{x}_{\text{test}}, \widehat{y}_{\text{test}})$ is a training instance of the same class as the given test instance.

$$\underset{\boldsymbol{z}=(\boldsymbol{x},y)\in\mathcal{D}}{\arg\max}\ R\big(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}\big) = (\bar{\boldsymbol{x}}, \bar{y}) \implies \bar{y} = \widehat{y}_{\text{test}}. \tag{1}$$

Although this test may look trivial, some relevance metrics do not satisfy this minimal requirement, as demonstrated in Section 4.2.

3.3 Identical Subclass Test

The third minimal requirement is that the raised similar instance should belong to the same subclass as the test instance when the the classes consist of latent subclasses, as shown in Figure 2. For example, consider a problem of classifying images of CIFAR10 to two classes, animal and vehicle. The animal class consists of images from subclasses such as cat and frog, while the vehicle class consists of images from subclasses such as airplane and automobile. Under the presence of subclasses, the violation of this third requirement again leads to nonsense explanations such as "I think this image (cat) is animal because a similar image (frog) I saw in the past was also animal." in Figure 2. This observation leads to the identical subclass test below.

Definition 4 (Identical Subclass Test)**.** Let $s(\boldsymbol{z})$ be its subclass for an instance $\boldsymbol{z} = (\boldsymbol{x}, y)$. Here, the most similar instance for the prediction of a test instance $\boldsymbol{z}_{\text{test}} = (\boldsymbol{x}_{\text{test}}, \widehat{y}_{\text{test}})$ with a correct prediction $\widehat{y}_{\text{test}} = y_{\text{test}}$ is the training instance of the same subclass as the test instance.[1]

$$\underset{\boldsymbol{z}\in\mathcal{D}}{\arg\max}\, R\big(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}\big) = \widehat{\boldsymbol{z}} \implies s(\widehat{\boldsymbol{z}}) = s(\boldsymbol{z}_{\text{test}}). \tag{2}$$

Note that the identical subclass test depends on whether the model has obtained internal representations that can distinguish between subclasses. If the model mixes up subclasses in its reasoning process, any explanation will fail the test. To confirm the validity of the identical subclass test in our experiments, we verified that all the models we used satisfied this condition. See Appendix E.

4 Evaluation Results

Here, we examine the validity of the relevance metrics with respect to the three minimal requirements. For this evaluation, we used two image datasets (MNIST (LeCun et al., 1998), CIFAR10 (Krizhevsky, 2009)), two text datasets (TREC (Li & Roth, 2002), AGNews (Zhang et al., 2015)) and one table dataset (Vehicle (Dua & Graff, 2017)). As benchmarks, we employed logistic regression and deep neural networks trained on these datasets. The details of the datasets, models, and computing infrastructure we used are given in Appendix B.

Procedure We repeated the following procedure for 10 times for each evaluation test.

1. Train a model using a subset of the training instances.[2] Then, randomly sample 500 test instances from the test set.
2. For each test instance, compute the relevance score for all the instances used for training.
3. (a) For model randomization test, compute the Spearman rank correlation coefficients between the relevance scores from the trained model and relevance scores from the randomized model.

[1] We require correct predictions in this test because the subclass does not match for incorrect cases.

[2] We randomly sampled 10% of MNIST and CIFAR10, 50% of TREC, 5% of AGNews, and 50% of Vehicle.

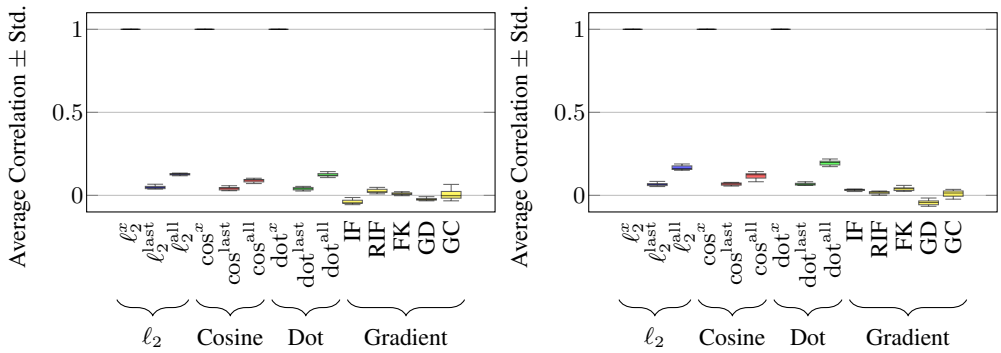


Figure 3: Result of the model randomization test. The correlation close to zero is ideal.

(b) For identical class test and identical subclass test, compute the success rate, which is the ratio of test instances that passed the test.

In this section, we mainly present the results for CIFAR10 with CNN, and AGNews with Bi-LSTM. The other results were similar, and can be found in Appendix F.

Summary of the Results We summarize the main results prior to discussing individual result.

- ℓ_2^{last}, $\cos^{\text{last}}$, and the gradient-based metrics scored law correlation in the model randomization test for all the datasets and models, indicating that they are model-dependent.
- GC performed the best in most of the identical class test and identical subclass test; thus, GC would be the recommended choice in practice.
- Dot metrics as well as IF, FK, and GD performed poorly on the identical class test and identical subclass test.

In Section 5, we analyze why some relevance metrics are successful and why some are not in the identical class test and identical subclass test.

4.1 Result of Model Randomization Test

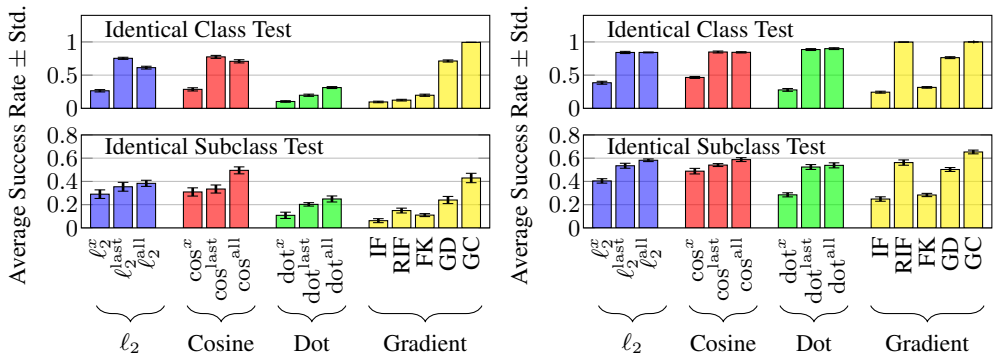
Figure 3 shows the Spearman rank correlation coefficients for the model randomization test. The similarities with the identity feature map ℓ_2^x, $\cos^x$, and dot^x are irrelevant to the model and their correlations are trivially one. In the figures, the other metrics scored correlations close to zero, indicating they will be model-dependent. However, on MNIST and Vehicle datasets (see Appendix F), we observed that ℓ_2^{all}, $\cos^{\text{all}}$, dot^{last}, and dot^{all} scored correlations more than 0.7. We therefore concluded that these relevance metrics failed the model randomization test because they can raise instances irrelevant to the model for some datasets.

4.2 Results of Identical Class Test and Identical Subclass Test

For the identical subclass test, we used modified datasets: we split the dataset into two classes by randomly assigning existing classes to one of the two classes "A" and "B". The new classes "A" and "B" now contain the original data classes as subclasses, which can be used for the identical subclass test. We sampled 500 instances with correct predictions from the test set as test instances.

Figure 4 shows the success rates for the identical class test and identical subclass test. We also summarized the average success rates of all of our experiments in Table 1. A remarkable observation is that GC performed consistently well both on the identical class test and identical subclass test for all the datasets and the models we used in the experiment (see Appendix F). By contrast, some relevance metrics such as dot metrics as well as IF, FK, and GD performed poorly on both of the tests. We investigate the reason of their failure in the next section.

To conclude, the results of our evaluations indicate that only GC performed well on all the tests. That is, only GC seems to meet the minimal requirements; thus, it would be a recommended choice for similarity-based explanation.



(a) CIFAR10 with CNN (b) AGNews with Bi-LSTM

Figure 4: Results of the identical class test and identical subclass test.

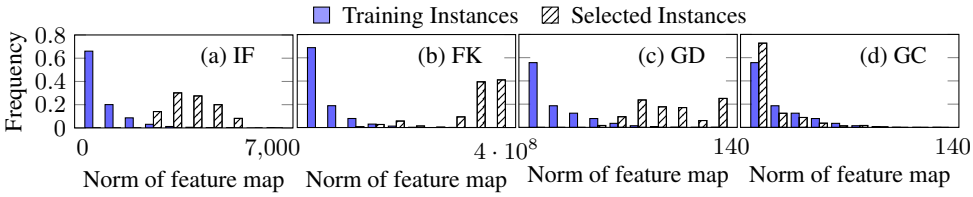


Figure 5: Distributions of the norms of the feature maps of all the training instances (colored), and the instances selected by the identical class test (meshed) on CIFAR10 with CNN.

5 WHY SOME METRICS ARE SUCCESSFUL AND WHY SOME ARE NOT

We observed that the dot metrics and the gradient-based metrics such as IF, FK, and GD, failed the identical class test and identical subclass test, in contrast to GC who performed remarkably well. Here, we analyze the reason why the above relevance metrics failed and why GC performed well. In Appendix D, we also discuss a way to *repair* IF, FK, and GD to improve their performances based on the findings in this section.

Failure of the Dot Metrics and Gradient-based Metrics To understand the failure, we reformulate IF, FK, and GD as dot metrics of the form $R_{\text{dot}}(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}_{\text{train}}) = \langle \phi(\boldsymbol{z}_{\text{test}}), \phi(\boldsymbol{z}_{\text{train}}) \rangle$ so that the following discussion to hold true for any relevance metrics in this form. It is easy to see that IF, FK, and GD can be expressed in this form by defining the feature maps by $\phi(\boldsymbol{z}) = \boldsymbol{H}^{-1/2} g(\boldsymbol{z}; \widehat{\boldsymbol{\theta}})$, $\phi(\boldsymbol{z}) = \boldsymbol{I}^{-1/2} g(\boldsymbol{z}; \widehat{\boldsymbol{\theta}})$, and $\phi(\boldsymbol{z}) = g(\boldsymbol{z}; \widehat{\boldsymbol{\theta}})$, respectively.

Given a criterion, let $\boldsymbol{z}_{\text{train}}^{(i)}$ be a desirable instance for a test instance $\boldsymbol{z}_{\text{test}}$. The failures of the dot metrics indicate the existence of an undesirable instance $\boldsymbol{z}_{\text{train}}^{(j)}$ such that $\langle \phi(\boldsymbol{z}_{\text{test}}), \phi(\boldsymbol{z}_{\text{train}}^{(i)}) \rangle < \langle \phi(\boldsymbol{z}_{\text{test}}), \phi(\boldsymbol{z}_{\text{train}}^{(j)}) \rangle$. The following sufficient condition for $\boldsymbol{z}_{\text{train}}^{(j)}$ is useful to understand the failure.

$$\|\phi(\boldsymbol{z}_{\text{train}}^{(i)})\| < \|\phi(\boldsymbol{z}_{\text{train}}^{(j)})\| \cos(\phi(\boldsymbol{z}_{\text{test}}), \phi(\boldsymbol{z}_{\text{train}}^{(j)})). \tag{3}$$

The condition implies that any instance with an extremely large norm and cosine slightly larger than zero can be the candidate of $\boldsymbol{z}_{\text{train}}^{(j)}$. In our experiments, we observed that the condition on the norm is especially crucial. Indeed, as we can see in Figure 5, even though instances with extremely large norms were scarce, only such extreme instances were selected as relevant instances by IF, FK, and GD. That is, these metrics tend to consider such extreme instances as relevant ones. By contrast, GC was not attracted by large norms because it completely cancels out the norm through normalization.

Figure 6 shows some training instances frequently selected in the identical class test on CIFAR10 with CNN. When using IF, FK, and GD, these training instances were frequently selected regardless of their classes, simply because the training instances had large norms. In these metrics, the term $\cos(\phi(\boldsymbol{z}_{\text{test}}), \phi(\boldsymbol{z}_{\text{train}}))$ seems to have only negligible effects. By contrast, GC successfully selected instances of the same class and ignored instances with large norms.

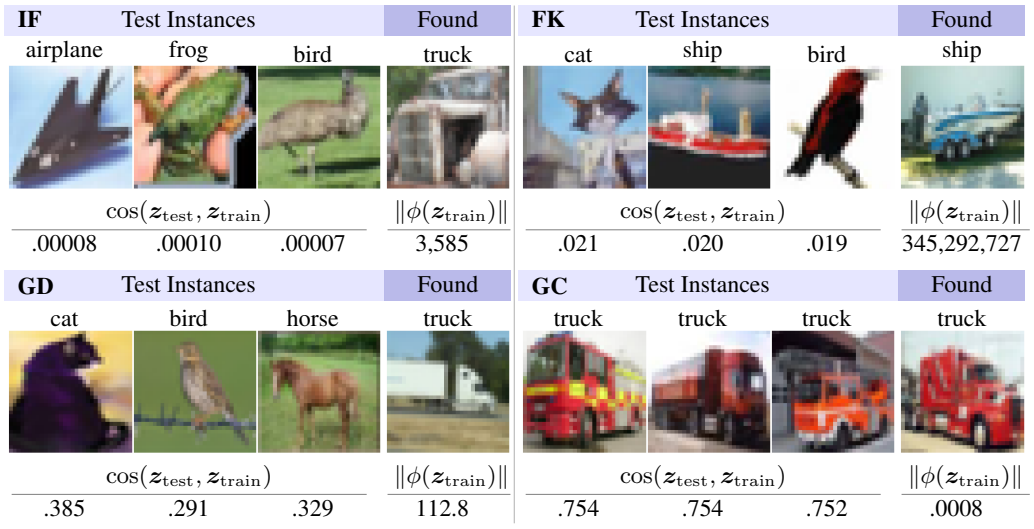


Figure 6: Training instances frequently selected in the identical class test with multiple test instances on CIFAR10 with CNN, the cosine between them, and the norm of the training instances.

Success of GC We now analyze why GC performed well in particular in the identical class test. To simplify the discussion, we consider linear logistic regression whose conditional distribution $p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$ is given by the y-th entry of $\sigma(W\boldsymbol{x})$, where σ is the softmax function, $\boldsymbol{\theta} = W \in \mathbb{R}^{C \times d}$, and C and d are the number of classes and the dimensionality of $\boldsymbol{x}$, respectively. With some algebra, we obtain $R_{\text{GC}}(\boldsymbol{z}, \boldsymbol{z}') = \cos(\boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'}) \cos(\boldsymbol{x}, \boldsymbol{x}')$ for $\boldsymbol{z} = (\boldsymbol{x}, y)$ and $\boldsymbol{z}' = (\boldsymbol{x}', y')$, where $\boldsymbol{r}^{\boldsymbol{z}} = \sigma(W\boldsymbol{x}) - \boldsymbol{e}_y$ is the *residual* for the prediction on $\boldsymbol{z}$ and $\boldsymbol{e}_y$ is a vector whose y-th entry is one and zero otherwise. See Appendix C for the derivation. Here, the term $\cos(\boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'})$ plays an essential role in GC. From the definition, $r^{\boldsymbol{z}}_c \leq 0$ if $c = y$ and $r^{\boldsymbol{z}}_c \geq 0$ otherwise. Thus, $\cos(\boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'}) \geq 0$ always holds true when $y = y'$, while $\cos(\boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'})$ can be negative for $y \neq y'$. Hence, the chance of $R_{\text{GC}}(\boldsymbol{z}, \boldsymbol{z}')$ being positive can be larger for the instances from the same class compared to the ones from a different class.

Figure 7 shows that $\cos(\boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'})$ is essential also for deep neural networks. Here, for each test instance $\boldsymbol{z}_{\text{test}}$ on CIFAR10 with CNN, we randomly sampled two training instances $\boldsymbol{z}_{\text{train}}$, one with the same class and the other with a different class, and computed $R_{\text{GC}}(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}_{\text{train}})$ and $\cos(\boldsymbol{r}^{\boldsymbol{z}_{\text{test}}}, \boldsymbol{r}^{\boldsymbol{z}_{\text{train}}})$.

We also note that $\cos(\boldsymbol{r}^{\boldsymbol{z}_{\text{test}}}, \boldsymbol{r}^{\boldsymbol{z}_{\text{train}}})$ alone was not helpful for the identical subclass test, whose success rate was around the chance level. We thus conjecture that, while $\cos(\boldsymbol{r}^{\boldsymbol{z}_{\text{test}}}, \boldsymbol{r}^{\boldsymbol{z}_{\text{train}}})$ is particularly helpful for the identical class test, the use of the entire gradient is still essential for GC to work well.

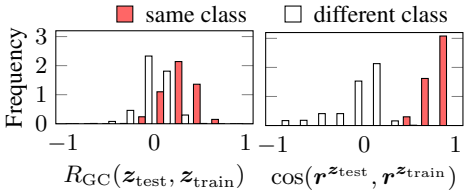


Figure 7: Distributions of $R_{\text{GC}}(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}_{\text{train}})$ and $\cos(\boldsymbol{r}^{\boldsymbol{z}_{\text{test}}}, \boldsymbol{r}^{\boldsymbol{z}_{\text{train}}})$ for training instances with the same / different classes on CIFAR10 with CNN.

6 Conclusion

We investigated which of the relevance metrics are effective for similarity-based explanation. For this purpose, we evaluated whether the metrics satisfy the minimal requirements for similarity-based explanation. In this study, we conducted three tests: the model randomization test of Adebayo et al. (2018) to evaluate whether the metrics are model-dependent, and the two newly designed tests, the identical class test and identical subclass test, to evaluate whether the metrics can provide plausible explanation. Our quantitative evaluations based on the three tests revealed that the cosine similarity of the gradients performs best, which would be a recommended choice in practice. We also observed that some relevance metrics do not meet the requirements; thus, the use of such metrics would not be appropriate for similarity-based explanation. We expect our insights to help practitioners to select appropriate relevance metrics, and also to help further researches for designing better relevance metrics for instance-based explanations.

REFERENCES

Haneen Arafat Abu Alfeilat, Ahmad B.A. Hassanat, Omar Lasassmeh, Ahmad S. Tarawneh, Mahmoud Bashir Alhasanat, Hamzeh S. Eyal Salman, and V.B. Surya Prasath. Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. *Big Data*, 7(4): 221–248, 2019.

Amina Adadi and Mohammed Berrada. Peeking Inside the Black-box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity Checks for Saliency Maps. In *Advances in Neural Information Processing Systems 31*, pp. 9505–9515. 2018.

David Alvarez-Melis and Tommi S Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018.

Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. *The Journal of Machine Learning Research*, 18 (1):8753–8830, 2017.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. RelatIF: Identifying Explanatory Training Samples via Relative Influence. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pp. 1899–1909, 2020.

Jacob Bien and Robert Tibshirani. Prototype Selection for Interpretable Classification. *Annals of Applied Statistics*, 5(4):2403–2424, 2011.

Rich Caruana, Hooshang Kangarloo, John David N. Dionisio, Usha Sinha, and David Johnson. Case-Based Explanation of Non-Case-Based Learning Methods. In *Proceedings of the AMIA Symposium*, pp. 212–215, 1999.

Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. Input Similarity from the Neural Network Perspective. In *Advances in Neural Information Processing Systems 32*, pp. 5342–5351. 2019.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This Looks Like That: Deep Learning for Interpretable Image Recognition. In *Advances in Neural Information Processing Systems 32*, pp. 8930–8941. 2019.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 670–680, 2017.

Pádraig Cunningham, Dónal Doyle, and John Loughrey. An Evaluation of the Usefulness of Case-Based Explanation. In *International Conference on Case-Based Reasoning*, pp. 122–130. Springer, 2003.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. Working Set Selection Using Second Order Information for Training Support Vector Machines. *Journal of Machine Learning Research*, 6: 1889–1918, 2005.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys*, 51 (5):1–42, 2018.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

Li Yu Hu, Min Wei Huang, Shih Wen Ke, and Chih Fong Tsai. The Distance Function Effect on k-Nearest Neighbor Classification for Medical Datasets. *SpringerPlus*, 5(1):1304, 2016.

Muhammad Hussain, Summrina Kanwal Wajid, Ali Elzaart, and Mohammed Berbar. A Comparison of SVM Kernel Functions for Breast Cancer Detection. In *Proceedings of the 8th International Conference on Computer Graphics, Imaging and Visualization*, pp. 145–150, 2011.

Alon Jacovi and Yoav Goldberg. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4198–4205, 2020.

Rajiv Khanna, Been Kim, Joydeep Ghosh, and Sanmi Koyejo. Interpreting Black Box Predictions using Fisher Kernels. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, volume 89, pp. 3382–3390, 2019.

Been Kim, Cynthia Rudin, and Julie A Shah. The Bayesian Case Model: A Generative Approach for Case-Based Reasoning and Prototype Classification. In *Advances in Neural Information Processing Systems 27*, pp. 1952–1960, 2014.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples Are Not Enough, Learn to Criticize! Criticism for Interpretability. In *Advances in neural information processing systems 29*, pp. 2280–2288, 2016.

Gary A Klein. Strategies of Decision Making. Technical report, 1989.

Gary A Klein and Roberta Calderwood. How Do People Use Analogues to Make Decisions? In *Proceedings of the DARPA Workshop on Case-Based Reasoning, 1988*, pp. 209–223, 1988.

Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1885–1894, 2017.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An Evaluation of the Human-Interpretability of Explanation. 2019. URL `http://arxiv.org/abs/1902.00006`.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 131–138, 2019.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 107–117, 2016.

Xin Li and Dan Roth. Learning Question Classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

Zhou Li and Chunxiang Li. Selection of Kernel Function for Least Squares Support Vector Machines in Downburst Wind Speed Forecasting. In *Proceedings of the 11th International Symposium on Computational Intelligence and Design*, volume 2, pp. 337–341, 2018.

Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, pp. 4765–4774, 2017.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint:1802.03426*, 2018.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. 2013.

Christoph Molnar. *Interpretable Machine Learning*. Lulu. com, 2020.

Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-Scale Image Retrieval With Compressed Fisher Vectors. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3384–3391, 2010.

Tobias Plötz and Stefan Roth. Neural Nearest Neighbors Networks. In *Advances in Neural Information Processing Systems 31*, pp. 1087–1098. 2018.

Stephen J Read and Ian L Cesa. This Reminds Me of the Time When…: Expectation Failures in Reminding and Explanation. *Journal of Experimental Social Psychology*, 27(1):1–25, 1991.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You" Explaining the Predictions of Any Classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

Bernhard Schölkopf, Alexander J Smola, and Francis Bach. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2002.

Julia Strout, Ye Zhang, and Raymond J. Mooney. Do Human Rationales Improve Machine Explanations? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 56–62, 2019.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2): 841–887, 2018.

Fulton Wang and Cynthia Rudin. Falling rule lists. In *Artificial Intelligence and Statistics*, pp. 1013–1022, 2015.

Hongyu Yang, Cynthia Rudin, and Margo Seltzer. Scalable bayesian rule lists. In *International Conference on Machine Learning*, pp. 3921–3930, 2017.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer Point Selection for Explaining Deep Neural Networks. In *Advances in Neural Information Processing Systems 31*, pp. 9291–9301, 2018.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28*, pp. 649–657. 2015.

A GRADIENT-BASED METRICS

In gradient-based metrics, we consider a model with parameter $\boldsymbol{\theta}$, its loss $\ell(\boldsymbol{z};\boldsymbol{\theta})$, and its gradient $\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z};\boldsymbol{\theta})$ to measure relevance, where $\boldsymbol{z} = (\boldsymbol{x}, y)$ is an input-output pair.

Influence Function (Koh & Liang, 2017) Koh & Liang (2017) proposed to measure relevance according to "how largely the test loss will increase if the training instance is omitted from the training set." Here, the model parameter trained using all of the training set is denoted by $\widehat{\boldsymbol{\theta}}$, and the parameter trained using all of the training set except the i-th instance $\boldsymbol{z}^{(i)}_{\mathrm{train}}$ is denoted by $\widehat{\boldsymbol{\theta}}_{-i}$. The relevance metric proposed by Koh & Liang (2017) is then defined as the difference between the test loss under parameters $\widehat{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{\theta}}_{-i}$ as follows:

$$R_{\mathrm{IF}}(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}^{(i)}_{\mathrm{train}}) := \ell(\boldsymbol{z}_{\mathrm{test}}; \widehat{\boldsymbol{\theta}}_{-i}) - \ell(\boldsymbol{z}_{\mathrm{test}}; \widehat{\boldsymbol{\theta}}). \tag{4}$$

Here, a greater value indicates that the loss on the test instance increases drastically by removing the i-th training instance from the training set. Thus, the i-th training instance is essential relative to predicting the test instance; therefore, it is highly relevant.

In practice, the following approximation is used to avoid computing $\widehat{\boldsymbol{\theta}}_{-i}$ explicitly.

$$R_{\mathrm{IF}}(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}^{(i)}_{\mathrm{train}}) \approx \langle \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\mathrm{test}}; \widehat{\boldsymbol{\theta}}), \boldsymbol{H}^{-1}\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}^{(i)}_{\mathrm{train}}; \widehat{\boldsymbol{\theta}})\rangle, \tag{5}$$

where $\boldsymbol{H}$ is the Hessian matrix of the loss $\mathcal{L}_{\mathrm{train}}$.

Relative IF (Barshan et al., 2020) Barshan et al. (2020) proposed to measure relevance according to "how largely the test loss will increase if the training instance is omitted from the training set under the constraint that the expected squared change in loss is sufficiently small"[3], which is the modified version of the influence function. Relative IF is computed as the cosine similarity of $\phi(\boldsymbol{z}) = \boldsymbol{H}^{-1/2}\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}; \widehat{\boldsymbol{\theta}})$:

$$R_{\mathrm{RIF}}(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}_{\mathrm{train}}) := \cos(\boldsymbol{H}^{-1/2}\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\mathrm{test}}; \widehat{\boldsymbol{\theta}}), \boldsymbol{H}^{-1/2}\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\mathrm{train}}; \widehat{\boldsymbol{\theta}})). \tag{6}$$

Fisher Kernel (Khanna et al., 2019) Khanna et al. (2019) proposed to measure the relevance of instances using the Fisher kernel as follows:

$$R_{\mathrm{FK}}(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}^{(i)}_{\mathrm{train}}) := \langle \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\mathrm{test}}; \widehat{\boldsymbol{\theta}}), \boldsymbol{I}^{-1}\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}^{(i)}_{\mathrm{train}}; \widehat{\boldsymbol{\theta}})\rangle, \tag{7}$$

where $\boldsymbol{I}$ is the Fisher information matrix of the loss $\mathcal{L}_{\mathrm{train}}$.

Grad-Dot, Grad-Cos (Perronnin et al., 2010; Yeh et al., 2018; Charpiat et al., 2019) Charpiat et al. (2019) proposed to measure relevance according to "how largely the loss will decrease when a small update is added to the model using the training instance." This can be computed as the dot product of the loss gradients, which we refer to as Grad-Dot.

$$R_{\mathrm{GD}}(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}_{\mathrm{train}}) := \langle \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\mathrm{test}}; \widehat{\boldsymbol{\theta}}), \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\mathrm{train}}; \widehat{\boldsymbol{\theta}})\rangle. \tag{8}$$

Note that a similar metric is studied by Yeh et al. (2018) as the *representer point value*.

As a modification of Grad-Dot, Charpiat et al. (2019) also proposed the following cosine version, which we refer to as Grad-Cos.

$$R_{\mathrm{GC}}(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}_{\mathrm{train}}) := \cos(\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\mathrm{test}}; \widehat{\boldsymbol{\theta}}), \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\mathrm{train}}; \widehat{\boldsymbol{\theta}})). \tag{9}$$

Note that the use of the cosine between the gradients is also proposed by Perronnin et al. (2010).

B EXPERIMENTAL SETUP

B.1 DATASETS AND MODELS

MNIST (LeCun et al., 1998) The MNIST dataset is used for handwritten digit image classification tasks. Here, input $\boldsymbol{x}$ is an image of a handwritten digit, and the output y consists of 10 classes ("0"

[3]This metric is called ℓ-RelatIF by Barshan et al. (2020)

to "9"). We adopted logistic regression and a CNN as the classification models. The CNN has six convolutional layers, and max-pooling layers for each two convolutional layers. The features obtained by these layers are fed into the global average pooling layer followed by a single linear layer. The number of the output channels of all the convolutional layers is set to 16. We trained the models using the Adam optimizer with a learning rate of 0.001. We used randomly sampled 5,500 training instances to train the models.

CIFAR10 (Krizhevsky, 2009) The CIFAR10 dataset is used for object recognition tasks. Here, input $\boldsymbol{x}$ is an image containing a certain object, and output y consists of 10 classes, e.g., "bird" or "airplane." Note that we used the same models as for the MNIST dataset. In addition, we adopted MobileNetV2 (Sandler et al., 2018) as a model with a higher performance than the previous model. We trained the models using the Adam optimizer with a learning rate of 0.001. In the experiments, we first pre-trained the models using all the training instances of CIFAR10, and then trained the models using randomly sampled 5,000 training instances. Without the pre-training, the classification performance of the models dropped significantly.

Note that we did not examine IF and FK on MobileNetV2 because the matrix inverse in these metrics required too much time to calculate even with the conjugate gradient approximation proposed by Koh & Liang (2017).

TREC (Li & Roth, 2002) The TREC dataset is used for question classification tasks. Here, input $\boldsymbol{x}$ is a question sentence, and output y is a question category consisting of six classes, e.g., "LOC" and "NUM." We used bag-of-words logistic regression and a two-layer Bi-LSTM as the classification models. In the Bi-LSTM, the last state is fed into one linear layer. The word embedding dimension is set to 16, and the dimension of the LSTM is set to 16 also. We trained the models using the Adam optimizer with a learning rate of 0.001. We used randomly sampled 2,726 training instances to train the models.

AGNews (Zhang et al., 2015) The AGNews dataset is used for news article classification tasks. Here, input $\boldsymbol{x}$ is a sentence, and output y is a category comprising four classes, e.g., "business" and "sports." We used the same models as TREC. We trained the models using the Adam optimizer with a learning rate of 0.001. We used randomly sampled 6,000 training instances to train the models.

Vehicle (Dua & Graff, 2017) The vehicle dataset is used for vehicle type classification tasks. Here, the input $\boldsymbol{x}$ consists of 18 features, and the output y is a type of vehicle comprising four classes, e.g., "bus" and "van." We used logistic regression and a three-layer MLP as the classification models. We trained the models using the Adam optimizer with a learning rate of 0.001. We used randomly sampled 423 training instances to train the models.

B.2 Computing Infrastructure

In our experiments, training of the models was run on a NVIDIA GTX 1080 GPU with Intel Xeon Silver 4112 CPU and 64GB RAM. Testing and computing relevance metrics were run on Xeon E5-2680 v2 CPU with 256GB RAM.

C Derivation of GC for Linear Logistic Regression

We consider linear logistic regression whose conditional distribution $p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$ is given by the y-th entry of $\sigma(W\boldsymbol{x})$, where σ is the softmax function, $\boldsymbol{\theta} = W \in \mathbb{R}^{C \times d}$, and C and d are the number of classes and the dimensionality of $\boldsymbol{x}$, respectively. Recall that the cross entropy loss for linear logistic regression is given as

$$\ell(\boldsymbol{z}; \boldsymbol{\theta}) = -\sum_{c=1}^{C} y_c \langle \boldsymbol{w}_c, \boldsymbol{x} \rangle + \log \sum_{c'=1}^{C} \exp(\langle \boldsymbol{w}_{c'}, \boldsymbol{x} \rangle), \tag{10}$$

where $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_C]^\top$. Let $\boldsymbol{e}_y$ be a vector whose y-th entry is one and zero otherwise. Then, the gradient of the loss with respect to $\boldsymbol{w}_c$ can be expressed as

$$\nabla_{\boldsymbol{w}_c} \ell(\boldsymbol{z}; \boldsymbol{\theta}) = (\sigma(W\boldsymbol{x}) - (\boldsymbol{e}_y)_c)\boldsymbol{x} = (\boldsymbol{r}^{\boldsymbol{z}})_c \boldsymbol{x}, \tag{11}$$

where $\boldsymbol{r}^{\boldsymbol{z}} = \sigma(W\boldsymbol{x}) - \boldsymbol{e}_y$is the residual for the prediction on $\boldsymbol{z}$. Hence, we have

$$\langle \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{z};\boldsymbol{\theta}), \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{z}';\boldsymbol{\theta}) \rangle = \sum_{c=1}^{C} \langle \nabla_{\boldsymbol{w}_c} \ell(\boldsymbol{z};\boldsymbol{\theta}), \nabla_{\boldsymbol{w}_c} \ell(\boldsymbol{z}';\boldsymbol{\theta}) \rangle \tag{12}$$

$$= \sum_{c=1}^{C} (\boldsymbol{r}^{\boldsymbol{z}})_c (\boldsymbol{r}^{\boldsymbol{z}'})_c \langle \boldsymbol{x}, \boldsymbol{x}' \rangle \tag{13}$$

$$= \langle \boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'} \rangle \langle \boldsymbol{x}, \boldsymbol{x}' \rangle, \tag{14}$$

which yields

$$R_{\mathrm{GC}}(\boldsymbol{z}, \boldsymbol{z}') = \frac{\langle \boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'} \rangle \langle \boldsymbol{x}, \boldsymbol{x}' \rangle}{\|\boldsymbol{r}^{\boldsymbol{z}}\| \|\boldsymbol{x}\| \|\boldsymbol{r}^{\boldsymbol{z}'}\| \|\boldsymbol{x}'\|} \tag{15}$$

$$= \cos(\boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'}) \cos(\boldsymbol{x}, \boldsymbol{x}'). \tag{16}$$

D Repairing Gradient-based Metrics

As described in Section 5, we found that training instances with extremely large norms were selected as relevant by IF, FK, and GD. Thus, to repair these metrics, we need to design metrics that can ignore instances with large norms. A simple yet effective way of repairing the metrics is to use ℓ_2 or cosine instead of the dot product. As Figure 4 shows, the ℓ_2 and cosine metrics performed better than the dot metrics. Indeed, the ℓ_2 metrics do not favor instances with large norms that lead to large ℓ_2-distance, and, through normalization, the cosine metrics completely ignore the effect of the norms

We name the repaired metrics of IF, FK, and GD based on the ℓ_2 metric as ℓ_2^{IF}, ℓ_2^{FK}, and ℓ_2^{GD}, respectively, and the repaired metrics based on the cosine metric as $\cos^{\mathrm{IF}}$ and $\cos^{\mathrm{FK}}$, and $\cos^{\mathrm{GD}}$, respectively[4]. We observed that these repaired metrics attained higher success rates on several evaluation criteria. The details of the results can be found in Appendix F.

E Do the models capture subclasses?

The identical subclass test requires the model to obtain internal representations that can distinguish subclasses. Here, we confirm that this condition is satisfied for all the datasets and models we used in the experiments. We consider that the model captures the subclasses if the latent representation $\boldsymbol{h}^{\mathrm{all}}$ has cluster structures. Figure 9 visualizes $\boldsymbol{h}^{\mathrm{all}}$ for each dataset and model using UMAP (McInnes et al., 2018). The figures show that the instances from different subclasses are not mixed completely random. MNIST and TREC have relatively clear cluster structures, while CIFAR10 and AGNews have vague clusters without explicit boundaries. These figures imply that the models capture subclases (although it may not be perfect).

F Complete Evaluation Results

F.1 Full Results

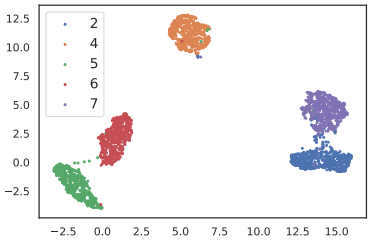
We show the complete results of the model randomization test in Table 2, the identical class test in Table 3, and the identical subclass test in Table 4. The results we present here are consistent with our observations in Section 4.

[4]Note that $\cos^{\mathrm{IF}}$ is the same as RIF and $\cos^{\mathrm{GD}}$ is the same as GC.

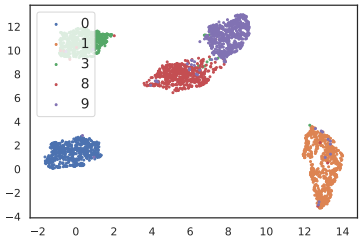
Table 2: Average Spearman rank correlation coefficients $\pm$ std. of each similarity function for model randomization test. The metrics prefixed with $\diamondsuit$ are the ones we have repaired. The results with the average score in the 95% confidence interval of the null distribution that the correlation is zero, which is [-0.088, 0.088], are colored.

	MNIST		CIFAR10		
Model	CNN	logreg	MobilenetV2	CNN	logreg
Parameter size	12K	8K	2.2M	12K	31K
Accuracy	0.98 ± 0.00	0.92 ± 0.00	0.89 ± 0.01	0.72 ± 0.02	0.35 ± 0.01
ℓ_2^x	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
ℓ_2^{last}	0.15 ± 0.01	-	0.07 ± 0.00	0.05 ± 0.01	-
ℓ_2^{all}	0.79 ± 0.00	-	0.02 ± 0.01	0.13 ± 0.01	-
$\cos^x$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
$\cos^{\text{last}}$	0.24 ± 0.00	-	0.07 ± 0.01	0.04 ± 0.01	-
$\cos^{\text{all}}$	0.78 ± 0.00	-	0.02 ± 0.01	0.09 ± 0.01	-
dot^x	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
dot^{last}	0.39 ± 0.01	-	0.05 ± 0.01	0.04 ± 0.01	-
dot^{all}	0.80 ± 0.00	-	-0.00 ± 0.01	0.12 ± 0.01	-
IF	0.05 ± 0.00	-0.00 ± 0.00	-0.05 ± 0.01	-0.04 ± 0.01	-0.04 ± 0.01
$\diamondsuit$ ℓ_2^{IF}	0.00 ± 0.02	-0.11 ± 0.00	0.01 ± 0.02	-0.03 ± 0.02	-0.05 ± 0.01
$\diamondsuit$ $\cos^{\text{IF}}$	0.02 ± 0.00	-0.05 ± 0.00	0.04 ± 0.01	0.03 ± 0.01	-0.03 ± 0.01
FK	-0.02 ± 0.01	0.02 ± 0.01	-0.02 ± 0.01	0.01 ± 0.01	-0.03 ± 0.01
$\diamondsuit$ ℓ_2^{FK}	-0.10 ± 0.04	0.05 ± 0.00	-0.16 ± 0.05	-0.12 ± 0.02	0.03 ± 0.01
$\diamondsuit$ $\cos^{\text{FK}}$	-0.00 ± 0.02	0.05 ± 0.01	-0.05 ± 0.01	-0.03 ± 0.01	-0.01 ± 0.01
GD	-0.08 ± 0.02	0.01 ± 0.01	-0.03 ± 0.01	-0.02 ± 0.01	0.04 ± 0.01
GC	-0.07 ± 0.03	-0.03 ± 0.01	-0.02 ± 0.02	0.01 ± 0.03	-0.05 ± 0.01
$\diamondsuit$ ℓ_2^{grad}	-0.09 ± 0.04	-0.13 ± 0.01	-0.09 ± 0.04	-0.07 ± 0.02	-0.06 ± 0.01

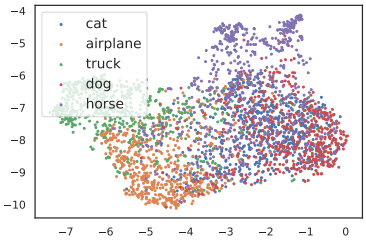
	TREC		AGNews		Vehicle	
Model	Bi-LSTM	logreg	Bi-LSTM	logreg	MLP	logreg
Parameter size	20K	7K	27K	9K	1K	76
Accuracy	0.86 ± 0.01	0.81 ± 0.02	0.80 ± 0.02	0.80 ± 0.01	0.77 ± 0.02	0.77 ± 0.01
ℓ_2^x	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
ℓ_2^{last}	0.19 ± 0.01	-	0.07 ± 0.01	-	0.16 ± 0.04	-
ℓ_2^{all}	0.25 ± 0.02	-	0.17 ± 0.01	-	0.20 ± 0.10	-
$\cos^x$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
$\cos^{\text{last}}$	0.17 ± 0.02	-	0.07 ± 0.01	-	-0.09 ± 0.18	-
$\cos^{\text{all}}$	0.26 ± 0.03	-	0.12 ± 0.02	-	-0.01 ± 0.13	-
dot^x	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
dot^{last}	0.25 ± 0.01	-	0.07 ± 0.01	-	0.85 ± 0.33	-
dot^{all}	0.26 ± 0.03	-	0.20 ± 0.01	-	0.97 ± 0.03	-
IF	0.01 ± 0.01	0.06 ± 0.01	0.03 ± 0.01	0.05 ± 0.01	-0.01 ± 0.03	-0.01 ± 0.02
$\diamondsuit$ ℓ_2^{IF}	-0.00 ± 0.02	-0.13 ± 0.02	-0.04 ± 0.02	-0.13 ± 0.00	-0.18 ± 0.24	-0.01 ± 0.28
$\diamondsuit$ $\cos^{\text{IF}}$	-0.01 ± 0.01	0.03 ± 0.01	0.02 ± 0.01	0.03 ± 0.01	-0.01 ± 0.03	-0.01 ± 0.05
FK	0.01 ± 0.00	0.03 ± 0.00	0.04 ± 0.01	0.03 ± 0.00	0.01 ± 0.06	0.02 ± 0.07
$\diamondsuit$ ℓ_2^{FK}	-0.14 ± 0.03	0.15 ± 0.01	-0.17 ± 0.04	0.14 ± 0.00	-0.18 ± 0.21	-0.01 ± 0.17
$\diamondsuit$ $\cos^{\text{FK}}$	-0.07 ± 0.02	-0.03 ± 0.00	-0.00 ± 0.03	-0.03 ± 0.00	0.08 ± 0.13	-0.04 ± 0.12
GD	-0.04 ± 0.01	-0.02 ± 0.02	-0.04 ± 0.01	0.03 ± 0.01	0.01 ± 0.11	-0.02 ± 0.05
GC	-0.04 ± 0.02	-0.01 ± 0.01	0.01 ± 0.02	0.04 ± 0.01	0.02 ± 0.13	-0.06 ± 0.11
$\diamondsuit$ ℓ_2^{grad}	-0.02 ± 0.02	-0.10 ± 0.02	-0.01 ± 0.02	-0.14 ± 0.00	-0.13 ± 0.21	0.11 ± 0.23



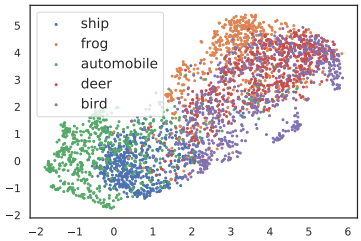
(a) MNIST with CNN. $y = \mathrm{A}$.



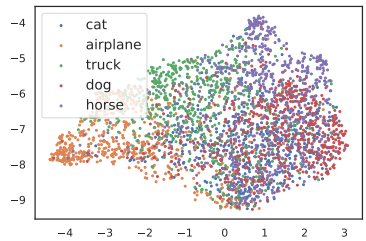
(b) MNIST with CNN. $y = \mathrm{B}$.



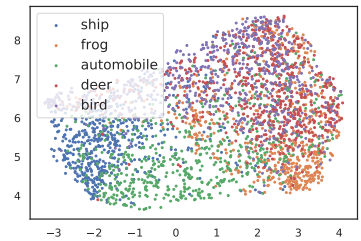
(c) CIFAR10 with MobileNetV2. $y = \mathrm{A}$.



(d) CIFAR10 with MobileNetV2. $y = \mathrm{B}$.



(e) CIFAR10 with CNN. $y = \mathrm{A}$.



(f) CIFAR10 with CNN. $y = \mathrm{B}$.

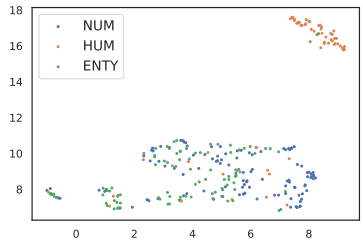
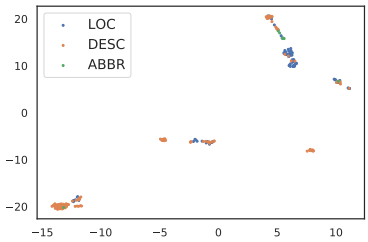
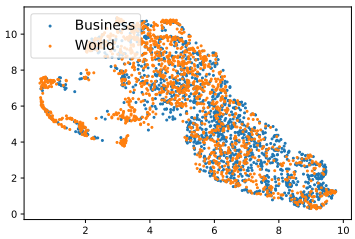


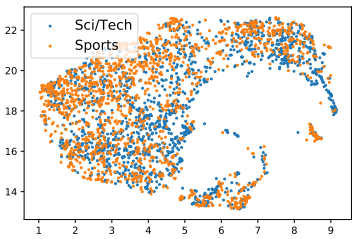
Figure 8: TREC with LSTM. $y = \mathrm{A}$.



(a) TREC with LSTM. $y = \mathrm{B}$.



(b) AGNews with LSTM. $y = \mathrm{A}$.



(c) AGNews with LSTM. $y = \mathrm{B}$.

Figure 9: visualization of $\boldsymbol{h}^{\mathrm{all}}$ in each dataset and model using UMAP.

Table 3: Average success rate ± std. of each relevancy metric for identical class test. The metrics prefixed with ♢ are the ones we have repaired. The results with the average success rate over 0.5 are colored.

	MNIST		CIFAR10		
Model	CNN	logreg	MobilenetV2	CNN	logreg
Parameter size	12K	8K	2.2M	12K	31K
Accuracy	0.98 ± 0.00	0.92 ± 0.00	0.89 ± 0.01	0.72 ± 0.02	0.35 ± 0.01
ℓ_2^x	0.93 ± 0.01	0.88 ± 0.01	0.26 ± 0.02	0.26 ± 0.02	0.24 ± 0.02
ℓ_2^{last}	0.99 ± 0.01	-	1.00 ± 0.00	0.75 ± 0.02	-
ℓ_2^{all}	0.98 ± 0.00	-	0.93 ± 0.02	0.61 ± 0.02	-
$\cos^x$	0.94 ± 0.01	0.88 ± 0.01	0.30 ± 0.03	0.29 ± 0.02	0.26 ± 0.02
$\cos^{\text{last}}$	0.99 ± 0.01	-	1.00 ± 0.00	0.78 ± 0.02	-
$\cos^{\text{all}}$	0.98 ± 0.00	-	0.97 ± 0.01	0.71 ± 0.02	-
dot^x	0.69 ± 0.02	0.68 ± 0.02	0.09 ± 0.02	0.10 ± 0.01	0.11 ± 0.02
dot^{last}	0.67 ± 0.02	-	1.00 ± 0.00	0.20 ± 0.02	-
dot^{all}	0.96 ± 0.01	-	0.96 ± 0.01	0.31 ± 0.01	-
IF	0.09 ± 0.01	0.26 ± 0.02	-	0.10 ± 0.01	0.09 ± 0.02
♢ ℓ_2^{IF}	0.72 ± 0.01	0.62 ± 0.02	-	0.14 ± 0.01	0.14 ± 0.01
♢ $\cos^{\text{IF}}$	0.82 ± 0.01	0.69 ± 0.02	-	0.12 ± 0.01	0.13 ± 0.02
FK	0.10 ± 0.01	0.21 ± 0.02	-	0.20 ± 0.02	0.20 ± 0.02
♢ ℓ_2^{FK}	0.77 ± 0.02	0.93 ± 0.01	-	0.82 ± 0.01	0.98 ± 0.00
♢ $\cos^{\text{FK}}$	0.92 ± 0.01	0.97 ± 0.01	-	0.93 ± 0.01	0.99 ± 0.00
GD	0.30 ± 0.01	0.87 ± 0.01	0.26 ± 0.03	0.71 ± 0.02	1.00 ± 0.00
GC	0.99 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	1.00 ± 0.00
♢ ℓ_2^{grad}	0.94 ± 0.01	0.99 ± 0.00	0.97 ± 0.01	0.99 ± 0.00	1.00 ± 0.00

	TREC		AGNews		Vehicle	
Model	Bi-LSTM	logreg	Bi-LSTM	logreg	MLP	logreg
Parameter size	20K	7K	27K	9K	1K	76
Accuracy	0.86 ± 0.01	0.81 ± 0.02	0.80 ± 0.02	0.80 ± 0.01	0.77 ± 0.02	0.77 ± 0.01
ℓ_2^x	0.70 ± 0.00	0.75 ± 0.00	0.39 ± 0.02	0.40 ± 0.02	0.65 ± 0.02	0.62 ± 0.02
ℓ_2^{last}	0.89 ± 0.00	-	0.84 ± 0.02	-	0.72 ± 0.03	-
ℓ_2^{all}	0.88 ± 0.00	-	0.84 ± 0.01	-	0.72 ± 0.03	-
$\cos^x$	0.73 ± 0.00	0.76 ± 0.00	0.47 ± 0.01	0.51 ± 0.02	0.66 ± 0.02	0.63 ± 0.02
$\cos^{\text{last}}$	0.89 ± 0.00	-	0.85 ± 0.01	-	0.74 ± 0.04	-
$\cos^{\text{all}}$	0.90 ± 0.00	-	0.84 ± 0.01	-	0.73 ± 0.04	-
dot^x	0.33 ± 0.00	0.34 ± 0.00	0.28 ± 0.02	0.47 ± 0.02	0.25 ± 0.00	0.27 ± 0.01
dot^{last}	0.93 ± 0.00	-	0.89 ± 0.01	-	0.26 ± 0.02	-
dot^{all}	0.93 ± 0.00	-	0.90 ± 0.01	-	0.27 ± 0.06	-
IF	0.29 ± 0.00	0.86 ± 0.00	0.24 ± 0.01	0.67 ± 0.02	0.39 ± 0.16	0.78 ± 0.08
♢ ℓ_2^{IF}	0.98 ± 0.00	0.95 ± 0.00	0.99 ± 0.00	0.92 ± 0.01	0.88 ± 0.06	0.95 ± 0.01
♢ $\cos^{\text{IF}}$	0.99 ± 0.00	0.96 ± 0.00	1.00 ± 0.00	0.97 ± 0.01	0.96 ± 0.02	0.99 ± 0.01
FK	0.28 ± 0.00	0.24 ± 0.00	0.32 ± 0.01	0.29 ± 0.03	0.31 ± 0.18	0.26 ± 0.17
♢ ℓ_2^{FK}	0.99 ± 0.00	0.96 ± 0.00	0.94 ± 0.01	0.68 ± 0.02	0.93 ± 0.04	0.94 ± 0.03
♢ $\cos^{\text{FK}}$	1.00 ± 0.00	0.96 ± 0.00	0.95 ± 0.01	0.84 ± 0.02	0.99 ± 0.01	0.99 ± 0.01
GD	0.49 ± 0.00	1.00 ± 0.00	0.76 ± 0.01	1.00 ± 0.00	0.90 ± 0.10	0.98 ± 0.02
GC	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
♢ ℓ_2^{grad}	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.01	1.00 ± 0.00

Table 4: Average success rate $\pm$ std. of each relevancy metric for identical subclass test. The metrics prefixed with $\diamondsuit$ are the ones we have repaired. The results with the average success rate over 0.5 are colored.

	MNIST		CIFAR10		
Model	CNN	logreg	MobilenetV2	CNN	logreg
Parameter size	12K	8K	2.2M	12K	31K
Accuracy	0.99 ± 0.00	0.88 ± 0.01	0.92 ± 0.01	0.84 ± 0.03	0.71 ± 0.03
ℓ_2^x	0.93 ± 0.01	0.96 ± 0.02	0.26 ± 0.02	0.29 ± 0.04	0.31 ± 0.03
ℓ_2^{last}	0.89 ± 0.02	-	0.29 ± 0.04	0.35 ± 0.04	-
ℓ_2^{all}	0.97 ± 0.01	-	0.49 ± 0.04	0.38 ± 0.03	-
$\cos^x$	0.95 ± 0.01	0.96 ± 0.02	0.29 ± 0.03	0.31 ± 0.04	0.31 ± 0.03
$\cos^{\text{last}}$	0.89 ± 0.02	-	0.32 ± 0.03	0.33 ± 0.03	-
$\cos^{\text{all}}$	0.98 ± 0.00	-	0.71 ± 0.04	0.50 ± 0.03	-
dot^x	0.70 ± 0.03	0.75 ± 0.03	0.09 ± 0.02	0.11 ± 0.03	0.09 ± 0.02
dot^{last}	0.24 ± 0.04	-	0.22 ± 0.02	0.20 ± 0.01	-
dot^{all}	0.94 ± 0.01	-	0.68 ± 0.03	0.25 ± 0.03	-
IF	0.12 ± 0.01	0.39 ± 0.05	-	0.06 ± 0.02	0.08 ± 0.02
$\diamondsuit$ ℓ_2^{IF}	0.62 ± 0.04	0.76 ± 0.03	-	0.17 ± 0.02	0.12 ± 0.02
$\diamondsuit$ $\cos^{\text{IF}}$	0.70 ± 0.02	0.87 ± 0.02	-	0.15 ± 0.02	0.09 ± 0.02
FK	0.19 ± 0.03	0.14 ± 0.02	-	0.11 ± 0.01	0.11 ± 0.02
$\diamondsuit$ ℓ_2^{FK}	0.81 ± 0.02	0.76 ± 0.03	-	0.31 ± 0.03	0.24 ± 0.02
$\diamondsuit$ $\cos^{\text{FK}}$	0.91 ± 0.02	0.85 ± 0.02	-	0.37 ± 0.03	0.23 ± 0.02
GD	0.42 ± 0.05	0.48 ± 0.03	0.20 ± 0.02	0.24 ± 0.03	0.21 ± 0.04
GC	0.97 ± 0.01	0.98 ± 0.01	0.54 ± 0.03	0.43 ± 0.04	0.39 ± 0.03
$\diamondsuit$ ℓ_2^{grad}	0.91 ± 0.02	0.95 ± 0.01	0.28 ± 0.03	0.38 ± 0.03	0.34 ± 0.03

	TREC		AGNews		Vehicle	
Model	Bi-LSTM	logreg	Bi-LSTM	logreg	MLP	logreg
Parameter size	20K	7K	27K	9K	1K	38
Accuracy	0.86 ± 0.01	0.81 ± 0.02	0.80 ± 0.02	0.80 ± 0.01	0.73 ± 0.02	0.73 ± 0.01
ℓ_2^x	0.78 ± 0.03	0.78 ± 0.02	0.40 ± 0.02	0.41 ± 0.01	0.67 ± 0.03	0.65 ± 0.02
ℓ_2^{last}	0.76 ± 0.02	-	0.53 ± 0.02	-	0.64 ± 0.05	-
ℓ_2^{all}	0.77 ± 0.03	-	0.58 ± 0.01	-	0.66 ± 0.04	-
$\cos^x$	0.82 ± 0.02	0.81 ± 0.02	0.49 ± 0.02	0.53 ± 0.02	0.68 ± 0.04	0.66 ± 0.03
$\cos^{\text{last}}$	0.75 ± 0.02	-	0.54 ± 0.01	-	0.68 ± 0.06	-
$\cos^{\text{all}}$	0.77 ± 0.02	-	0.59 ± 0.02	-	0.67 ± 0.03	-
dot^x	0.33 ± 0.03	0.34 ± 0.03	0.28 ± 0.02	0.48 ± 0.02	0.26 ± 0.03	0.26 ± 0.03
dot^{last}	0.40 ± 0.03	-	0.52 ± 0.02	-	0.27 ± 0.03	-
dot^{all}	0.59 ± 0.03	-	0.54 ± 0.02	-	0.28 ± 0.04	-
IF	0.31 ± 0.02	0.49 ± 0.03	0.25 ± 0.02	0.48 ± 0.01	0.34 ± 0.12	0.54 ± 0.09
$\diamondsuit$ ℓ_2^{IF}	0.68 ± 0.02	0.79 ± 0.02	0.56 ± 0.02	0.77 ± 0.02	0.76 ± 0.14	0.86 ± 0.06
$\diamondsuit$ $\cos^{\text{IF}}$	0.72 ± 0.01	0.75 ± 0.03	0.56 ± 0.02	0.80 ± 0.02	0.86 ± 0.09	0.91 ± 0.08
FK	0.30 ± 0.02	0.16 ± 0.02	0.28 ± 0.01	0.25 ± 0.02	0.16 ± 0.08	0.20 ± 0.05
$\diamondsuit$ ℓ_2^{FK}	0.73 ± 0.03	0.78 ± 0.02	0.56 ± 0.02	0.63 ± 0.02	0.73 ± 0.13	0.62 ± 0.09
$\diamondsuit$ $\cos^{\text{FK}}$	0.81 ± 0.02	0.79 ± 0.01	0.61 ± 0.02	0.73 ± 0.02	0.80 ± 0.06	0.67 ± 0.10
GD	0.45 ± 0.02	0.60 ± 0.02	0.50 ± 0.02	0.54 ± 0.02	0.47 ± 0.09	0.43 ± 0.03
GC	0.81 ± 0.01	0.87 ± 0.02	0.65 ± 0.02	0.72 ± 0.02	0.82 ± 0.06	0.83 ± 0.07
$\diamondsuit$ ℓ_2^{grad}	0.78 ± 0.02	0.88 ± 0.02	0.61 ± 0.02	0.73 ± 0.03	0.72 ± 0.10	0.75 ± 0.09

F.2 ADDITIONAL RESULTS

The identical class test require the most relevant instance to be of the same class as the test instance. In practice, users can be more confident about a model's output if several instances are provided as evidence. In other words, we expect that the most relevant and a first few relevant instances will be of the same class. This observation leads to the additional criterion, which is a generalization of the identical class test.

Definition 5 (Top-k Identical Class Test)**.** For $\boldsymbol{z}_{\text{test}} = (\boldsymbol{x}_{\text{test}}, \widehat{y}_{\text{test}})$, let $\bar{\boldsymbol{z}}^j = (\bar{\boldsymbol{x}}^j, \bar{y}^j)$ be a training instance with the j-th largest relevance score. Then, we require $\bar{y}^j = \widehat{y}_{\text{test}}$ for any $j \in \{1, 2, \ldots, k\}$.

This observation also applies to identical subclass test, which leads to the following criterion

Definition 6 (Top-k Identical Subclass Test)**.** For $\boldsymbol{z}_{\text{test}} = (\boldsymbol{x}_{\text{test}}, \widehat{y}_{\text{test}})$, let $\bar{\boldsymbol{z}}^j = (\bar{\boldsymbol{x}}^j, \bar{y}^j)$ be a training instance with the j-th largest relevance score. Then, we require $s(\bar{\boldsymbol{z}}^j) = s(\widehat{\boldsymbol{z}}_{\text{test}}),\ \forall j \in \{1, 2, \ldots, k\}$.

We show the results of the top-10 identical class test in Table 3, and the top-10 identical subclass test in Table 4.

Table 5: Average success rate $\pm$ std. of each relevancy metric for top-10 identical class test. The metrics prefixed with $\diamondsuit$ are the ones we have repaired. The results with the average success rate over 0.5 are colored.

	MNIST		CIFAR10		
Model	CNN	logreg	MobilenetV2	CNN	logreg
Parameter size	12K	8K	2.2M	12K	31K
Accuracy	0.98 ± 0.00	0.92 ± 0.00	0.89 ± 0.01	0.72 ± 0.02	0.35 ± 0.01
ℓ_2^x	0.63 ± 0.02	0.63 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
ℓ_2^{last}	0.95 ± 0.01	-	0.98 ± 0.01	0.30 ± 0.01	-
ℓ_2^{all}	0.89 ± 0.01	-	0.64 ± 0.05	0.14 ± 0.01	-
$\cos^x$	0.67 ± 0.02	0.65 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
$\cos^{\text{last}}$	0.97 ± 0.01	-	0.98 ± 0.01	0.33 ± 0.02	-
$\cos^{\text{all}}$	0.92 ± 0.00	-	0.84 ± 0.03	0.23 ± 0.02	-
dot^x	0.19 ± 0.01	0.20 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
dot^{last}	0.42 ± 0.03	-	0.98 ± 0.01	0.04 ± 0.01	-
dot^{all}	0.88 ± 0.01	-	0.79 ± 0.03	0.05 ± 0.01	-
IF	0.00 ± 0.00	0.00 ± 0.00	-	0.00 ± 0.00	0.00 ± 0.00
$\diamondsuit$ ℓ_2^{IF}	0.25 ± 0.01	0.10 ± 0.01	-	0.00 ± 0.00	0.00 ± 0.00
$\diamondsuit$ $\cos^{\text{IF}}$	0.59 ± 0.02	0.17 ± 0.01	-	0.00 ± 0.00	0.00 ± 0.00
FK	0.00 ± 0.00	0.02 ± 0.01	-	0.00 ± 0.00	0.06 ± 0.01
$\diamondsuit$ ℓ_2^{FK}	0.23 ± 0.03	0.65 ± 0.02	-	0.25 ± 0.02	0.87 ± 0.01
$\diamondsuit$ $\cos^{\text{FK}}$	0.59 ± 0.01	0.82 ± 0.02	-	0.54 ± 0.02	0.93 ± 0.01
GD	0.00 ± 0.00	0.41 ± 0.02	0.00 ± 0.00	0.15 ± 0.01	1.00 ± 0.00
GC	0.95 ± 0.01	0.99 ± 0.01	0.92 ± 0.02	0.92 ± 0.01	1.00 ± 0.00
$\diamondsuit$ ℓ_2^{grad}	0.57 ± 0.02	0.95 ± 0.01	0.78 ± 0.03	0.80 ± 0.01	0.99 ± 0.00

	TREC		AGNews		Vehicle	
Model	Bi-LSTM	logreg	Bi-LSTM	logreg	MLP	logreg
Parameter size	20K	7K	27K	9K	1K	76
Accuracy	0.86 ± 0.01	0.81 ± 0.02	0.80 ± 0.02	0.80 ± 0.01	0.77 ± 0.02	0.77 ± 0.01
ℓ_2^x	0.23 ± 0.00	0.23 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.09 ± 0.02	0.09 ± 0.02
ℓ_2^{last}	0.68 ± 0.00	-	0.48 ± 0.03	-	0.19 ± 0.07	-
ℓ_2^{all}	0.66 ± 0.00	-	0.46 ± 0.01	-	0.16 ± 0.06	-
$\cos^x$	0.24 ± 0.00	0.24 ± 0.00	0.01 ± 0.00	0.02 ± 0.01	0.10 ± 0.02	0.10 ± 0.01
$\cos^{\text{last}}$	0.69 ± 0.00	-	0.51 ± 0.03	-	0.22 ± 0.07	-
$\cos^{\text{all}}$	0.68 ± 0.00	-	0.48 ± 0.02	-	0.17 ± 0.06	-
dot^x	0.05 ± 0.00	0.05 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.15 ± 0.12	0.16 ± 0.13
dot^{last}	0.75 ± 0.00	-	0.64 ± 0.03	-	0.13 ± 0.11	-
dot^{all}	0.84 ± 0.00	-	0.66 ± 0.03	-	0.15 ± 0.11	-
IF	0.00 ± 0.00	0.24 ± 0.00	0.00 ± 0.00	0.02 ± 0.01	0.01 ± 0.01	0.10 ± 0.03
$\diamondsuit$ ℓ_2^{IF}	0.83 ± 0.00	0.47 ± 0.00	0.94 ± 0.01	0.20 ± 0.02	0.25 ± 0.13	0.47 ± 0.05
$\diamondsuit$ $\cos^{\text{IF}}$	0.91 ± 0.00	0.65 ± 0.00	0.97 ± 0.01	0.48 ± 0.01	0.42 ± 0.12	0.61 ± 0.03
FK	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.05 ± 0.11	0.08 ± 0.11
$\diamondsuit$ ℓ_2^{FK}	0.90 ± 0.00	0.71 ± 0.00	0.61 ± 0.02	0.06 ± 0.01	0.55 ± 0.19	0.64 ± 0.12
$\diamondsuit$ $\cos^{\text{FK}}$	0.95 ± 0.00	0.77 ± 0.00	0.71 ± 0.03	0.15 ± 0.01	0.85 ± 0.06	0.85 ± 0.08
GD	0.11 ± 0.00	0.99 ± 0.00	0.55 ± 0.02	0.98 ± 0.01	0.56 ± 0.19	0.70 ± 0.05
GC	0.96 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.95 ± 0.04	1.00 ± 0.00
$\diamondsuit$ ℓ_2^{grad}	0.94 ± 0.00	1.00 ± 0.00	0.99 ± 0.01	0.98 ± 0.00	0.81 ± 0.09	0.95 ± 0.03

Table 6: Average success rate $\pm$ std. of each relevancy metric for top-10 identical subclass test. The metrics prefixed with $\diamondsuit$ are the ones we have repaired. The results with the average success rate over 0.5 are colored.

	MNIST		CIFAR10		
Model	CNN	logreg	MobilenetV2	CNN	logreg
Parameter size	12K	8K	2.2M	12K	31K
Accuracy	0.99 ± 0.00	0.88 ± 0.01	0.92 ± 0.01	0.84 ± 0.03	0.71 ± 0.03
ℓ_2^x	0.64 ± 0.02	0.71 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
ℓ_2^{last}	0.54 ± 0.04	-	0.00 ± 0.00	0.00 ± 0.00	-
ℓ_2^{all}	0.85 ± 0.02	-	0.08 ± 0.02	0.01 ± 0.00	-
$\cos^x$	0.67 ± 0.02	0.74 ± 0.03	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00
$\cos^{\text{last}}$	0.57 ± 0.05	-	0.00 ± 0.00	0.00 ± 0.00	-
$\cos^{\text{all}}$	0.89 ± 0.02	-	0.16 ± 0.02	0.02 ± 0.01	-
dot^x	0.21 ± 0.02	0.23 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
dot^{last}	0.08 ± 0.02	-	0.00 ± 0.00	0.00 ± 0.00	-
dot^{all}	0.79 ± 0.03	-	0.13 ± 0.02	0.01 ± 0.01	-
IF	0.01 ± 0.01	0.00 ± 0.00	-	0.00 ± 0.00	0.00 ± 0.00
$\diamondsuit$ ℓ_2^{IF}	0.14 ± 0.03	0.16 ± 0.02	-	0.00 ± 0.00	0.00 ± 0.00
$\diamondsuit$ $\cos^{\text{IF}}$	0.37 ± 0.02	0.35 ± 0.04	-	0.00 ± 0.00	0.00 ± 0.00
FK	0.00 ± 0.00	0.00 ± 0.00	-	0.00 ± 0.00	0.00 ± 0.00
$\diamondsuit$ ℓ_2^{FK}	0.22 ± 0.02	0.30 ± 0.03	-	0.00 ± 0.00	0.00 ± 0.00
$\diamondsuit$ $\cos^{\text{FK}}$	0.58 ± 0.02	0.46 ± 0.04	-	0.00 ± 0.00	0.00 ± 0.00
GD	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00
GC	0.86 ± 0.03	0.87 ± 0.02	0.06 ± 0.02	0.01 ± 0.01	0.01 ± 0.01
$\diamondsuit$ ℓ_2^{grad}	0.50 ± 0.03	0.69 ± 0.04	0.02 ± 0.01	0.00 ± 0.00	0.00 ± 0.00

	TREC		AGNews		Vehicle	
Model	Bi-LSTM	logreg	Bi-LSTM	logreg	MLP	logreg
Parameter size	20K	7K	27K	9K	1K	38
Accuracy	0.86 ± 0.01	0.81 ± 0.02	0.80 ± 0.02	0.80 ± 0.01	0.73 ± 0.02	0.73 ± 0.01
ℓ_2^x	0.27 ± 0.05	0.25 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.10 ± 0.00	0.09 ± 0.00
ℓ_2^{last}	0.30 ± 0.02	-	0.01 ± 0.00	-	0.07 ± 0.00	-
ℓ_2^{all}	0.34 ± 0.02	-	0.01 ± 0.01	-	0.08 ± 0.00	-
$\cos^x$	0.28 ± 0.04	0.27 ± 0.02	0.01 ± 0.00	0.02 ± 0.01	0.10 ± 0.00	0.09 ± 0.00
$\cos^{\text{last}}$	0.30 ± 0.02	-	0.01 ± 0.00	-	0.06 ± 0.00	-
$\cos^{\text{all}}$	0.34 ± 0.02	-	0.02 ± 0.01	-	0.10 ± 0.00	-
dot^x	0.05 ± 0.01	0.05 ± 0.01	0.01 ± 0.00	0.02 ± 0.01	0.00 ± 0.00	0.00 ± 0.00
dot^{last}	0.14 ± 0.01	-	0.01 ± 0.00	-	0.00 ± 0.00	-
dot^{all}	0.17 ± 0.02	-	0.02 ± 0.00	-	0.00 ± 0.00	-
IF	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.00	0.10 ± 0.00
$\diamondsuit$ ℓ_2^{IF}	0.11 ± 0.02	0.24 ± 0.02	0.01 ± 0.00	0.10 ± 0.01	0.02 ± 0.00	0.20 ± 0.00
$\diamondsuit$ $\cos^{\text{IF}}$	0.22 ± 0.03	0.25 ± 0.02	0.01 ± 0.00	0.20 ± 0.02	0.02 ± 0.00	0.38 ± 0.00
FK	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.00	0.00 ± 0.00
$\diamondsuit$ ℓ_2^{FK}	0.28 ± 0.04	0.26 ± 0.02	0.01 ± 0.00	0.02 ± 0.01	0.18 ± 0.00	0.02 ± 0.00
$\diamondsuit$ $\cos^{\text{FK}}$	0.41 ± 0.03	0.25 ± 0.02	0.01 ± 0.00	0.09 ± 0.01	0.10 ± 0.00	0.01 ± 0.00
GD	0.10 ± 0.02	0.01 ± 0.00	0.01 ± 0.00	0.06 ± 0.01	0.39 ± 0.00	0.39 ± 0.00
GC	0.37 ± 0.03	0.37 ± 0.02	0.04 ± 0.00	0.10 ± 0.01	0.16 ± 0.00	0.30 ± 0.00
$\diamondsuit$ ℓ_2^{grad}	0.24 ± 0.03	0.34 ± 0.02	0.02 ± 0.01	0.08 ± 0.01	0.37 ± 0.00	0.30 ± 0.00