A PSEUDO CODE OF GRDNORM SCORE

Our proposed GRDNORM Score for OOD error estimation can be calculated as shown in Algorithm 1.

Algorithm 1 OOD Error Estimation via GRDNORM Score

Input: OOD test dataset $\tilde{\mathcal{D}} = \{\tilde{\mathbf{x}}_i\}_{i=1}^{m}$, a pre-trained model $f_{\boldsymbol{\theta}} = f_{\mathbf{g}} \circ f_{\boldsymbol{\omega}}$ (feature extractor $f_{\mathbf{g}}$ and classifier $f_{\boldsymbol{\omega}}$), a threshold value τ.
Output: The GRDNORM Score.
for each OOD instance $\tilde{\mathbf{x}}_i$ **do**
  Obtain the maximum softmax probability via $\tilde{r}_i = \max_k \mathrm{s}_{\boldsymbol{\omega}}^{(k)}(f_{\mathbf{g}}(\tilde{\mathbf{x}}_i))$.
  if $\tilde{r}_i > \tau$ **then**
    Obtain pseudo labels via $\tilde{y}_i' = \arg\max_k f_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_i)$,
  else
    Obtain random labels via $\tilde{y}_i' \sim U[1, K]$.
  end if
end for
Calculate the cross-entropy loss using assigned labels $\tilde{y}_i$ via Eq. 6.
Calculate gradients of the weights in the classification layer via Eq. 7.
Calculate GRDNORM Score $S(\tilde{\mathcal{D}})$ via Eq. 8.

B RELATED WORK

In this section, we first introduce the relevant literature in the field of our interest, OOD error estimation. Then, we list two fields, OOD detection and generalization error bounds, with their respective related work to clarify the difference from OOD error estimation, avoiding potential misunderstanding and confusion with our problem.

OOD error estimation. OOD error estimation is a vital topic in practical applications due to frequent distribution shifts and infeasible ground-truth labels of the test samples. To comprehensively understand this field, we introduce two main existing settings which are related to this topic.

1. Some works aim to estimate the test error or gauge the accuracy discrepancy between the training and the test set only via the training data (Corneanu et al., 2020; Jiang et al., 2019; Neyshabur et al., 2017; Unterthiner et al., 2020; Yak et al., 2019; Martin & Mahoney, 2020). For example, the model-architecture-based algorithm (Corneanu et al., 2020) derives plenty of persistent topology properties from the training data, which can identify when the model learns to generalize to unseen datasets. However, those algorithms are deployed under the assumption that the training and the test data are drawn from the same distribution, which means they are vulnerable to distribution shifts.
2. Our work belongs to the second setting, which aims to estimate the classification accuracy of a specific OOD test dataset during evaluation using unlabeled test samples and/or labeled training datasets. The main research direction is to explore the negative relationship between the distribution discrepancy and model performance from the space of features (Deng & Zheng, 2021), parameters (Yu et al., 2022b) and labels (Lu et al., 2023). Another popular direction is to design an OOD estimation score via the softmax outputs of the test samples (Guillory et al., 2021; Jiang et al., 2021; Guillory et al., 2021; Garg et al., 2022), which heavily relies on model calibration. Some works also learn from the field of unsupervised learning, such as agreement across multiple classifiers (Jiang et al., 2021; Madani et al., 2004; Platanios et al., 2016; 2017) and image rotation (Deng et al., 2021). In addition, the property of the test datasets presented during evaluation has been also studied recently (Xie et al., 2023). To the best of our knowledge, our work is the first to study the linear relationship between the gradients and model performance.

OOD detection. Out-of-distribution (OOD) detection is another essential building block for machine learning safety, whose goal is to determine whether a given sample is in-distribution (ID) or out-of-distribution (Hendrycks & Gimpel, 2016; Hendrycks et al., 2018; Liu et al., 2020; Yang et al., 2021;

Liang et al., 2017). For example, a common baseline uses the maximum softmax probabilities to detect OOD data, assuming that samples with lower softmax probabilities tend to be OOD samples (Hendrycks & Gimpel, 2016). Particularly, gradient norm is also explored in this field under the intuition that ID data need a higher magnitude of gradients than OOD data for adjusting from the softmax probabilities to a uniform distribution (Huang et al., 2021). Nevertheless, OOD detection discusses label-space shifts, where ID and OOD data have disjoint label sets. In contrast, distribution shifts in OOD error estimation do not change the label space shared by the training and the test datasets.

Remark B.1 (Main differences between OOD detection and OOD error estimation)**.** *The OOD error estimation problem should not be confused with the OOD detection problem which received a significant amount of attention in the literature as well. Indeed, the latter, as considered for instance in Huang et al. (2021); Igoe et al. (2022), seeks to solve a binary classification problem of predicting whether a given test instance is OOD. Additionally, the meaning of OOD in such a setting commonly refers to a dataset with a non-overlapping label set. OOD error estimation requires a dataset-based score, mainly considered in distribution shift cases where the label sets can be overlapping. The two are also evaluated differently: AUROC score for OOD detection and correlation coefficients for error estimation. Those differences are summarized in Table 2. In Appendix E, we show that a recent gradient-based OOD detection method (Huang et al., 2021) is inferior to our approach for OOD error estimation suggesting that the two problems cannot be tackled with the same tools.*

Table 2: Main differences between OOD detection and OOD error estimation.

Learning Problem	Goal	Scope	Metric
OOD detection	Predict ID/OOD	$\tilde{\mathbf{x}}_i$	AUROC
OOD error estimation	Proxy to test error	$\mathcal{D}_{\text{test}}$	R^2 and ρ

Generalization error bounds. Generalization error, also known as the out-of-sample error, gauges the generalization performance of the hypothesis learned from the training data and applied to previous unseen samples (Hardt et al., 2016; London, 2017; Rivasplata et al., 2018). Many studies try to provide a tight upper bound for generalization error from the view of gradient descent theoretically, which indicates that gradients correlate with the discrepancy between the empirical loss and the population loss (Li et al., 2019; Chatterjee, 2020; Negrea et al., 2019; An et al., 2020). However, those works mainly focus on the generalization performance from seen data to unseen data under the identical distribution, while OOD error estimation discusses a more complex and realistic issue that seen and unseen data come from different distributions.

C Baseline methods

Rotation. (Deng et al., 2021) By rotating images from both the training and the test sets with different angles, we can obtain new inputs and their corresponding labels y_i^r which indicate by how many degrees they rotate. During pre-training, an additional classifier about rotation degrees should be learned. Then the *Rotation Prediction* (Rotation) metric can be calculated as:

$$S_r(\mathcal{D}_{\text{test}}) = \frac{1}{m}\sum_{i=1}^{m}\Big(\frac{1}{4}\sum_{r\in\{0°,90°,180°,270°\}}(\mathbb{1}(\hat{y}_i^r \neq y_i^r))\Big),$$

where $\hat{y}_i^r$ denotes the predicted labels about rotation degrees.

ConfScore. (Hendrycks & Gimpel, 2016) This metric directly leverages the average maximum softmax probability as the estimation of OOD error, which is expressed as:

$$S_{cf}(\mathcal{D}_{\text{test}}) = \frac{1}{m}\sum_{i=1}^{m}\max(\mathrm{s}_{\boldsymbol{\omega}}(f_{\boldsymbol{g}}(\tilde{\mathbf{x}}_i))).$$

Entropy. (Guillory et al., 2021) This metric estimates OOD error via the average entropy loss:

$$S_e(\mathcal{D}_{\text{test}}) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(f_{\boldsymbol{g}}(\tilde{\mathbf{x}}_i))\log\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(f_{\boldsymbol{g}}(\tilde{\mathbf{x}}_i)).$$

AgreeScore. (Jiang et al., 2021) This method trains two independent neural networks simultaneously during pre-training, and estimates OOD error via the rate of disagreement across the two models:

$$S_{ag}(\mathcal{D}_{\text{test}}) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{1}(\tilde{y}'_{1,i} \neq \tilde{y}'_{2,i}),$$

where $\tilde{y}'_{1,i}$ and $\tilde{y}'_{2,i}$ denote the predicted labels by the two models respectively.

ATC. (Garg et al., 2022) It measures how many test samples have a confidence larger than a threshold that is learned from the source distribution. It can be expressed as:

$$S_{atc}(\mathcal{D}_{\text{test}}) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{1}(\sum_{k=1}^{K}\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(f_{\boldsymbol{g}}(\tilde{\mathbf{x}}_i))\log\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(f_{\boldsymbol{g}}(\tilde{\mathbf{x}}_i)) < t),$$

where t is the threshold value learned from the validation set of the training dataset.

Fréchet. (Deng & Zheng, 2021) This method utilizes Fréchet distance to measure the distribution gap between the training and the test datasets, which provides the OOD error estimation:

$$S_{fr}(\mathcal{D}_{\text{test}}) = ||\mu_{train} - \mu_{test}|| + Tr(\Sigma_{train} + \Sigma_{test} - 2(\Sigma_{train}\Sigma_{test})^{\frac{1}{2}}),$$

where μ_{train} and μ_{test} denote the mean feature vector of $\mathcal{D}$ and $\mathcal{D}_{test}$, respectively. Σ_{train} and Σ_{test} refer to the covariance matrices of corresponding datasets.

Dispersion. (Xie et al., 2023) This paper estimates OOD error by gauging the feature separability of the test dataset in the feature space:

$$S_{dis}(\mathcal{D}_{\text{test}}) = \log\frac{\sum_{k=1}^{K} m_k \cdot \|\bar{\boldsymbol{\mu}} - \tilde{\boldsymbol{\mu}}_k\|_2^2}{K-1},$$

where $\boldsymbol{\mu}$ denotes the center of the whole features, and $\boldsymbol{\mu}_k$ denotes the mean of k^{th}-class features.

ProjNorm. (Yu et al., 2022b) This method fine-tunes the pre-trained model on the test dataset with pseudo-labels, and measures the distribution discrepancy between the training and the test datasets in the parameter level:

$$S_{pro}(\mathcal{D}_{\text{test}}) = ||\tilde{\boldsymbol{\theta}}_{ref} - \tilde{\boldsymbol{\theta}}||_2,$$

where $\boldsymbol{\theta}_{ref}$ denotes the parameters of the pre-trained model, while $\boldsymbol{\theta}$ denotes the parameters of the fine-tuned model.

Those algorithms mentioned in this paper can be summarized as Table 3.

D ENTROPY LOSS FOR LOW-CONFIDENCE SAMPLES

In the ablation study, we explore the impact of loss selection on the performance of OOD error estimation. In particular, the detail about the entropy loss for samples with low confidence is expressed as follows: In particular, the entropy loss can be expressed as follows:

$$\begin{aligned}\mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})) = &-\frac{1}{m_1}\sum_{i=1}^{m_1}\sum_{k=1}^{K}\tilde{y}_{i,con>\tau}^{(k)}\log\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(f_{\boldsymbol{g}}(\tilde{x}_i^{con>\tau})) \\ &-\frac{1}{m_2}\sum_{i=1}^{m_2}\sum_{k=1}^{K}\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(f_{\boldsymbol{g}}(\tilde{\mathbf{x}}_i^{con\leq\tau}))\log\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(f_{\boldsymbol{g}}(\tilde{\mathbf{x}}_i^{con\leq\tau})),\end{aligned}$$

where the first term denotes the cross-entropy loss calculated for samples with confidence larger than the threshold value τ, the second term denotes the entropy loss for samples with lower confidence than τ, and con means the sample confidence, m_1 and m_2 denote the total number of samples with higher confidence and lower confidence than τ, respectively.

Table 3: Method property summary including whether this method belongs to self-training or training-free approaches, and if this method requires training data or specific model architectures.

Method	Self-training	Training-free	Training-data-free	Architecture-requirement-free
Rotation	✗	✓	✓	✗
ConfScore	✗	✓	✓	✓
Entropy	✗	✓	✓	✓
Agreement	✗	✓	✓	✗
ATC	✗	✓	✗	✓
Fréchet	✗	✓	✗	✓
Dispersion	✗	✓	✓	✓
ProjNorm	✓	✗	✓	✓
Ours	✓	✗	✓	✓

E DISCUSSION: RELATION TO HUANG ET AL. (2021)

A current work, GradNorm (Huang et al., 2021), employs gradients to detect OOD samples which labels belong to a different label space from the training data. It gauges the magnitude of gradients in the classification layer, backpropagated from a KL-divergence between the softmax probability and a uniform distribution. Compared with GradNorm, our method bears three critical differences, in terms of the problem setting, methodology, and theoretical insights. We also empirically demonstrate the superiority of our method in Table 4.

1) *Problem setting*: GradNorm focuses on OOD detection, where the label spaces of OOD data and training data are disjoint, while our method aims to estimate the test error without ground-truth test labels, where the training and the OOD label spaces are shared.

2) *Methodology*: Essentially, GradNorm measures the magnitude of gradients from the prediction probability to the uniform distribution, while our method measures it from the source distribution to the target distribution. This basic difference is due to different aims, and is specifically reflected in the design of losses, label generalization strategies and evaluation approaches.

3) *Theoretical insights*: Theoretically, GradNorm captures the joint information between features and outputs to detect OOD data from the oncoming dataset, while our method provides two types of parameter discrepancy information that are beneficial to predicting OOD performance. Formally, we also demonstrate that our score formulates the upper bound of the true OOD error, which further explains the effectiveness of our method.

In Table 4, we present the performance comparison of the two methods in OOD error estimation on 7 datasets across 3 types of distribution shifts with ResNet18. This table illustrates that our method is superior to Huang et al. (2021): for example, our method outperforms Huang et al. (2021) on TinyImageNet with a large margin from 0.894 to 0.971. This shows that comparing the softmax outputs to uniform distribution as done by Huang et al. (2021) is relevant for detecting test samples from a different label space only. However, for OOD error estimation with overlapping labels, estimating the target distribution through pseudo-labeling – rather than assuming it to be uniform – is more informative and achieves much better results.

Table 4: Performance comparison between Huang et al. (2021) and our methods on 7 datasets with ResNet18. The metric used in this table is the coefficient of determination R^2. The best results are highlighted in **bold**.

Method	CIFAR 10	CIFAR 100	TinyImageNet	Office-31	Office-Home	Entity-30	Living-17
(Huang et al., 2021)	0.951	0.978	0.894	0.596	0.848	0.964	0.942
Ours	**0.972**	**0.983**	**0.971**	**0.675**	**0.876**	**0.970**	**0.949**

F CHOICE OF PROPER THRESHOLD τ

In our experiments (see Section 4), we set the value of τ as 0.5 across all datasets and network architectures. This choice of τ is due to the intuition that if a label contains a softmax probability below 0.5, it means that this predicted label has over 50% chances of being wrong. It means that this label has a higher probability of being incorrect than to be correct. Thus, we tend to regard it as an incorrect prediction. To demonstrate the impact of threshold τ on the final performance, we conduct an ablation study on CIFAR-10 with ResNet18 with varying values of τ. We display in Table 5 the corresponding values of R^2. We can observe that the final performance improves and achieves its best value for τ is 0.5, before decreasing slightly.

Table 5: Performance on CIFAR-10 with ResNet18 for varying value of τ. The metric used in this table is the coefficient of determination R^2. The best result is highlighted in **bold**.

Threshold	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
R^2	0.963	0.963	0.964	0.965	0.971	**0.972**	0.967	0.962	0.963	0.959

G INFLUENCE OF CALIBRATION ERROR

We have mentioned earlier that, in theory, the proposed pseudo-labeling strategy depends on how well the prediction probabilities are calibrated. In degraded cases, this can have a negative impact on our approach, e.g., one can imagine a model that outputs only one-hot probabilities with not a high accuracy. However, this it is generally not the case. Indeed, in practice, we do not need to have a perfectly calibrated model as we employ a mixed strategy that assigns pseudo-labels to high-confidence examples and random labels to low-confidence ones. The recent success of applying self-training models to different problems (Sohn et al., 2020; Dong et al., 2021; Yu et al., 2022a) provides evidence of the suitability of the label generation strategy we adopted.

When we speak of deep neural networks, which are widely accepted to be poorly calibrated, Minderer et al. (2021) showed that modern SOTA image models tend to be well-calibrated across distribution shifts. To demonstrate it empirically, in Table 6, we provide the expected calibration error (ECE, Guo et al. (2017)) of ResNet18, one of the considered base models, depending on a difficulty of test data. For this, we test first on CIFAR-10 (ID), and then on CIFAR-10C corrupted by brightness across diverse severity from 1 to 5. We can see that ECE is very low for ID data and remains relatively low across all levels of corruption severity, which shows that ResNet is quite well-calibrated on CIFAR-10.

Table 6: Expected Error Calibration (ECE) of ResNet18 on CIFAR-10 (ID) and CIFAR-10C corrupted by brightness across diverse severity from 1 to 5.

Corruption Severity	ID	1	2	3	4	5
ECE	0.0067	0.0223	0.0230	0.0243	0.0255	0.0339

On the other hand, in the case of more complex distribution shift like Office-31 data set, we can see that the calibration error has been increased noticeably (Table 7). It is interesting to analyze this result together with Figure 3 of the main paper, where we compared the results between the usual pseudo-labeling strategy and the proposed one. Although our method has room for improvement compared to the oracle method, it is also significantly better than "pseudo-labels", indicating that the proposed label generation strategy is less sensitive to the calibration error.

Table 7: Expected Error Calibration (ECE) of ResNet18 on Office-31 data set.

Domain	DSLR (ID)	Amazon	Webcam
ECE	0.2183	0.2167	0.4408

H PROOFS

H.1 PROOF OF THEOREM 3.1

We start by proving the following lemma.

Lemma H.1. *For any convex function $f : \mathbb{R}^D \to \mathbb{R}$ and any $p, q \geq 1$ such that $\frac{1}{p} + \frac{1}{q} = 1$, we have:*

$$\forall \mathbf{a}, \mathbf{b} \in dom(f), \quad |f(\mathbf{a}) - f(\mathbf{b})| \leq \max_{\mathbf{c} \in \{\mathbf{a}, \mathbf{b}\}} \{\|\nabla f(\mathbf{c})\|_p\} \cdot \|\mathbf{a} - \mathbf{b}\|_q .$$

Proof. Using the fact that f is convex, we have:

$$\begin{aligned} f(\mathbf{a}) - f(\mathbf{b}) &\leq \langle \nabla f(\mathbf{a}), \mathbf{a} - \mathbf{b} \rangle \\ &\leq |\langle \nabla f(\mathbf{a}), \mathbf{a} - \mathbf{b} \rangle| \\ &\leq \sum_{i=1}^{p} |\nabla f(\mathbf{a})_i \, (\mathbf{a}_i - \mathbf{b}_i)| \\ &\leq \|\nabla f(\mathbf{a})\|_p \|\mathbf{a} - \mathbf{b}\|_q, \end{aligned}$$

where we used Hölder's inequality for the last inequality. The same argument gives:

$$f(\mathbf{b}) - f(\mathbf{a}) \leq \|\nabla f(\mathbf{b})\|_p \|\mathbf{b} - \mathbf{a}\|_q.$$

Using the absolute value, we can combining the two previous results and obtain the desired inequality. □

The proof of Theorem 3.1 follows from applying Lemma H.1 to the convex function $\mathcal{L}_T$.

H.2 PROOF OF THEOREM 3.3

We start by introducing some notations. We denote $\mathcal{L}_{\mathbf{x},y}$ the loss evaluated on a specific data-point $(\mathbf{x}, y) \sim P_T(\mathbf{x}, \mathbf{y})$. We can then decompose the expected loss as $\mathcal{L}_T = \mathbb{E}_{P_T(\mathbf{x},y)} \mathcal{L}_{\mathbf{x},y}$. It follows by linearity of the expectation that

$$\nabla \mathcal{L}_T = \mathbb{E}_{P_T(\mathbf{x},y)} \nabla \mathcal{L}_{\mathbf{x},y}.$$

Then, we prove the following lemma that gives the formulation of the gradient of the cross-entropy.

Lemma H.2. *The gradient of the cross-entropy loss with respect to $\boldsymbol{\omega} = (\mathbf{w}_k)_{k=1}^K$ writes*

$$\nabla \mathcal{L}_{\mathbf{x},y}(\boldsymbol{\omega}) = \left(-y^{(k)} \mathbf{x} (1 - \mathrm{s}_{\boldsymbol{\omega}}^{(k)}(\mathbf{x}) \right)_{k=1}^{K}.$$

Proof. First, let's compute the partial derivative of the softmax w.r.t. $\mathbf{w}_k$ for any $k \in \{1, \ldots, K\}$. We have:

$$\begin{aligned} \frac{\partial \mathrm{s}_{\boldsymbol{\omega}}^{(k)}(\mathbf{x})}{\partial \mathbf{w}_k} &= \frac{\mathbf{x} e^{\mathbf{w}_k^\top \mathbf{x}} \left(\sum_{\tilde{k}} e^{\mathbf{w}_{\tilde{k}}^\top \mathbf{x}} - e^{\mathbf{w}_k^\top \mathbf{x}} \right)}{\left(\sum_{\tilde{k}} e^{\mathbf{w}_{\tilde{k}}^\top \mathbf{x}} \right)^2} \\ &= \mathbf{x} \left(\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(\mathbf{x}) - \left[\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(\mathbf{x}) \right]^2 \right). \end{aligned}$$

Using the chain rule, the partial derivative of the loss w.r.t. $\mathbf{w}_k$ writes:

$$\begin{aligned} \frac{\partial \mathcal{L}_{\mathbf{x},y}(\boldsymbol{\omega})}{\partial \mathbf{w}_k} &= \frac{\partial \mathcal{L}_{\mathbf{x},y}(\boldsymbol{\omega})}{\partial s(\boldsymbol{\omega}, \mathbf{x})} \cdot \frac{\partial s(\boldsymbol{\omega}, \mathbf{x})}{\partial \mathbf{w}_k} \\ &= \begin{cases} -\frac{1}{\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(\mathbf{x})} \cdot \mathbf{x} \left(\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(\mathbf{x}) - \left[\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(\mathbf{x}) \right]^2 \right), & \text{if } y^{(k)} = 1 \\ 0, & \text{otherwise} \end{cases} \\ &= -y^{(k)} \mathbf{x} \left(1 - \mathrm{s}_{\boldsymbol{\omega}}^{(k)}(\mathbf{x}) \right) \end{aligned}$$

As the $\frac{\partial \mathcal{L}_{\mathbf{x},y}(\boldsymbol{\omega})}{\partial \mathbf{w}_k}$ are the coordinates of $\nabla \mathcal{L}_{\mathbf{x},y}(\boldsymbol{\omega})$, we obtain the desired formulation. □

We now proceed to the proof of Theorem 3.3.

Proof. Using the convexity of $\|\cdot\|_p$ and the Jensen inequality, we have that

$$\begin{aligned}
\|\nabla\mathcal{L}_T(\boldsymbol{\omega})\|_p &= \|\mathbb{E}_{P_T(\mathbf{x},y)}\nabla\mathcal{L}_{\mathbf{x},y}(\boldsymbol{\omega})\|_p \\
&\leq \mathbb{E}_{P_T(\mathbf{x},y)}\|\mathcal{L}_{\mathbf{x},y}(\boldsymbol{\omega})\|_p && \text{(Jensen inequality)} \\
&= \mathbb{E}_{P_T(\mathbf{x},y)}\left(\sum_{i=1}^{D}\sum_{k=1}^{K}|-y^{(k)}\mathbf{x}_i(1-\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(\mathbf{x})|^p\right)^{1/p} \\
&= \mathbb{E}_{P_T(\mathbf{x},y)}\left(\sum_{k=1}^{K}y^{(k)}\left(1-\mathrm{s}_{\boldsymbol{\omega}}^{(k)}(\mathbf{x})\right)^p\right)^{1/p}\left(\sum_{i=1}^{D}|\mathbf{x}_i^p|\right)^{1/p} \\
&= \mathbb{E}_{P_T(\mathbf{x},y)}\left(\left(1-\mathrm{s}_{\boldsymbol{\omega}}^{(k_y)}(\mathbf{x})\right)^p\right)^{1/p}\left(\sum_{i=1}^{D}|\mathbf{x}_i^p|\right)^{1/p} && (k_y \text{ such that } y^{(k_y)}=1) \\
&= \mathbb{E}_{P_T(\mathbf{x},y)}\alpha(\boldsymbol{\omega},\mathbf{x},y)\|\mathbf{x}\|_p,
\end{aligned}$$

where $\alpha(\boldsymbol{\omega},\mathbf{x},y) = \left(1-\mathrm{s}_{\boldsymbol{\omega}}^{(k_y)}(\mathbf{x})\right)$, with k_y such that $y^{(k_y)}=1$. We used the fact that $\mathbf{y}$ is a one-hot vector so it has only one nonzero entry. □

H.3 Proof of Remark 5.1

Proof. Using the reverse Minkowski inequality, as $0 < p < 1$, we have that

$$\begin{aligned}
&\|\boldsymbol{\omega}_s\|_p = \|\mathbf{c} + \eta\cdot\nabla\mathcal{L}_T(\boldsymbol{\omega}_s)\|_p \geq \|\mathbf{c}\|_p + \eta\cdot\|\nabla\mathcal{L}_T(\boldsymbol{\omega}_s)\|_p \\
\implies &\|\boldsymbol{\omega}_s\|_p - \|\mathbf{c}\|_p \geq \eta\cdot\|\nabla\mathcal{L}_T(\boldsymbol{\omega}_s)\|_p.
\end{aligned}$$

In the same fashion, we have that

$$\begin{aligned}
&\|\mathbf{c}\|_p = \|\boldsymbol{\omega}_s - \eta\cdot\nabla\mathcal{L}_T(\boldsymbol{\omega}_s)\|_p \geq \|\boldsymbol{\omega}_s\|_p + \eta\cdot\|\nabla\mathcal{L}_T(\boldsymbol{\omega}_s)\|_p \\
\implies &\|\mathbf{c}\|_p - \|\boldsymbol{\omega}_s\|_p \geq \eta\cdot\|\nabla\mathcal{L}_T(\boldsymbol{\omega}_s)\|_p.
\end{aligned}$$

We obtain the desired upper bound by combining those results. □