Learning to Double-check Model Prediction from a Causal Perspective

Xun Deng*, Fuli Feng*, Xiang Wang, Xiangnan He, Hanwang Zhang, Tat-Seng Chua

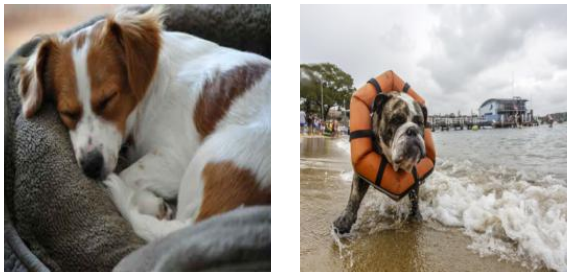
***Abstract*—The present machine learning schema typically uses a one-pass model inference (*e.g.,* forward propagation) to make predictions in the testing phase. It is inherently different from human students who double-check the answer during examinations especially when the confidence is low. To bridge this gap, we propose a Learning to Double-check (L2D) framework, which formulates double-check as a learnable procedure with two core operations: recognizing unreliable predictions and revising predictions. To judge the correctness of a prediction, we resort to counterfactual faithfulness in causal theory and design a contrastive faithfulness measure. In particular, L2D generates counterfactual features by imagining: "*what would the sample features be if its label was the predicted class*"; and judges the prediction by the faithfulness of the counterfactual features. Furthermore, we design a simple and effective revision module to revise the original model prediction according to the faithfulness. We apply the L2D framework to three classification models and conduct experiments on two public datasets for image classification, validating the effectiveness of L2D in prediction correctness judgment and revision.**

***Index Terms*—Classification, Double-check, Counterfactual Faithfulness, Causality.**

I. INTRODUCTION

Machine learning models are widely used in various real-world applications such as machine translation [1], image recognition [2], and recommender system [3]. In practice, the model is typically trained offline and deployed to serve the samples coming during the testing period. That is, the model indiscriminately makes predictions for all testing samples, while they can differ a lot. For instance, some samples (Figure 1(b)) can be hard to make confident predictions. Apparently, it differs from the behavior of human students in the testing period (*e.g.,* an examination), who would double-check the answer for hard questions. Due to the lack of double-check, the current models encounter sharp performance drops on low confidence samples [4], [5].

Existing methods attempt to bridge the gap mainly by *post-processing* [6] the model prediction with heuristic strategies. For instance, the ensemble methods [7] revise the prediction according to the consensus across models. However, hard samples in practice can result in low confidence on different models, limiting the effect of ensemble. Hand-craft rules based on domain knowledge are also used for checking model prediction [8]. However, the strategies are mainly hand-crafted in an application-specific manner, which are hard to generalize.



(a) Prob. of *dog*: *0.99* (b) Prob. of *dog*: *0.17*

Fig. 1: Examples of a normal sample (a) and hard sample (b) in the class of *dog* and the corresponding model predictions.

This work aims to build a uniform framework to achieve double-check. Our belief is that double-check is a learnable ability like making predictions where the key lies in identifying unreliable predictions. We hypothesize the source of unreliable predictions as misrecognizing feature patterns. In this light, double-check follows an opposite direction, which starts at assuming the prediction (*e.g., dog*) is correct, and then reversely imagines the representative features of the class. By testing the consensus between the imagination and the fact, we can introspect the appropriateness of the assumption and estimate the reliability of the corresponding prediction. In this light, the key to learning double-check lies in the modeling of imagination and consensus testing.

We resort to causal theory to model double-check. Apparently, we can express the imagination as a counterfactual statement, *e.g.,* "*given the factual feature (Figure 1(b)), what it would be if its class was cat?*" We thus formulate the imagination operation as a counterfactual inference $P\left(X_{Y=y} | X=\boldsymbol{x}, Y=\bar{y}\right)$ where y is the assumed class, $\boldsymbol{x}$ and $\bar{y}$ are factual features and label, respectively. According to counterfactual faithfulness [9] and consistency rule [10], the counterfactual features should be close to the factual features when the assumed class is indeed the label, *i.e.,* $y = \bar{y}$. Therefore, we measure the faithfulness for each candidate class and estimate the reliability of model prediction (*e.g.,* y) as the relative faithfulness against the other candidates, which is termed *contrastive faithfulness measure*. Undoubtedly, we can achieve learnable double-check once designing proper neural network modules to achieve the counterfactual inference and faithfulness measure.

Towards this end, we propose a Learning to Double-check

X. Deng, F. Feng, X. Wang, X. He are with University of Science and Technology of China (dx981228@mail.ustc.edu.cn, fulifeng93@gmail.com, xiangwang@u.nus.edu, hexn@ustc.edu.cn). H. Zhang is with Nanyang Technological University (hangwangzhang@ntu.edu.sg). TS. Chua is with the National University of Singapore (dcscts@nus.edu.sg). Xun Deng and Fuli Feng contribute equally. Fuli Feng is the corresponding author. This work is supported by the National Key R&D Program of China (2020YFB1406703), the National Natural Science Foundation of China (62272437, U21B2026), and the CCCD Key Lab of Ministry of Culture and Tourism.

framework, which consists of two main modules: 1) *counterfactual inference* (CI) module; and 2) *consensus measure* (CM) module. In particular, the CI module is a generative network to infer the counterfactual feature conditioned on the fact and the prediction according to a partial causal graph of the classification problem. The CM module is a siamese network to measure the consensus of factual and counterfactual features, which is trained by a common triplet retrieval objective [11]. Furthermore, we pursue revising the original model prediction according to the consensus when necessary. Toward this goal, we further design a *revision module* for the L2D framework, which is a simple yet effective convolutional network trained as a normal classifier. We apply the L2D framework on three representative image classification models: ResNet [12], RSC [13], and DSL [14] and conduct extensive experiments on two real-world image classification datasets. Empirical results validate the effectiveness of L2D in classification mistake identification and revision, achieving significant improvement (from 6.6% to 9.8%) over vanilla models.

The main contributions are summarized as follows:

- We highlight the importance of double-check for model prediction and propose a *Learning to Double-check* framework, which can check and revise the prediction.
- We propose a new *contrastive faithfulness measure*, which reveals the correctness of a prediction according to the faithfulness of counterfactual features.
- We conduct extensive experiments on two image classification datasets, validating the effectiveness of the L2D framework.

II. Methods

A. Formulation of L2D

As an initial attempt for learning to double-check, we narrow down the scope to image classification models and adopt a general C-way classification setting, which aims to learn a function $\boldsymbol{y} = f(\boldsymbol{x}|\boldsymbol{\theta})$. $\boldsymbol{x} \in \mathbb{R}^D$ denotes the feature of a sample. $\boldsymbol{y} \in \mathbb{R}^C$ is a distribution over the classes. The class with the maximum probability, $y = \arg\max_i \ \boldsymbol{y}_i, i \in [1, C]$, is the prediction of sample $\boldsymbol{x}$. Assume there is a classification model trained over labeled data $\mathcal{T} = \{(\boldsymbol{x}, \bar{y})\}$. Formally,

$$\hat{\boldsymbol{\theta}} = \min_{\boldsymbol{\theta}} \sum_{(\boldsymbol{x},\bar{y})\in\mathcal{T}} l(\bar{y}, f(\boldsymbol{x}|\boldsymbol{\theta})) + \lambda\|\boldsymbol{\theta}\|, \quad (1)$$

where $\hat{\boldsymbol{\theta}}$ is the learned parameters of the model, $l(\cdot)$ is a classification loss such as cross-entropy [15], and λ is a hyper-parameter to adjust the regularization. Double-check is a procedure to: 1) *reveal the correctness of a prediction*; and 2) *revise the prediction if necessary* (see Figure 2).

We model the prediction correctness judgment as two-steps of *counterfactual inference*, which generates the counterfactual feature for the input sample under an assumed class; and *contrastive faithfulness measurement*, which is based on the consensus between counterfactual and factual features. According to Pearl's expression of counterfactual [16], we first provide a formal definition of counterfactual features.

Definition 1 (Counterfactual Features): Given a sample $(\boldsymbol{x}, \bar{y})$, the counterfactual features for assuming the label to be y are: $\boldsymbol{x}_y^* = E\left(X_{Y=y}|X = X_{Y=\bar{y}} = \boldsymbol{x}, Y = \bar{y}\right)$, where X and Y are random variables; and $E(\cdot)$ denotes the expectation. $X_{Y=\bar{y}}$ denotes the value of X conditioned on the event $Y = \bar{y}$.

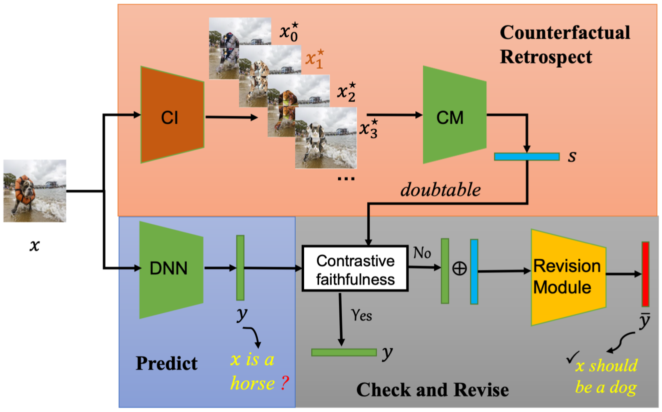


Fig. 2: Illustration of L2D framework.

Note that $P(X_{Y=y}|X = X_{Y=\bar{y}} = \boldsymbol{x}, Y = \bar{y})$ is the distribution of counterfactual features, which is inherently different from the normal conditional probability distribution $P(X|Y = y)$. This is because $X = \boldsymbol{x}$ and $Y = \bar{y}$ are given known facts in the counterfactual distribution.

Accordingly, we construct a set of counterfactual samples $\{\boldsymbol{x}_{y'}^* | y' \in [1, C]\}$ for the input factual sample $\boldsymbol{x}$ by successively assuming the label as each candidate class. Upon the counterfactual samples, we define a contrastive faithfulness measure of the model prediction y.

Definition 2 (Contrastive Faithfulness): Given the model prediction y on sample $\boldsymbol{x}$, the contrastive faithfulness is:

$$z_y = \begin{cases} 1, \textit{iff } y = \arg\max\limits_{y'} \ s(\boldsymbol{x}, \boldsymbol{x}_{y'}^*), y' \in [1, C], \\ 0, \ \textit{otherwise}, \end{cases} \quad (2)$$

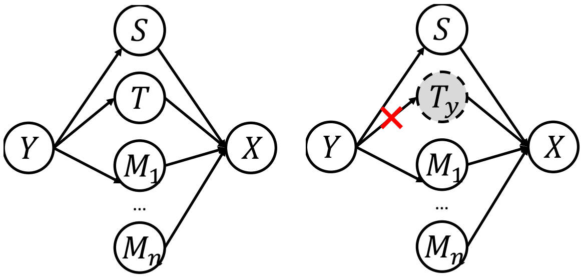
where $s(\cdot)$ is a consensus measure between factual and counterfactual features.

We treat z_y as a correctness measurement according to the consistency rule in [10]. $z_y = 0$ indicates that the prediction y on sample $\boldsymbol{x}$ is unreliable due to the existence of a candidate class whose counterfactual features achieve higher consensus with the factual features than the predicted class y. On the contrary, $z_y = 1$ indicates a reliable prediction. In the following, we term predictions with $z_y = 0$ and $z_y = 1$ as *unfaithful predictions* and *faithful predictions*, respectively. Obviously, to realize learnable double-check, we have to learn:

- $P\left(X_{Y=y}|X = X_{Y=\bar{y}} = \boldsymbol{x}, Y = \bar{y}\right)$, the counterfactual distribution to infer counterfactual features $E\left(X_{Y=y}|X = X_{Y=\bar{y}} = \boldsymbol{x}, Y = \bar{y}\right)$.
- $s(\boldsymbol{x}, \boldsymbol{x}_y^*)$, a function to estimate feature consensus.

B. Estimating Counterfactual Features

Apparently, we cannot directly infer $E\left(X_{Y=y}|X = X_{Y=\bar{y}} = \boldsymbol{x}, Y = \bar{y}\right)$ for three reasons: 1) The probability $P\left(X_{Y=y}|X = X_{Y=\bar{y}} = \boldsymbol{x}, Y = \bar{y}\right)$ is unidentifiable since the label of testing samples (*i.e.*, $Y = \bar{y}$) are not available. 2) Such causal inference requires the whole



(a) Causal Graph (b) Counterfactual

Fig. 3: The causal graph for: (a) the factual feature generation procedure; (b) a counterfactual with hypothetical condition $T = T_y$. As studying feature generation, we adopt the anticausal setting [17], [18] where X is the outcome.

causal graph that describes the generation procedure of the feature $\boldsymbol{x}$. In most practical cases, such causal graph is not available and hard to be constructed due to the large amount of mediators between the variables Y and X (Figure 3(a)). The mediators can be understood as concepts used to define the classes. For instance, in the object recognition task, the mediator can be the shape (S) and the texture (T) of the object. 3) The hypothetical condition $Y = y$ is too strong, which can thoroughly change the feature, resulting in unrealistic generation [10].

In this light, we relax the hypothetical condition to be $T = T_y$, which intervenes only one mediator (Figure 3(b)). $T = T_y$ denotes the value of T under the condition $Y = y$. There are various candidate mediators available in image classification such as the texture of the object and the shape of the object. In this study, we select T as the texture of the object. Accordingly, the counterfactual statement becomes: "*What the feature would be if its texture was T_y*, where all the other variables remain their factual values (*e.g.,* $S = S_{\bar{y}} = \bar{s}$). Formally, the counterfactual feature becomes: $\boldsymbol{x}^*_y = E\left(X_{T=T_y} | X = X_{Y=\bar{y}} = \boldsymbol{x}, T = T_{\bar{y}} = \bar{t}\right)$, where $\bar{t}$ denotes the factual value of T. Apparently, this counterfactual is identifiable as long as $T_{Y=y}$ and $T_{X=\boldsymbol{x}}$ are identifiable, which can be easily inferred through texture extraction tools [19].

Counterfactual inference module. Generative networks have become an emerging technique to model the inference of counterfactual features [3], [10], [19]. Among the existing models, we find that Counterfactual Generative Network (CGN) [19] can well support our requirement, which accounts for the mediator between X and Y and consists of components to model $T_{Y=y}$ and $T_{X=\boldsymbol{x}}$. We thus directly use CGN as the CI module in the L2D framework to generate the counterfactual samples for all candidate classes: $\{\boldsymbol{x}^*_{y'} | y' \in [1, C]\}$.

C. Learning $s(\boldsymbol{x}, \boldsymbol{x}^*_{y'})$

Functionally speaking, $s(\boldsymbol{x}, \boldsymbol{x}^*_{y'})$ aims to distinguish faithful counterfactual samples from unfaithful ones. Accordingly, we formulate an image retrieval problem to learn the function. In particular, we define: 1) the factual sample $\boldsymbol{x}$ as a query; 2) the counterfactual sample $\boldsymbol{x}^*_{y'}$ with $y' = \bar{y}$ as a positive counterfactual sample; and 3) the one with $y' \neq \bar{y}$ as a negative counterfactual sample. Therefore, we model the consensus measure module as an image retrieval model, which is formulated as $s(\boldsymbol{x}^*_{y'}, \boldsymbol{x} | \boldsymbol{\eta})$ where $\boldsymbol{\eta}$ denotes model parameters.

Consensus measure module. Inspired by the success of siamese network [20] in image retrieval tasks [20], we devise the CM module as a siamese network, which adopts the same structure as the prediction model $f(\boldsymbol{x}|\boldsymbol{\theta})$ being double-checked. In particular, we estimate the consensus through the *cosine*-similarity of the latent representation of $\boldsymbol{x}$ and $\boldsymbol{x}^*_{y'}$. As $f(\cdot)$ and $s(\cdot)$ serve for different targets, we additionally optimize the parameters of $s(\cdot)$ (*i.e.,* $\boldsymbol{\eta}$). In particular, we minimize a widely used triplet retrieval loss [11] over the counterfactual samples of the training data. Formally,

$$\hat{\boldsymbol{\eta}} = \min_{\boldsymbol{\eta}} \sum_{(\boldsymbol{x},\bar{y})\in\mathcal{T}} \sum_{y'\in[1,C]\&y'\neq\bar{y}} \max\left(0, s(\boldsymbol{x}, \boldsymbol{x}^*_{y'}) - s(\boldsymbol{x}, \boldsymbol{x}^*_{\bar{y}}) + \alpha\right), \tag{3}$$

where α is a hyperparameter of to what extent the positive and negative counterfactual samples should be separated.

D. Revision Module

Based on the CI and CM modules, the L2D framework can judge the correctness of a prediction $(\boldsymbol{x}, y)$ by calculating its contrastive faithfulness z_y. Considering that we humans further call for a revision once identify a wrong answer, we further devise a revision module. Our belief is that the level of feature consensus provides clues for adjusting the original prediction distribution $\boldsymbol{y}$. In this light, we formulate the revision module as: $r(\boldsymbol{y}, \boldsymbol{s}|\boldsymbol{\omega})$, where $\boldsymbol{s} \in \mathbf{R}^C$ includes the feature consensus of all counterfactual samples $\{\boldsymbol{x}^*_{y'} | y' \in [1, C]\}$, *i.e.,* the *cosine*-similarity given by the CM module.

Module design. Considering the success of Convolutional Neural Network (CNN) in recognizing local-region patterns, we devise the module as a CNN, which consists of a stack layer, a convolution layer, and two fully-connected layers.

- *Stack layer.* The layer stacks the original distribution and similarity scores as a matrix $\boldsymbol{Y} = [\boldsymbol{y}, \boldsymbol{s}] \in \mathcal{R}^{C\times 2}$, which can facilitate observing the local-region patterns.
- *Convolution layer.* The layer consists of 1D vertical filters to distill patterns within the classification distribution and similarity scores, which is formulated as: $\boldsymbol{C} =< \boldsymbol{F}, \boldsymbol{Y} >$. $\boldsymbol{F} \in \mathcal{R}^{C\times K}$ denotes the filters of the layer and $\boldsymbol{C} \in \mathcal{R}^{K\times 2}$ denotes the recognized signals. K is the number of filters.
- *FC layers.* The FC layers learn strategies to combine the probability and similarity and perform revision, where the output is normalized through softmax.

Training. We optimize the parameters of the revision module by minimizing the classification loss over the factual training samples, which is:

$$\hat{\boldsymbol{\omega}} = \min_{\boldsymbol{\omega}} \sum_{(\boldsymbol{x},\bar{y})\in\mathcal{T}} l(\bar{y}, r(\boldsymbol{y}, \boldsymbol{s}|\boldsymbol{\omega})) + \beta\|\boldsymbol{\omega}\|, \tag{4}$$

where β denotes the hyper-parameter to adjust the weight of the regularization term. In practice, most of the training samples obtain faithful predictions (*i.e.,* $z_y = 1$). To balance the occurrence of faithful and unfaithful predictions, we perform down-sampling on the training data.

Machine learning schema with L2D. To summarize, the final L2D framework consists of three modules: CI module, CM module, and revision module, to recognize the unfaithful predictions and perform revision. To incorporate the L2D framework into the present machine learning schema, we revise both the training and testing procedures. Algorithm 1 illustrates the new schema with L2D.

Algorithm 1 Learning schema with L2D.

```
Input: Training data $\mathcal{T}$.
        /* Training */
Train base model (optimize Eq. (1));
Train CGN;
Train CM module (optimize Eq. (3);
Train revision module (optimize Eq. (4);
Return $\hat{\boldsymbol{\theta}}$, CGN, $\hat{\boldsymbol{\eta}}$, and $\hat{\boldsymbol{\omega}}$.
        /* Testing */
Infer $\boldsymbol{y} = f(\boldsymbol{x}|\hat{\boldsymbol{\theta}})$;
for $y' = 1 \rightarrow C$ do
    Infer $\boldsymbol{x}^*_{y'} =$ CGN$(\boldsymbol{x}, y')$;
end for
Calculate $z_y$ (Eq. (2);
$z_y = 1$ ? return $y$ : return $r(\boldsymbol{y}, \boldsymbol{s}|\hat{\boldsymbol{\omega}})$;
```

III. Experiments

We aim to answer the following research questions:

- **RQ1:** How effective is our L2D framework in distinguishing wrong and correct predictions?
- **RQ2:** How effective is our L2D framework in amending the predictions?
- **RQ3**: What revision patterns are uncovered? In what cases our L2D framework performs as expected or fails?

A. Experimental Settings

Datasets. We perform experiments on the Animal and Vehicle datasets in NICO, for 10-way and 8-way classification, respectively [21]. Following the OOD[1] data split setting in [14], [21], we split these datasets by restricting the number of contexts (*e.g.,* beach and sky) that appear in the training set. Specifically, for each class, we randomly select five contexts appearing in the training set, while the rest five contexts are in the testing set. The discrepancy in contexts will result in more hard samples. In fact, a sharp drop on classification accuracy exists between the validation and testing sets: around 15% on Animal and 10% on Vehicle, which offers evidence of the distribution shift between the training and testing sets. The number of images in the training/validation/testing set is 5318/1088/2524 for Animal and 4332/885/2073 for Vehicle. We tune hyperparameters on the validation set and report the average testing accuracy of five different runs.

Counterfactual generation. Recall that we use CGN [19] to generate counterfactual samples for each training and testing samples (*cf.* Section II-B) by intervening the texture of the object (*i.e.,* T represents texture). In particular, given a sample $\boldsymbol{x}$, we generate a counterfactual image $\boldsymbol{x}^*_{y'}$ by feeding CGN with the common textual of class y' where we enumerate all possible classes y'. Considering that CGN is not a perfect generative network, which shows high cognitive uncertainty, we run four repeats for each $\boldsymbol{x}^*_{y'}$. As to the consensus measurement $s(\boldsymbol{x}, \boldsymbol{x}^*_{y'})$, we use the mean of the four repeats to mitigate the impact of cognitive uncertainty.

Baselines. To demonstrate the effectiveness of the L2D framework, we compare it with five baselines covering the fields of counterfactual data augmentation, stable learning, and domain generalization. Among these baselines, three typical models are equipped with L2D. We adopt ResNet-18 [12] as the backbone model and initialize each model with the weights pretrained on ImageNet[2] [22].

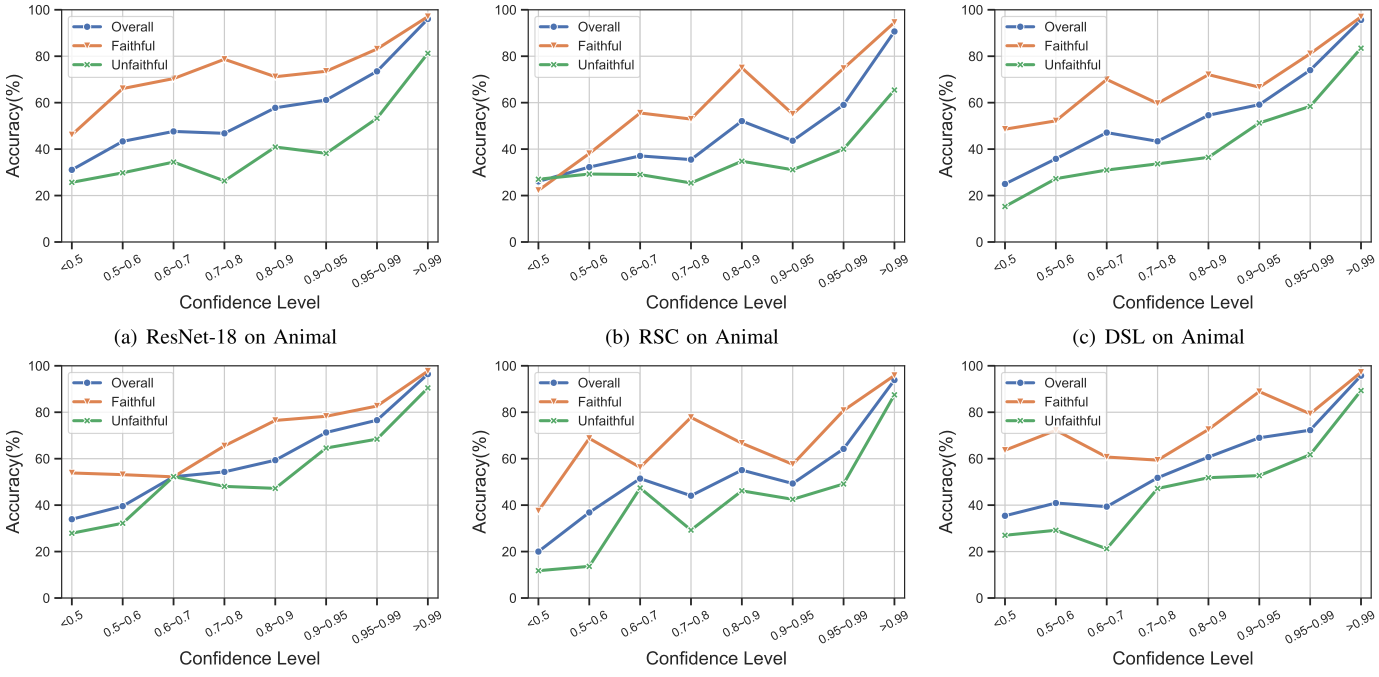
- **ResNet-18** [12]: ResNet-18 is widely used as the backbone in image classification.
- **CNBB** [21]: The prior study [23] proposes a causally regularized logistic regression (CRLR) for OOD image classification. However, as CRLR requires access to all training samples during each iteration, it is not feasible for CNN-based models. To resolve the limitations of CRLR, a very recent work [21] devises a new weight learning model, ConvNet with Batch Balancing (CNBB), which balances the confounder distribution within a minibatch.
- **DSL** [14]: To improve the generalization ability of deep neural networks (DNN), DSL learns independent features for different images via weight learning in function space of Random Fourier Features. It makes the DNN models concentrate more on the objects in the image.
- **CGN** [19]: CGN is a data augmentation method, which generates various counterfactual samples with the imagined texture and background. Following the rules [19], we generate the same amount of counterfactual samples as the training set to train models.
- **RSC** [13]: RSC belongs to the line of adversarial dropout. It improves the robustness of CNN by focusing on the subset of representation with smaller gradients and mutes the rest during the backpropagation. The model is forced to pay attention to more features of the target object after training.

Parameter Settings. We detail the parameter settings of classification models and our L2D:

- **Classification Models.** Adam optimizer is adopted to train the models and the learning rate starts from 0.001 and is decayed by 0.1 after 14 epochs. For each model, we set the batch size as 128 and epoch number as 20. There are some hyperparameters specific to each model, which we set as the value suggested in the original paper for fair comparison. For both Animal and Vehicle datasets, we use two widely used data augmentation strategies: 1) randomly cropping

[1] The reason for adopting OOD testing is that the revision of prediction is more necessary for OOD cases.

[2] We adopt this initialization for fair comparison since the CGN used in L2D is trained on ImageNet [22].



(a) ResNet-18 on Animal

(b) RSC on Animal

(c) DSL on Animal

(d) ResNet-18 on Vehicle

(e) RSC on Vehicle

(f) DSL on Vehicle

Fig. 4: Model's original classification accuracy at each CL level and the accuracy on faithful samples and unfaithful samples.

the images with random retain ratio in [0.8, 1.0]; and 2) randomly applied horizontal flipping with 50% probability.

- **L2D**[3]**.** The CM module is initialized with the parameters of the well-trained baseline model. We further train the CM module for 200 iterations by optimizing Eq. 3 with $\alpha = 0.4$, the learning rate of 0.0001 and batch size of 64. Meanwhile, we train the revision module for 10 epochs, where the learning rate is 0.01 and the batch size is 32. As severe overfitting issue will occur in revision module training, we adopt an aggressive early stopping rule: stopping once the accuracy decreases on the validation set.

B. *Performance on Prediction Correctness Measure (RQ1)*

We first investigate how L2D performs in terms of prediction correctness judgment, *i.e.,* whether the contrastive faithfulness measure z_y can recognize the wrong predictions. To this end, we set eight Confidence Levels (CL) based on the range of maximum class probability (MCP) [4] given by the original classification model. In particular, CL1: MCP < 0.5; CL2: $0.5 <$ MCP < 0.6; CL3: $0.6 <$ MCP < 0.7; CL4: $0.7 <$ MCP < 0.8; CL5: $0.8 <$ MCP < 0.9; CL6: $0.9 <$ MCP < 0.95; CL7: $0.95 <$ MCP < 0.99; CL8: MCP > 0.99. For each model, we can split the testing samples into eight groups according to their prediction probability. The samples in each group are identified as faithful samples, if their contrastive faithfulness is 1, otherwise as unfaithful samples. In each group, we calculate the *overall accuracy*, *faithful accuracy* for the faithful samples, and *unfaithful accuracy* for the unfaithful samples. The group-wise results of models are shown in Figure 4. We have the following observations based on these results:

- In most cases, the original accuracy exhibits an increasing trend from CL 1 to 8, which means the vanilla models have inferior performance on low confidence samples. This observation is consistent with previous work [4], where we view samples in low CL groups as hard samples of the model.
- Across all eight CL groups, the faithful accuracy is consistently better than the overall accuracy. Meanwhile, the gap between faithful accuracy and unfaithful accuracy is around 40% in Animal and 30% in Vehicle. It means that a prediction has a much higher chance to be correct if it is faithful (*i.e.,* $z_y = 1$). The significant gap validates the effectiveness of the proposed contrastive faithfulness measure in recognizing wrong predictions regardless of model confidence level.
- In most cases, unfaithful accuracy is less than 30%. Even for the CL interval where $0.9 <$ MCP < 0.99, the unfaithful accuracy is still less than 60% in both datasets. This result reveals the potential for revising unfaithful model predictions, which can thus validate the rationality of equipping the L2D framework with a revision module.

TABLE I: Accuracy(%) of CM module on separating positive and negative counterfactual samples.

	Animal			Vehicle		
	ResNet-18	RSC	DSL	ResNet-18	RSC	DSL
Training	96.52	96.84	96.50	87.82	89.40	89.00
Testing	90.26	90.32	89.02	86.06	86.16	86.48

Performance of CM module. Beyond the final contrastive faithfulness, we further evaluate the specific feature consensus given by the CM module (*cf.* Section II-C). We randomly sample 5,000 triplets of (*query factual sample*, *positive counterfactual sample*, *negative counterfactual sample*) from the training and testing samples to demonstrate the performance of the CM module. We report the accuracy over the triplets with a binary correctness criterion that the positive counterfactual sample receives higher feature consensus. Recall that we ini-

[3] Source code: https://github.com/xiangtanshi/L2D.

TABLE II: Overall classification accuracy of all compared methods on Animal and Vehicle.

Method	ResNet-18	CNBB	RSC	CGN	DSL
Animal	75.04	74.48	78.26	75.90	74.61
+L2D	**76.47**$_{+1.43}$	—	**79.32**$_{+1.06}$	—	**77.10**$_{+2.49}$
Vehicle	83.99	84.01	85.32	84.44	83.26
+L2D	**84.50**$_{+0.51}$	—	**85.88**$_{+0.56}$	—	**84.21**$_{+0.95}$

tialize the CM module with the original classification model. Table I shows the training and testing accuracy of CM modules based on ResNet-18, DSL, and RSC. Generally speaking, the CM module achieves great accuracy around 90%, which means the CM module can accurately judge the faithfulness of counterfactual samples. Remarkably, the accuracy of the CM module on the testing set is only slightly lower than that on the training set. Considering the sharp distribution shift in the testing set, we postulate that the generated counterfactual features are stable and informative for distinguishing between the positive and negative counterfactual samples.

- As a brief summary, we validate the rationality of judging model prediction correctness based on counterfactual faithfulness and the effectiveness of the CI and CR modules in our L2D framework.

C. Performance on Prediction Revision (RQ2)

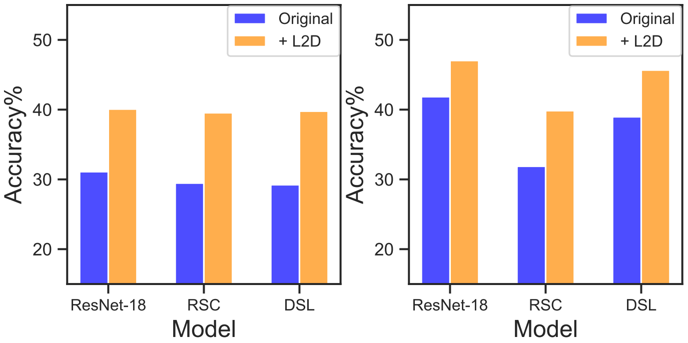
We then investigate the effects of the whole L2D framework with revision module *w.r.t.* image classification performance.

Overall performance. Table II shows the image classification performance of all compared methods on Animal and Vehicle, where we apply L2D on ResNet-18, DSL, and RSC. Obviously, leveraging our L2D framework achieves consistent improvements over the vanilla models (ResNet-18, DSL, RSC) across all cases in Table II, indicating the rationality and effectiveness of our L2D framework. Remarkably, L2D also achieves stable improvement on the highly competitive RSC model, which leverages adversarial learning to distill stable features. We attribute the gain harvested by applying L2D to the retrospection ability, which double-checks and adjusts model predictions during inference. Furthermore, we test the performance of directly utilizing counterfactual consensus for classification by judging the prediction as the class with the largest consensus measure. It encounters performance drops of 5% and 15% on the Animal and Vehicle datasets. The result indicates that retrospection is not a simple operation, which thus requires a revision module.

Effects on hard samples. To further explore the characteristics of L2D, we take a close look at the testing set by exploring hard samples whose contrastive faithfulness is 0 and its MCP is less than 0.9. Figure 5 summarizes the performance and comparison between different models. From Figure 5, we find that applying L2D achieves significant improvements over the vanilla models by 9.8% and 6.6% in Animal and Vehicle, respectively. We attribute these improvements to our consideration of contrastive faithfulness, which better captures the consensus among features and endows the models with powerful discrimination ability. Besides, the revision module indeed learn some effective double-check strategies.

TABLE III: Performance of vanilla models and applying L2D on the perturbed test of Animal and Vehicle.

Method	ResNet-18	RSC	DSL
Animal	72.58	76.15	72.34
+L2D	**75.16**$_{+2.58}$	**77.50**$_{+1.35}$	**75.35**$_{+3.01}$
Vehicle	82.58	83.84	82.00
+L2D	**83.72**$_{+1.14}$	**84.61**$_{+0.89}$	**83.44**$_{+1.44}$



(a) On Animal (b) On Vehicle

Fig. 5: Classification performance on hard samples and improvement when applying L2D.

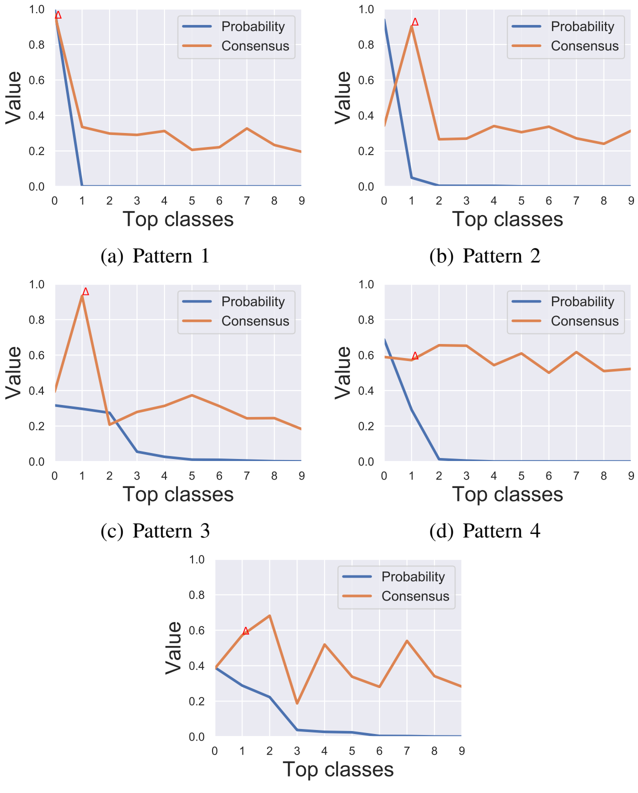
Study on robustness. We then study the robustness of L2D against invisible perturbations. In particular, instead of directly downscaling the testing images to (224,224), we add an additional interpolation by resizing to (256,256) before to (224,224). Table III shows the performance on the Animal dataset. Through a cross comparison with Table II, we can find that the original models, ResNet-18, RSC, and DSL, encounters a performance drop of about 2.5% due to the impact of perturbations. Applying L2D achieves performance gain by a large margin, which validates the robustness of the CM module and the contrastive faithfulness measure.

D. In-depth Analysis (RQ3)

To investigate the working mechanism of the proposed L2D framework, we visualize the patterns learned by the revision module and conduct a case study over the testing samples.

1) Revision Patterns: We first reveal how the revision module works by mining revision *patterns* on representative inputs. Recall that the inputs of the revision module are the classification probabilities $\boldsymbol{y}$ and the feature consensus values across all counterfactual samples $\boldsymbol{s}$. Taking DSL+L2D as an example, we depict five patterns in Figure 6. Note that we index the candidate classes according to a descending order of predicted probabilities. In particular,

- **Pattern 1**, where both curves exhibit peaks at the top 1 class. In this case, the revision module remains the original classification.
- **Pattern 2**, where both curves also show high peaks, but the peak of consensus is at class with the second highest probability. Here, the revision module chooses the second candidate class.
- **Pattern 3**, where the consensus curve is similar to Pattern 2, but the probability curve shows a plateau. Similarly, the revision module decides to revise the prediction.



(a) Pattern 1

(b) Pattern 2

(c) Pattern 3

(d) Pattern 4

(e) Pattern 5

Fig. 6: Pattern 1 to 5 presents five ways that the revision module used to revise the prediction. The horizontal axis is the candidate classes ranked according to their probability and the triangle in each pattern denotes the revised prediction.

- **Pattern 4**. There is a low peak on the probability curve and overall high plateau on the similarity score, the output is top 2 class.
- **pattern 5**. There is a plateau on the probability curve, and the low peak of the similarity score is at the top 3 or lower classes, the output is still the second candidate class.

We then count the testing samples under each pattern. Generally, Pattern 1, 2, and 3 mostly lead to correct revisions. Pattern 4 is a bad pattern, where the original prediction is typically correct. This pattern occurs mostly in images where the shape is not correctly given by CGN. Consequently, only few pixels of the object are detected, making the CI module fail in intervening texture. The revision in Pattern 5 is also not very accurate since the true class is beyond the top 2 candidates given by the original model. It reflects that the revision module cannot precisely revise the prediction for some very hard cases. According to the patterns, we think incorporating more signals such as the uncertainty of probability and consensus will further improve the revision performance, which is left for future exploration.

By further analyzing the wrong predictions missed by the revision module, we find that most of them have very high MCP. Consequently, the probabilities of the remaining classes are all very low, making the revision module hardly to recognize the correct class. The issue is caused by the softmax normalization, which only cares about the absolute difference between each logit, but is insensitive to the mean of logits[4]. A potential future direction for learning to double-check is pursuing reasonable probability estimations for low confidence candidate classes. Furthermore, Figure 6 shows that the classification probability exhibits highly skewed distribution. As such, L2D can only evaluate the consensus of top predicted classes for acceleration, especially when facing a large number of candidate classes.

TABLE IV: Performance comparison on the test set of Animal between vanilla models, applying L2D with revision module and applying L2D with SVM classifier.

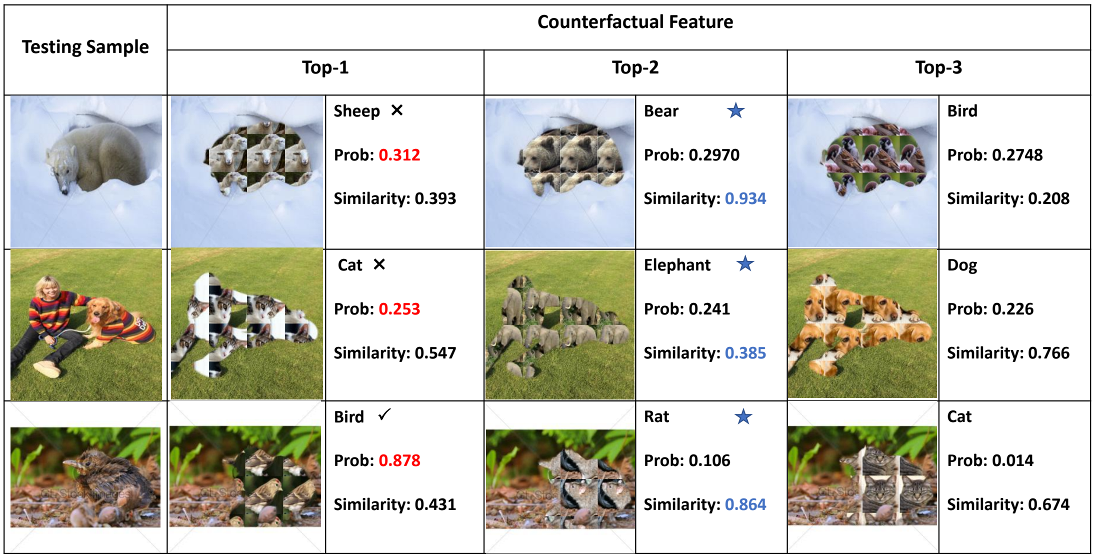
Method	ResNet-18	RSC	DSL
Original Model	75.04	78.26	74.61
+L2D,revision module	**76.47**+1.43	**79.32**+1.06	**77.10**+2.49
+L2D,SVM	**75.35**+0.31	**78.26**+0	**75.31**+0.7

As these revision patterns are clear, we further investigate whether shallow revision module can perform retrospection. In particular, we evaluate a variant of L2D by replacing its revision module with an SVM classifier. The result is shown in Table IV. We can find that the SVM revision module also achieves performance gain in some cases, which indicates its capability of capturing some revision patterns. However, there is a clear gap between the performance of L2D with SVM and the proposed revision module. Shallow models may fail to make good use of the information in feature consensus.

2) Case Study: Figure 7 shows three testing samples; their predictions from DSL; and corresponding counterfactual samples and revisions from L2D.

- **Case 1: Correcting false prediction.** There is a polar bear lying on the ice. The vanilla model has never seen such images in the training set but it has witnessed a lot of images of white sheep. With high probability, the model is biased towards *sheep*. However, this image exhibits the core features (*e.g.,* face shape and eyes) of the bear. Hence, our CM module is able to give a quite high consensus score (0.934) to counterfactual features with *bear* texture as opposed to sheep (0.393). This accords with Pattern 2 (*cf.* Figure 6(b)) and the revision module makes correct revision. It is clear that L2D frees the model from the influence of spurious correlations between white fur and sheep.
- **Case 2: Revising wrong prediction to a new class.** This is absolutely a hard sample for the classification model as dogs are rarely seen wearing clothes and staying with humans in the Animal dataset. It is quite interesting that the model views *elephant* as the second possible candidate class. The main reason stems from the bias in the training set that most elephants appear on green grass fields or forests and stand by people. We hypothesize that the model was confounded by the background when predicting this case. Whereas, by imagining and comparing the texture of elephants with that of cats and dogs, the CM module finds that the third indeed identifies dog as the most faithful candidate. Nevertheless, the revision module labels the image as elephant, which accords to Pattern 5 (*cf.* Figure 6(e)).
- **Case 3: Making mistakes on revision.** On this case, the revision module changes the prediction from *bird* to *rat*. The

[4] Note that logits x is the same as x+100 and x-100 in view of softmax.



Testing Sample	Counterfactual Feature					
	Top-1		Top-2		Top-3	
		Sheep ✗ Prob: 0.312 Similarity: 0.393		Bear ★ Prob: 0.2970 Similarity: 0.934		Bird Prob: 0.2748 Similarity: 0.208
		Cat ✗ Prob: 0.253 Similarity: 0.547		Elephant ★ Prob: 0.241 Similarity: 0.385		Dog Prob: 0.226 Similarity: 0.766
		Bird ✓ Prob: 0.878 Similarity: 0.431		Rat ★ Prob: 0.106 Similarity: 0.864		Cat Prob: 0.014 Similarity: 0.674

Fig. 7: Illustration of three representative cases. Row 1 to 3 corresponds to cases 1 to 3. The first image in each row is the testing sample, the left 3 images are counterfactual features of top 3 possible classes as predicted by the DSL model. $\times/\surd$ marks the correstness of model prediction; $\star$ indicates the revised prediction given by the revision model. The highest probability and consensus are highlighted with red and blue colors, respectively.

main cause is that the generated counterfactual samples lack qualified texture.

To summarize, we postulate that the advantage of L2D comes from the mitigating of sample selection bias (the 1_{st} case) and confounding bias (the 2_{nd} case). It can also fail on cases (the 3_{rd}) where the intervened mediator is not informative.

IV. RELATED WORK

A. Counterfactual Thinking

Counterfactual Sample. In the field of vision and natural language processing, a line of research recently concentrates on generating counterfactual samples to augment the training data. This technique has been widely adopted in language understanding-related tasks, such as SA [24], NLI [25], question answering [26], dialogue system [27], and vision-language navigation [28]. Instead of masking objects in images [29] or modifying words in questions [25], another line generates counterfactual samples by adding label information with the help of generative networks. CGN [19] disentangles the components of an image into three independent mechanisms that are decided by the class label. Yue et al. [10] generate counterfactual images by decoding the combination of image and label features. Unlike our work which calculates the consensus for every class and utilizes the result for correction, CGN [19] simply adds these samples to the training set and GCM-CF [10] provides binary information about seen/unseen of an image, but do not interfere the inference.

Counterfactual Training. Beyond data augmentation under the standard supervised learning paradigm, a research line incorporates counterfactual samples into other learning paradigms like adversarial training [27]–[29] and contrastive learning [30]. This is orthogonal to the line that incorporates counterfactual samples into the decision-making procedure of model inference. CRM [31] accounts for counterfactual samples as additional clues for making classification. However, CRM requires manually constructed counterfactual samples in the model training stage, which cannot be applied to most classification tasks. Applying CRM will cost much more manual resources than the proposed L2D.

Counterfactual Inference. There are some prior studies [32], [33] incorporating counterfactual inference into the testing phase of models. They rely on the causal diagram to perform counterfactual inference, which requires insights into the specific tasks.

B. Hard sample and Revision.

Hard Sample. Much attention has been paid on generating hard samples to boost training. For the classification task, some work improves the robustness of models by feeding hard samples with special perturbations [34] in the training phase. In addition, adversarial training [35] steps further by forcing the model to fight against perturbations or attacks. This line differs from ours, since we aim to find out hard samples for the model with the help of contrastive faithfulness and then correct model predictions. Another line resorts to additional checks on the raw prediction in the inference stage, such as posterior regularization [36].

Confidence and Revision. Convolutional neural network is sensitive to small perturbations added on the input image [37]. The softmax output sometimes could not give us clue about the certainty of CNN's prediction [38]. Multiple solutions such as Histogram binning, Platt scaling, Matrix, and vector scaling [38] have been proposed to calibrate the softmax output. Our CM module provides a new way to check the confidence of softmax value, and the correction made by the revision module is consistently effective for different models on multiple datasets.

V. CONCLUSION

In this paper, we highlighted the importance of double-check in the testing phase of machine learning model. We resorted to causal theory to model the double-check procedure with a contrastive faithfulness measure. In this light, we proposed a Learning to Double-check framework, which is seamlessly incorporated into the present machine learning schema. We instantiated it on three image classification models and conducted extensive experiments on two datasets. The results justify that the L2D framework can accurately recognize and revise wrong predictions.

This work opens up a new research direction about the model inference stage, which is of great practical value. As an initial attempt, this work focuses on the image classification problem. In the future, we will extend the L2D framework to other classification tasks such as text classification; and broader settings such as regression and ranking. The L2D framework assumes that the intervened mediator T is not confounded. To address this issue, we will improve the CI module to account for confounders. Moreover, we will explore more potential signals for the revision module.

REFERENCES

[1] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma, "Dual learning for machine translation," *NeurIPS*, vol. 29, pp. 820–828, 2016.

[2] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *ICML*, 2020, pp. 3247–3258.

[3] H. Zou, P. Cui, B. Li, Z. Shen, J. Ma, H. Yang, and Y. He, "Counterfactual prediction for bundle treatment," in *NeurIPS*, vol. 33, 2020.

[4] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez, "Addressing failure prediction by learning model confidence," in *NeurIPS 2019*. Curran Associates, Inc., 2019, pp. 2898–2909.

[5] S. Depeweg, J.-M. Hernandez-Lobato, F. Doshi-Velez, and S. Udluft, "Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning," in *ICML*, 2018, pp. 1184–1193.

[6] S. Vannitsem, D. S. Wilks, and J. Messner, *Statistical postprocessing of ensemble forecasts*. Elsevier, 2018.

[7] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[8] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing, "Harnessing deep neural networks with logic rules," in *ACL*, 2016, pp. 2410–2420.

[9] M. Besserve, A. Mehrjou, R. Sun, and B. Schölkopf, "Counterfactuals uncover the modular structure of deep generative models," in *ICLR*, 2019.

[10] Z. Yue, T. Wang, H. Zhang, Q. Sun, and X.-S. Hua, "Counterfactual zero-shot and open-set visual recognition," in *CVPR*, 2021.

[11] J.-J. Chen, C.-W. Ngo, F.-L. Feng, and T.-S. Chua, "Deep understanding of cooking procedure for cross-modal recipe retrieval," in *ACMMM*, 2018, pp. 1020–1028.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[13] Z. Huang, H. Wang, E. Xing, and D. Huang, "Self-challenging improves cross-domain generalization," in *ECCV*, 2020.

[14] X. Zhang, P. Cui, R. Xu, L. Zhou, Y. He, and Z. Shen, "Deep stable learning for out-of-distribution generalization," in *CVPR*, 2021.

[15] S. Kullback, *Information theory and statistics*. Courier Corporation, 1997.

[16] J. Pearl, *Causality*. Cambridge university press, 2009.

[17] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij, "Semi-supervised learning in causal and anticausal settings," in *Empirical Inference*. Springer, 2013, pp. 129–141.

[18] J. Kügelgen, A. Mey, M. Loog, and B. Schölkopf, "Semi-supervised learning, causality, and the conditional cluster assumption," in *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 1–10.

[19] A. G. Axel Sauer, "Counterfactual generative networks," in *ICLR*, 2021.

[20] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *ICPR*. IEEE, 2016, pp. 378–383.

[21] Y. He, Z. Shen, and P. Cui, "Towards non-iid image classification: A dataset and baselines," *Pattern Recognition*, vol. 110, p. 107383, 2021.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[23] Z. Shen, P. Cui, K. Kuang, B. Li, and P. Chen, "Causally regularized learning with agnostic data selection bias," in *ACM MM*, 2018, pp. 411–419.

[24] L. Yang, E. Kenny, T. L. J. Ng, Y. Yang, B. Smyth, and R. Dong, "Generating plausible counterfactual explanations for deep transformers in financial text classification," in *COLING*, 2020, pp. 6150–6160.

[25] D. Kaushik, E. Hovy, and Z. Lipton, "Learning the difference that makes a difference with counterfactually-augmented data," in *ICLR*, 2019.

[26] L. Chen, X. Yan, J. Xiao, H. Zhang, S. Pu, and Y. Zhuang, "Counterfactual samples synthesizing for robust visual question answering," in *CVPR*, 2020, pp. 10 800–10 809.

[27] Q. Zhu, W. Zhang, T. Liu, and W. Y. Wang, "Counterfactual off-policy training for neural dialogue generation," in *EMNLP*, 2020, pp. 3438–3448.

[28] T.-J. Fu, X. E. Wang, M. F. Peterson, S. T. Grafton, M. P. Eckstein, and W. Y. Wang, "Counterfactual vision-and-language navigation via adversarial path sampler," in *ECCV*. Springer, 2020, pp. 71–86.

[29] D. Teney, E. Abbasnedjad, and A. van den Hengel, "Learning what makes a difference from counterfactual examples and gradient supervision," in *ECCV*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 580–599.

[30] Z. Liang, W. Jiang, H. Hu, and J. Zhu, "Learning to contrast the counterfactual samples for robust visual question answering," in *EMNLP*, 2020, pp. 3285–3292.

[31] F. Feng, J. Zhang, X. He, H. Zhang, and T.-S. Chua, "Empowering language understanding with counterfactual reasoning," *arXiv preprint arXiv:2106.03046*, 2021.

[32] Y. Niu, K. Tang, H. Zhang, Z. Lu, X.-S. Hua, and J.-R. Wen, "Counterfactual vqa: A cause-effect look at language bias," in *CVPR*, 2021.

[33] K. Tang, J. Huang, and H. Zhang, "Long-tailed classification by keeping the good and removing the bad momentum causal effect," in *NeurIPS*, vol. 33, 2020.

[34] E. Wong and J. Z. Kolter, "Learning perturbation sets for robust machine learning," *arXiv preprint arXiv:2007.08450*, 2020.

[35] D. Khashabi, T. Khot, and A. Sabharwal, "More bang for your buck: Natural perturbation for robust question answering," in *EMNLP*, 2020, pp. 163–170.

[36] S. Srivastava, I. Labutov, and T. Mitchell, "Zero-shot learning of classifiers from natural language quantification," in *ACL(Volume 1: Long Papers)*, 2018, pp. 306–316.

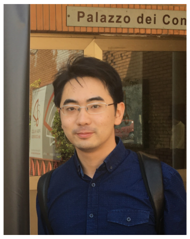
[37] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[38] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *ICML*. PMLR, 2017, pp. 1321–1330.

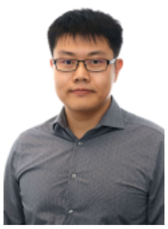
Xun Deng is currently a Master student in cyberspace science and technology from the University of Science and Technology of China(USTC), Hefei, China. He received the B.E. degree in Electronic and Information Engineering from USTC in 2021. His research interests lie in causal inference and counterfactual generation.

Fuli Feng is a professor in University of Science and Technology of China. He received Ph.D. in Computer Science from National University of Singapore in 2019. His research interests include information retrieval, data mining, causal inference and multi-media processing. He has over 60 publications appeared in several top conferences such as SIGIR, WWW, and SIGKDD, and journals including TKDE and TOIS. He has received the Best Paper Honourable Mention of SIGIR 2021 and Best Poster Award of WWW 2018. Moreover, he has been served as the PC member for several top conferences including SIGIR, WWW, SIGKDD, NeurIPS, ICML, ICLR, ACL and invited reviewer for prestigious journals such as TOIS, TKDE, TNNLS, TPAMI.

Xiang Wang is now a professor at the University of Science and Technology of China (USTC). He received his Ph.D. degree from National University of Singapore in 2019. His research interests include recommender systems, graph learning, and explainable deep learning techniques. He has published some academic papers on international conferences such as NeurIPS, ICLR, KDD, WWW, SIGIR. He serves as a program committee member for several top conferences such as KDD, SIGIR, WWW, and IJCAI, and invited reviewer for prestigious journals such as TKDE, TOIS, TNNLS.



Xiangnan He is a professor at the University of Science and Technology of China (USTC). He received his Ph.D. in Computer Science from the National University of Singapore (NUS). His research interests span information retrieval, data mining, and multi-media analytics. He has over 80 publications that appeared in several top conferences such as SIGIR, WWW, and MM, and journals including TKDE, TOIS, and TMM. His work has received the Best Paper Award Honorable Mention in WWW 2018 and ACM SIGIR 2016. He is in the editorial board of journals including Frontiers in Big Data, AI Open etc. Moreover, he has served as the PC chair of CCIS 2019 and SPC/PC member for several top conferences including SIGIR, WWW, KDD, MM, WSDM, ICML etc., and the regular reviewer for journals including TKDE, TOIS, TMM, etc.



Hanwang Zhang is currently an Assistant Professor at Nanyang Technological University, Singapore. He was a research scientist at the Department of Computer Science, Columbia University, USA. He has received the B.Eng (Hons.) degree in computer science from Zhejiang University, Hangzhou, China, in 2009, and the Ph.D. degree in computer science from the National University of Singapore in 2014. His research interest includes computer vision, multimedia, and social media. Dr. Zhang is the recipient of the Best Demo runner-up award in ACM MM 2012, the Best Student Paper award in ACM MM 2013, and the Best Paper Honorable Mention in ACM SIGIR 2016, and TOMM best paper award 2018. He is also the winner of Best Ph.D. Thesis Award of School of Computing, National University of Singapore, 2014.

Tat-Seng Chua is the KITHCT Chair Professor at the School of Computing, National University of Singapore. He was the Acting and Founding Dean of the School from 1998-2000. Dr Chua's main research interest is in multimedia information retrieval and social media analytics. In particular, his research focuses on the extraction, retrieval and question-answering (QA) of text and rich media arising from the Web and multiple social networks. He is the co-Director of NExT, a joint Center between NUS and Tsinghua University to develop technologies for live social media search. Dr Chua is the 2015 winner of the prestigious ACM SIGMM award for Outstanding Technical Contributions to Multimedia Computing, Communications and Applications. He is the Chair of steering committee of ACM International Conference on Multimedia Retrieval (ICMR) and Multimedia Modeling (MMM) conference series. Dr Chua is also the General Co-Chair of ACM Multimedia 2005, ACM CIVR (now ACM ICMR) 2005, ACM SIGIR 2008, and ACMWeb Science 2015. He serves in the editorial boards of four international journals. Dr. Chua is the co-Founder of two technology startup companies in Singapore. He holds a PhD from the University of Leeds, UK.